Preuves symboliques de propriétés d'indistinguabilité calculatoire

Adrien Koutsos

NNT : 2019SACLN029

Thèse de doctorat

Preuves symboliques de propriétés d'indistinguabilité calculatoire

Thèse de doctorat de l'Université Paris-Saclay
préparée à École Normale Supérieure Paris-Saclay
au sein du Laboratoire Spécification et Vérification

Ecole doctorale n°580 Sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat : Informatique

Thèse présentée et soutenue à Cachan, le 27 septembre 2019, par

ADRIEN KOUTSOS

Composition du Jury :

Catuscia Palamidessi Directrice de Recherche, INRIA	Présidente
Cas Cremers Professeur, CISPA Helmholtz	Rapporteur
Bogdan Warinschi Professeur, University of Bristol	Rapporteur
Myrto Arapinis Reader, University of Edinburgh	Examinatrice
Bruno Blanchet Directeur de Recherche, INRIA	Examinateur
Hubert Comon Professeur, ENS Paris-Saclay	Directeur de thèse

Résumé

Notre société utilise de nombreux systèmes de communications. Parce que ces systèmes sont omniprésents et sont utilisés pour échanger des informations sensibles, ils doivent être protégés. Cela est fait à l'aide de *protocoles cryptographiques*. Essentiellement, un protocole est un ensemble de règles détaillant comment des entités, par exemples des systèmes informatisés, doivent communiquer, et un protocole cryptographique est un protocole qui cherche à garantir certaines propriétés de sécurité. Il est crucial que ces protocoles assurent bien les propriétés de sécurité qu'ils affirment avoir, car les échecs peuvent avoir des conséquences importantes. Par exemple, ils peuvent entraîner des fuites de données confidentielles, ou des atteintes majeures au respect de la vie privée des utilisateurs.

Malheureusement, concevoir des protocoles cryptographiques est notoirement difficile, comme le montre la régularité avec laquelle de nouvelles attaques sont découvertes. De plus, des attaques sont trouvées régulièrement même sur des protocoles de premier plan, tel le protocole TLS qui est utilisé pour sécuriser les connections HTTPS. Nous pensons que la vérification formelle est le meilleur moyen d'avoir de bonnes garanties dans la sécurité d'un protocole. Essentiellement, il s'agit de prouver mathématiquement qu'un protocole satisfait une certaine propriété de sécurité. Bien entendu, ce n'est pas une tâche aisée. Tout d'abord, il faut modéliser fidèlement le protocole et la propriété de sécurité, tout en abstrayant les aspects du système qui ne sont pas pertinents. Ensuite, il faut prouver que le modèle du protocole satisfait bien la propriété voulue. En particulier, cela nécessite d'avoir formalisé la classe d'attaquants contre laquelle la propriété doit être valide. Plusieurs classes d'attaquants ont été proposées dans la littérature.

Un modèle d'attaquants populaire, le modèle de *Dolev-Yao*, donne à l'attaquant le contrôle du réseau: celui-ci peut intercepter et rediriger tous les messages. De plus, l'attaquant peut modifier les messages en utilisant un ensemble fixé de règles. Ce modèle est particulièrement adapté aux preuves automatiques de propriétés de sécurité, mais il donne des garanties limitées, puisque l'on prouve seulement l'absence d'attaques utilisant les capacités données à l'attaquant.

Un autre modèle d'attaquant, plus proche d'un attaquant réel, est celui de *l'attaquant calculatoire*. Dans ce modèle, l'attaquant contrôle aussi le réseau. Cependant, celui-ci n'est pas restreint à un ensemble fixé de règles: il peut effectuer n'importe quel calcul probabiliste polynomial. Ce modèle offre de meilleures garanties de sécurité, mais les preuves sont plus difficiles à réaliser, sujettes à erreurs et plus difficiles à automatiser.

Il existe une approche alternative, le modèle Bana-Comon. Dans ce modèle, on exprime la sécurité d'un protocole comme un problème de satisfaisabilité d'un ensemble de formules de la logique du premier ordre. Cet ensemble de formules contient la négation de la propriété de sécurité et un ensemble d'*axiomes*, qui correspondent à des hypothèses d'implémentations, telle que la correction fonctionnelle, et à des hypothèses cryptographiques sur les primitives de sécurité. Prouver l'insatisfaisabilité de cet ensemble de formules implique la sécurité du protocole dans le modèle calculatoire. De plus, puisqu'il s'agit d'une logique du premier ordre, ce modèle est adapté aux méthodes de preuves automatiques. Il existe deux modèles Bana-Comon, qui ciblent différentes propriétés de sécurité. Le modèle le plus ancien est destiné aux propriétés d'accessibilités, ou de traces, alors que le modèle le plus récent s'intéresse aux propriétés d'équivalences. Ces dernières sont plus expressives, et sont nécessaires pour énoncer des propriétés liées au respect de la vie privée, tels que l'anonymat ou la non-traçabilité.

Notre objectif est de développer les techniques permettant de vérifier formellement des propriétés d'équivalence sur des protocoles cryptographiques, en utilisant une méthode qui fournit de fortes garanties de sécurités, tout en étant adaptée à des procédures de preuve automatique. Dans cette thèse, nous défendons l'idée que le modèle Bana-Comon pour les propriétés d'équivalences satisfait ces objectifs. Nous soutenons cette affirmation à l'aide de trois contributions.

Tout d'abord, nous étayons le modèle Bana-Comon en concevant des axiomes pour les fonctions usuelles des protocoles de sécurités, comme le xor, et pour plusieurs hypothèses cryptographiques: IND-CCA$_1$, CR-HK, EUF-CMA et PRF.

Dans un second temps, nous illustrons l'utilité de ces axiomes et du modèle en réalisant deux études de cas de protocoles concrets. Nous commençons avec deux protocoles relativement simples, KCL et LAK. Puisque des attaques contre ces protocoles sont connues, nous proposons des corrections, et prouvons que les versions corrigées protègent la vie privée des utilisateurs, en supposant que les fonctions de hachages sont des PRF. Notre deuxième étude de cas est plus complexe. Dans cette étude de cas, nous nous intéressons au protocole d'authentification 5G-AKA, qui est utilisé dans les réseaux de téléphonie mobile, et montrons que de nombreuses attaques de la littérature sont applicables à ce protocole. Nous proposons alors une version modifiée du protocole, que nous appelons AKA$^+$, et nous prouvons à l'aide du modèle Bana-Comon que celle-ci garantie l'authentification mutuelle et la non-traçabilité des utilisateurs. Ce résultat est valide pour un nombre arbitraire d'utilisateurs et de sessions.

Finalement, nous étudions le problème de l'automatisation de la recherche de preuves dans le modèle Bana-Comon. Pour cela, nous prouvons la décidabilité d'un ensemble de règles d'inférences qui est une axiomatisation correcte, bien que incomplète, de l'indistingabilité calculatoire, lorsque l'on utilise un schéma de chiffrement IND-CCA$_2$. Du point de vue d'un cryptographe, cela peut être interprété comme la décidabilité d'un ensemble de transformations de jeux. Ce résultat repose sur des techniques de déduction automatiques standards, comme la normalisation de termes et l'élimination de coupures.

Abstract

Our society extensively relies on communications systems. Because such systems are used to exchange sensitive information and are pervasive, they need to be secured. *Cryptographic protocols* are what allow us to have secure communications. Basically, a protocol is a set of rules detailing how entities, e.g. computer systems, must communicate, and a cryptographic protocol is a protocol that aims at ensuring some security properties. It is crucial that such protocols do not fail in providing the security properties they claim, as such failures have dire consequences. For example, they can lead to sensitive data being stolen, or to large scale privacy breaches.

Unfortunately, designing cryptographic protocols is notoriously hard, and major protocols are regularly and successfully attacked. Moreover, this is true even for high-visibility protocols, such as the TLS protocol which is used to secure HTTPS connections. Formal verification is the best way to get a strong confidence in a protocol security. Basically, the goal is to mathematically prove that a protocol satisfies some security property. Of course, this is not an easy task. First, we need to faithfully model the protocol and the security property, while abstracting away irrelevant aspects of the system. Second, we have to prove that the modeled protocol indeed satisfies the desired property. In particular, this requires us to formally specify against what class of attackers the property must hold. Several classes of attackers have been considered in the literature.

A popular attacker model, the *Dolev-Yao attacker*, grants the attacker the complete control of the network: he can intercept and re-route all messages. Besides, the adversary is allowed to modify messages using a fixed set of rules. This model is very amenable to automatic verification of security properties, but the security obtained is limited: we only prove the absence of attacks using the capabilities granted to the adversary.

Another attacker model, closer to a real world attacker, is the *computational attacker* model. This adversary also controls the network, but this model does not restrict the attacker to a fixed set of operations: the adversary can perform any probabilistic polynomial time computation. This model offers stronger guarantees than the Dolev-Yao model, but formal proofs are harder to complete, more error-prone, and more difficult to automate.

There is an alternative approach, the Bana-Comon model. In this model, we express the security of a protocol as the unsatisfiability of a set of formulas in first-order logic. The formulas contain the negation of the security property and *axioms*, which reflect implementation assumptions, such as functional correctness and cryptographic hypotheses on the security primitives. Carrying out a proof of unsatisfiability in this logic entails the security of the protocol in the computational model. Moreover, because this is a first-order logic, this model may be amenable to automated or mechanized proofs. There exist two Bana-Comon models, which target different security properties. The oldest model aims at proving reachability or trace properties, while the newest and less studied model targets equivalence properties. These properties are more expressive, and allow to state privacy-related properties, such as *anonymity* or *unlinkability*.

Our objective is to develop techniques to formally verify equivalence properties of cryptographic protocols, using a method that provides strong security guarantees while being amenable to automated deduction techniques. In this thesis, we argue that the Bana-Comon model for equivalence properties meets these goals. We support this claim through three different contributions.

First, we design axioms for the usual functions used in security protocols, such as the xor operator, and for several cryptographic hypothesis: IND-CCA$_1$, CR-HK, EUF-CMA and PRF.

Second, we illustrate the usefulness of these axioms and of the model by completing two case studies of concrete protocols. We start with two simple RFID protocols, KCL and LAK. As these protocols are known to be unsecure, we propose security fixes, and prove that our fixed versions provide privacy under the PRF assumption. Our second case study is more involved. In this case study, we investigate the 5G-AKA authentication protocol used in mobile communication systems, and show that multiple privacy attacks from the literature apply to this protocol. We then propose a fixed version of this protocol, dubbed AKA$^+$, and prove using the Bana-Comon approach that it provides mutual authentication and a form of unlinkability. This result holds for any number of agents and sessions.

Finally, we study the problem of proof automation in the Bana-Comon model, by showing the decidability of a set of inference rules which is a sound, though incomplete, axiomatization of computational indistinguishability when using an IND-CCA$_2$ encryption scheme. From a cryptographer's point of view, this can be seen as the decidability of a fixed set of cryptographic game transformations. This result relies on standard automated deduction techniques, such as term normalization and proof cut eliminations.

Contents

Chapter 1

Introduction

"So it goes."

— Kurt Vonnegut

1.1 The Context

Our society extensively relies on communication systems. The most prominent communication systems are very large scale systems, such as the Internet or the mobile phone cellular networks, through which billions of users are connected. These systems are used by private individuals for messaging, online shopping, accessing bank accounts, paying taxes... They are also used by organizations, such as companies or states, to exchange sensitive data. But there are also smaller-scale and less visible communication systems, which are no less pervasive. For example, RFID badges and smart cards are extensively used for buildings access control or public transportation payment method. Often, the data exchanged through these systems is sensitive, e.g. credit card number or bank account details, or contains information that the user wants to keep private, e.g. his location. To prevent some malicious entity from stealing confidential data, or breaching a user's privacy, communication systems need to be secured.

Cryptographic protocols are what allow us to obtain secure communications. A protocol is a set of rules stating how two or more entities must communicate. Theses rules not only specify the content of the messages that are to be exchanged, but also the order and the recipients of these messages, as well as how the entities local states evolve during the protocol execution. A *cryptographic protocol* is a protocol that aims at ensuring some security properties. The HTTPS protocol is an example of cryptographic protocol, and is used to secure communications between a server and a browser on the *World Wide Web*. Another example of cryptographic protocol is the *Authentication and Key Agreement* (AKA) protocol, which allows a mobile phone and its service provider to authenticate each other and to establish a shared secret key. This key is used to protect future communications between the phone and the service provider.

Attacks Unfortunately, designing security protocols is hard, as can be seen from the numerous attacks against them that have been discovered in the last decades. For example, the TLS protocol, which is used to secure HTTPS connection, has been successfully attacked several times at the protocol level: the LogJam attack [ABD+15] allowed a man-in-the-middle attacker to force a TLS connection to use 512 bit Diffie-Hellman key exchange,[1] which is easy to break using current computing capabilities. This is far from being the only attack on TLS. To cite but a few: the TripleHandshake authentication attack [BDF+14]; or the FREAK downgrade attack [BBD+17a]. The TLS protocol is not the only major protocol to have been attacked. For example, the mobile network authentication protocol AKA is subject to several privacy attacks. The most important privacy attack against AKA is the IMSI catcher attack [Str07]. Using this attack, a rogue antenna can collect the identities of all mobile devices in range, which allows for large-scale surveillance. Note that, contrary to the TLS attacks, this privacy attack has not been fixed in the currently deployed version of the protocol (fourth generation, or 4G).

[1] This is an export-grade key exchange: in the 90's, the United States required that cryptographic software exported abroad used weak keys on purpose, so that they could be easily broken by intelligence agencies.

Formal Methods The fact that new attacks are regularly found on high-visibility protocols, such as TLS and AKA, shows how difficult it is to design secure cryptographic protocols. An explanation for this may be that the usual approach of bug finding, through automated testing, and bug fixing, does not work for security protocols. This is because automated testing is made to find errors occurring during executions of a program on random inputs. The problem is that a cryptographic protocol is not executed in a random environment, but in a hostile one. A corner-case, which has a very low probability of happening in a normal execution, may be systematically triggered by an adversary.

When this happens, or when the potential cost of a bug is deemed too high, such as in the aeronautic space industry, we rely on *formal verification* instead of testing. Basically, the goal of formal verification is to prove, in a mathematical sense, that some system satisfies some properties. Ideally, the proof should be machine-checked, to avoid any errors. This approach gives very strong guarantees, and has been successfully applied. For example, the flight-control program of some Airbus planes have been successfully verified [BCC+15] using abstract interpretation techniques [CC77]. They proved the absence of run-time errors such as division by zero or integer overflows.

To formally verify a system, we need to model the system, model the property, and prove that the system satisfies the property. The modeling is often not obvious, as it requires the prover to abstract away the aspects of the system that are irrelevant, while not forgetting to model any important feature.[2] When modeling a security protocol, we first need to determine what are the network capabilities of the adversary. A first possibility is to give only eavesdropping capacities to the adversary. Because he cannot interfere with the execution of the protocol, such an adversary is called a *passive adversary*. A stronger adversarial model, historically advocated by Needham and Schroeder in [NS78], also lets the adversary intercept, reroute or even forge messages: the adversary has complete control of the network. Such an adversary is called an *active adversary*.

1.2 Example: the AKA$^-$ Protocol

To make things more concrete, we give a simplified example of a real-world cryptographic protocol. The *Authentication and Key Agreement* (AKA) is used in mobile communication networks, and is part of a series of protocols which allow a user, typically a mobile phone, to connect wirelessly to its service provider, in order to send and receive text messages and calls, or to access the Internet. As indicated by its name, this is an authenticated key-exchange protocol. The goal of such a protocol is two-fold. First, it must ensure that the two parties, here the user and its service provider, properly authenticated each other. That is, after a successful completion of the protocol, the user must be certain that it communicated with its service provider, and not another potentially malicious agent. Conversely, the service provider must be certain of the identity of the user it interacted with. Then, this is a key-exchange protocol: at the end of a successful execution of the protocol, the user and the service provider must have established a *shared* and *secret* key, which they can use in subsequent communications.

There are several versions of the AKA protocol, one for each generation of mobile communication networks. The currently deployed variants are the *third* (3G) and *fourth* (4G) generations AKA protocols, but the *fifth generation* (5G) should be finalized soon, and drafts are already available. Our example, which we call AKA$^-$, is a simplified version of the 5G variant of AKA.

1.2.1 Cryptographic Primitives

Cryptographic primitives are the basic building blocks of cryptographic protocols, and provide interesting security properties. We present here three standard such primitives: symmetric encryptions, asymmetric encryptions and cryptographic hash functions.

Symmetric Encryption The best known, and oldest, cryptographic primitive is the *symmetric encryption*. Basically, a symmetric encryption scheme comprises two functions, senc and sdec. The encryption function senc takes as input a message m, called the *plain-text*, and a secret key k, and returns an encrypted message $\mathsf{senc}(m, \mathsf{k})$, called the *cipher-text*. The cipher-text must reveal nothing about the

[2] This is difficult to do, as seemingly irrelevant feature of a system may be used by to break the wanted property. A notable example of this in security are *side-channel attacks* [Koc96], which use timing or power-consumption information to break apparently secure cryptographic primitives and protocols.

message m to any agent who does not know the secret key k.[3] But anybody in possession of the key k should be able to retrieve the plain-text m from the cipher-text, using the decryption function sdec. That is, we must have the following algebraic property:

$$\mathsf{sdec}(\mathsf{senc}(m, \mathsf{k}), \mathsf{k}) = m$$

The equation above models the functional correctness of the symmetric encryption. Modeling its security is much harder, and depends on the class of attacker considered. We will say more on that point later. Modern symmetric encryptions are either build using a block-cipher, such as the *Advanced Encryption Standard* (AES) [DR02], or are stream ciphers (e.g. ChaCha20 [Ber08, NL15]).

Asymmetric Encryption The idea of asymmetric encryption is due to Diffie and Hellman [DH76] in 1976. It is motivated by the observation that, when using symmetric encryption, a user must have one different secret key for every person he may wish to communicate with.[4] One can avoid this by using two different keys, an *public key* pk, used to encrypt messages, and a *private key* sk used to decrypt them. We then have an encryption function $\{\_\}_\_$, which takes has argument a message m and a public key pk, and returns a cipher-text $\{m\}_{\mathsf{pk}}$. The decryption function dec takes as input a cipher-text and the secret key sk, and returns the plain-text m. That is, if pk and sk are a matching public/private key pair, then:

$$\mathsf{dec}(\{m\}_{\mathsf{pk}}, \mathsf{sk}) = m$$

The knowledge of the public key pk should not be of any help to decrypt a cipher-text $\{m\}_{\mathsf{pk}}$. Therefore, as indicated by its name, it can safely be made public. Anybody can then use it to encrypt messages, which can only be decrypted by the owner of the corresponding secret key.

Cryptographic Hash Function A *cryptographic hash function* is a function that maps a message m of any length to a value of fixed length, called the *hash* of m, which should leak no information about m. Because the co-domain of a hash function is finite and its domain is infinite, it is not injective. This implies that there exist distinct messages with identical hashes, called collisions. While collision exists, we require that they are difficult to find in practice. In particular, this means that the co-domain of a hash function must be large enough to ensure that the probability to find a collision through a brute-force search is very low. We may even make a stronger assumption, and ask that the hash function behaves as a random function: it should be computationally infeasible to distinguish it from a truly random function. An example of a modern cryptographic hash function is Keccak [BDPA14], which won the SHA3 standardization competition.

We will actually consider a variant of this, which are keyed hash function. A keyed hash function H takes a key k as additional input, and returns a hash $\mathsf{H}(m, \mathsf{k})$. This is used to build a *Message Authentication Code* (MAC). A MAC function attaches to a message an authentication code generated using a secret key k. This authentication code can be used by anybody knowing the key k to verify that the message has not be tampered with.

1.2.2 The AKA$^-$ Protocol

We now describe the protocol depicted in Figure 1.1. This is a very simplified version of the 5G-AKA protocol, with no re-synchronization mechanism. A quick word on notations: pairs are represented using angled bracket, e.g. $\langle a\,,\, b\rangle$; and the i-th component of a pair can be retrieved using the i-th projection function π_i (for example, we every $i \in \{1, 2\}$, we have that $\pi_i(\langle a_1\,,\, a_2\rangle) = a_i$).

The Setting In this protocol, a user A tries to establish a shared and authenticated key with its service provider. The user and the service provider both store in memory a shared symmetric long-term secret key k. The service provider has a secret key sk_{N}, and the user stores the corresponding public key pk_{N}. An important feature of the AKA$^-$ protocol is that a message that has already been accepted by the user must not be accepted again by a future session. This is done using a sequence number SQN. This sequence number is an integer value which is attached to the messages of the service provider, and is

[3] Except the plain-text length, which cannot be hidden if one wants to be able to encrypt messages of arbitrary length.
[4] E.g., this would require web browsers to store thousands of keys, which would pose a major key management problem.

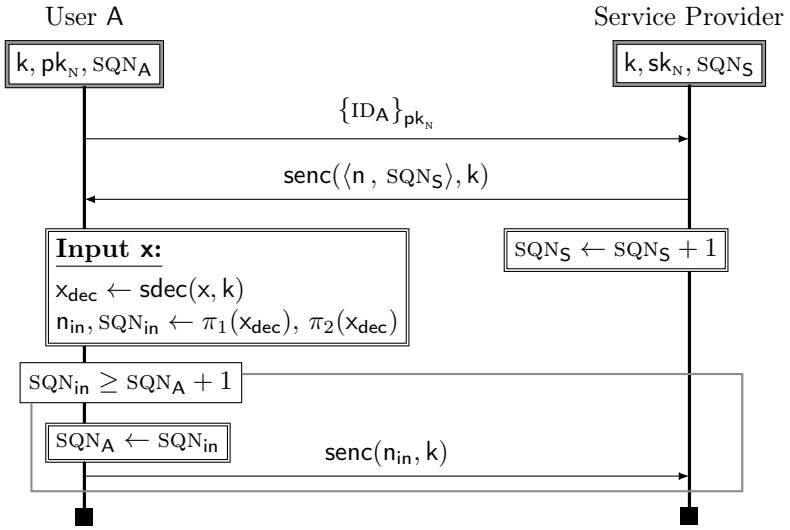


Figure 1.1: The AKA$^-$ Protocol.

incremented by the user after each successful completion of the protocol. By incrementing the sequence number when it accepts a message, the user ensures that this message cannot be accepted again, which prevents messages of the protocol to be replayed by an adversary. The value of the sequence number must be tracked by both the user and the service provider. Therefore, there are two different sequence numbers, the service provider sequence number SQN_S, and the user sequence number SQN_A. Because the sequence numbers must be tracked by the agents, the AKA$^-$ protocol is a *stateful* protocol.

The Protocol The AKA$^-$ protocol is a three-message protocol. The user initiates the protocol by sending to its service provider the asymmetric encryption of its identity ID_A using the public key pk_N. When receiving this message, the service provider retrieves the identity using the secret key sk_N. It then computes its answer, which is the symmetric encryption of a pair, using the secret key k. The first component of the pair is a challenge n, which the service provider samples uniformly at random among bit-strings of length η,[5] and the second component is the current value of the service provider sequence number SQN_S. After sending this message, the service provider updates its sequence number by incrementing it by one.

When it receives a message x from the network, the user starts by decrypting it using the secret key k, and stores the result in x_dec. Then the user retrieves the challenge n and the service provider sequence number SQN_S from x_dec using, respectively, the first and second projection of x_dec. At that point, the user verifies that the sequence number SQN_S it received has not been accepted before. To do this, the user checks that $\mathrm{SQN}_\mathsf{S} \geq \mathrm{SQN}_\mathsf{A} + 1$ (morally, SQN_A stores the highest sequence number accepted thus far). If the test succeed, the user authenticated the service provider. Then, it updates the sequence number SQN_A by setting it to the value SQN_S received from the network, and sends back the message $\mathsf{senc}(\mathsf{n}, \mathsf{k})$. This proves to the service provider that the user knows the key k. Finally, the user and the service provider can both compute a session key from the challenge n and the long-term secret key k, using a key-derivation mechanism which we do not describe here.

1.3 Security Properties

Before formally verifying such a protocol, there are some modeling issues that must be addressed. Mainly, we need to decide how security is expressed. Basically, there are two components to this problem: we need to state what must not happen during the execution of a given protocol (*the security property*), and against what class of adversaries. There are roughly two classes of security properties, trace properties

[5] η is called the *security parameter*. Larger values of η yield a better security.

and equivalence properties. Trace properties are simpler, and allow to express things like weak secrecy or authentication. Equivalence properties are more complex, and are used, e.g., to state that a protocol has some privacy properties (such as anonymity or unlinkability). We discuss and compare the different classes of attackers later.

Trace Properties We call *trace property* any statement about a single execution of a protocol at a time. A simple class of such properties is the class of *reachability properties*, which states that no execution of a given system reaches a bad state. This is a very studied class of properties in the area of formal verification. *Weak secrecy* is an example of reachability properties: informally, a value s in a protocol P (typically a key or a random nonce) is weakly secret if any execution of P followed by a guess $\mathsf{s}_{\mathsf{guess}}$ of s by the adversary is such that $\mathsf{s} \neq \mathsf{s}_{\mathsf{guess}}$. That is, the bad state is the event $\mathsf{s} = \mathsf{s}_{\mathsf{guess}}$, and weak secrecy holds if for any execution of P, no adversary can guess the value of s. Remark that the adversary may be able to guess a portion of s, for example half of it. We only know that he cannot get the full value.

There are trace properties that cannot be directly expressed as reachability properties. An example of such properties that are used in security are *correspondence properties* [WL93], which are of the form:

In any execution of P, if event A occurs, then event B occurred before it.

Authentication is modeled by a correspondence property. For example, consider a protocol between some users $\mathsf{U}_1, \mathsf{U}_2, \ldots, \mathsf{U}_n$ and a server S. Whenever a user U_i tries to authenticate himself to the server by running an authentication protocol, he emits an *event* $\mathsf{start\text{-}auth}_i$.[6] Moreover, whenever the server S completes a session of the authentication protocol with what he believed was user U_j, he emits an event $\mathsf{end\text{-}auth}_j$. Then the protocol provides authentication if, for any execution, if $\mathsf{end\text{-}auth}_j$ occurs at some point in time, then the event $\mathsf{start\text{-}auth}_j$ must occur before it. In natural language, if the server believes he authenticated some user U_j, then this user must have tried to authenticate himself to the server. In that case, the attacker cannot make the server believe that he authenticated U_j if this user never tried to established a connection to the server. One can have more refined properties than the one presented above, e.g. by attaching session numbers or protocol challenges to events. See [Low97] for a comparison of different authentication properties.

Equivalence Properties Some important security properties cannot be expressed as trace properties. Anonymity is a example of such a property. Basically, a protocol P is anonymity preserving if an adversary cannot know if a given agent A was involved in an execution of P. In other words, the adversary cannot distinguish between the scenario where P was executed by A and the scenario where P was executed by some other agent B. This is fundamentally a property about *two executions*.

Such properties are called *equivalence* or *indistinguishability* properties, and state that two different scenarios are indistinguishable to the adversary. Equivalence properties are more expressive than trace properties, but are more complex to verify. There are many examples of indistinguishability properties in security. All privacy properties are indistinguishability properties, e.g. *unlinkability* [Vau07] states that the adversary cannot find any links between two executions of a protocol by the same agent. *Strong secrecy* [Bla04] of a value v in P expresses the fact that the adversary cannot learn anything about v during the execution of P. To model this, we ask that no adversary can distinguish between a scenario where we leak the value v after completing the execution of the protocol P, and a scenario where we leak a different random value v'. If the adversary can learn a single bit of information on v, such as the fact that it satisfies some properties, then he could distinguish between the two scenarios: on the former scenario, the property would always hold, while on the latter, it would only hold with probability one-half.

1.4 Attacker Models

The goal of formal verification of security protocols is to prove that a protocol satisfies a security property against any adversary in some given class. Of course, different classes of adversaries yield different security guarantees. On the one hand, we wish to show that a protocol is secure against a class of adversaries as

[6] An event in an element of the execution of a protocol which is used to express some properties, but which is not visible to the adversary.

large as possible. On the other hand, if we consider a restricted class of adversaries, we may be able to use some proof techniques, which can allow for machine-checked proof, or even automatic proof search.

1.4.1 Symbolic Model

The *symbolic model*, introduced by Dolev and Yao [DY83] in 1983, tries to cover logical attacks. By logical attack, we mean an attack that does not try to break the cryptographic primitives used in the encryption, but instead uses flaws in the logical control-flow of the protocol. These attacks are the worst possible attacks, as they are independent of the implementation details, and are reliable.

While there are several ways of modeling such attackers and protocols, the applied pi-calculus [ABF18] is arguably the most prominent. In the applied pi-calculus, messages are represented by terms in some formal term algebra, which are build using constants, names (which model random challenges or session numbers), and function symbols such as the pair $\langle \_ , \_ \rangle$ or the encryption $\mathsf{senc}(\_, \_)$. A protocol is an element of a protocol algebra, and can typically do inputs and outputs of terms, conditional tests, parallel composition and replication.

The Adversary Since the adversary has complete control of the network, he knows all messages that where outputted, and chooses what messages are sent to the agents, with some restrictions: to send a message m to an agent, the adversary must be able to obtain m from his current knowledge (the sequence of all messages outputted since the protocol started), using some fixed set of capabilities which have been granted to him. These capabilities are expressed through rules, which can, for example, be given using deduction rules. E.g, given a pair $\langle a , b \rangle$, the adversary can retrieve the first and second component of the pair. Conversely, if he knows a and b then he knows the pair $\langle a , b \rangle$. Or, if the adversary knows an encryption $\mathsf{senc}(m, \mathsf{k})$ and the associated key k, he can get the plain-text m. Formally:

$$\frac{\langle a , b \rangle}{a} \qquad \frac{\langle a , b \rangle}{b} \qquad \frac{a \quad b}{\langle a , b \rangle} \qquad \frac{\mathsf{senc}(m, \mathsf{sk}) \quad \mathsf{sk}}{m}$$

The adversary can only apply the rules that are given to him. This means that the verifier must be careful to include all algebraic properties of the primitives used in its protocol. If such a property is forgotten, attacks may be missed.

Tools This model is very amenable to automatic verification of security properties. Since security in the symbolic model is undecidable [Hüt02], automated tools sometimes fail to find a proof of security or an attack, or are restricted to a decidable subset of protocols and properties. There are several automated tools for both trace or equivalence properties, based on various techniques such as Horn clause resolution (e.g. ProVerif [Bla]), multi-set rewriting (e.g. Tamarin [MSCB13]) or constraint solving (e.g. Deepsec [CKR18]).

1.4.2 Computational Model

Another attacker model, closer to a real world attacker, is the *computational attacker* model introduced by [GM84] in 1984. In this model, we do not restrict the attacker to a fixed set of operations: we do not try to "guess" which operations the adversary uses in an attack. Instead, the adversary can perform any probabilistic polynomial time computation. Of course, this offers stronger guarantees than the symbolic model, at the cost of more intricate model.

The Adversary More formally, messages are bit-strings, as in any real-world implementations. Random challenges of the protocol are sampled uniformly among bit-strings of some given length (usually in $\{0,1\}^\eta$, where η is the security parameter), and protocol agents and the adversary are (interactive) *probabilistic polynomial-time Turing machines* (PPTM*s*). Security properties are usually expressed through a game, where an adversary $\mathcal{A}$ interacts with the protocol through an oracle. There are typically two scenarios, i.e. two oracles $\mathcal{O}_0, \mathcal{O}_1$ that the adversary may interact with. Eventually, $\mathcal{A}$ tries to guess in what scenario he is by outputting a bit b. The advantage of the adversary is the probability that he guessed correctly:

$$\mathrm{Adv}_{\mathcal{A}}(\eta) \;=\; \left|\mathbf{Pr}(\mathcal{A}^{\mathcal{O}_1}(1^\eta) = 1) - \mathbf{Pr}(\mathcal{A}^{\mathcal{O}_0}(1^\eta) = 1)\right| \;=\; \left|2 \,.\, \mathbf{Pr}(\mathcal{A}^{\mathcal{O}_b}(1^\eta) = b) - 1\right|$$

This probability is a function of η, the security parameter. In the asymptotic security setting, we say that a protocol is secure if, for every adversary $\mathcal{A}$, the advantage $\mathrm{Adv}_{\mathcal{A}}(\eta)$ is negligible in η, where a function is negligible if it is asymptotically smaller than the inverse of any polynomials. In the concrete security setting [BKR00], we try to obtain precise upper-bounds on the advantage of $\mathcal{A}$, as function of $\mathcal{A}$ running time and its advantages against breaking cryptographic primitives of the protocol.

Security Proofs In the computational model, proofs are usually not unconditional, but rely on *computational hardness hypotheses*, which assume that some problems are not solvable in probabilistic polynomial-time. When trying to prove that a cryptographic *primitive* is secure, we usually rely on low-level hardness hypotheses such as the Computational or Decisional Diffie-Hellman assumptions, or the Discrete Logarithm assumption in some finite groups. When proving a cryptographic *protocol*, we use higher-level assumptions, like the Indistinguishability against Chosen Plain-text Attack assumption (IND-CPA). Basically, an asymmetric encryptions is IND-CPA if no adversary can distinguish between the encryptions of two messages of the same length, even if we let the adversary choose the messages.

We then show that a protocol is secure, assuming that no adversary can solve efficiently some problems. Conditional security proofs like this are reductions, as in complexity theory: given an adversary $\mathcal{A}$ breaking the security property, we build an adversary $\mathcal{B}$ breaking the hardness assumption. Such proofs are often long, complex and error-prone, even though game-hoping techniques [Sho04] allow to alleviate some of the complexity by splitting the proof in successive small changes to the game.

Tools As expected, tools in the computational model are less automatic than tools in the symbolic model. Nonetheless, there exists a tool, CryptoVerif [Bla08], which performs (semi-)automatic cryptographic game transformations. Also, there are some interactive formal verification tools in the computational model, such as EasyCrypt [BGHB11], an interactive theorem prover relying on the *probabilistic Relational Hoare Logic*, and F^* [BBD+17b], a high-level functional programming language with build-in support for verification. These two tools are interactive, with little support for automation.

1.4.3 Computational Soundness

There exists a line of research, due to Abadi and Rogaway [AR02], which tries to bridge the two approaches. This approach, called the *computational soundness approach*, consists in proving that, under some conditions, the security of a protocol in the symbolic model implies the security of the protocol in the computational model. In other words, the symbolic adversary is at least as strong as the computational adversary. While the first computational soundness result were against a passive adversary [AR02, BCK09], there are many computational soundness results against active adversaries, e.g. [MW04] for encryptions, [JLM05, BMU12, BMR14] for signatures, or [CKKW06] for hash functions in the Random Oracle Model. A survey of computational soundness results can be found in [CKW11].

There are several problems with this approach. First, these results make strong implementation and cryptographic assumptions. For example, they usually have a parsing assumption, which assume that all functions add unambiguous tags to their outputs. Then, there are some impossibility results for the computational soundness method, e.g. for the xor operator [BP05, Unr10] or for one-way hash functions [BPW06]. Finally, the approach is not modular, as each result is for a specific set of cryptographic primitives, cryptographic hypothesis and implementation assumptions. If we want to add support for another primitive, or to change an implementation assumption, we need to prove a new theorem.[7]

One can wonder why such results are so complicated and have such a limited scope. In his PhD thesis [Sce15], Scerri suggests that this is because the symbolic adversary is defined through what he can do, while the computational adversary is defined through what he cannot do (through the cryptographic games that he cannot win). In other word, a symbolic adversary is defined through a smallest fix-point, and a computational adversary through a greatest fix-point. A computational soundness result shows that the symbolic adversary contains the computational one. This does not leave any leeway: if one wishes to add a new cryptographic primitive, the symbolic adversary must be extended to ensure that he captures all possible attacks against the new primitive. A way of avoiding these problems is to have a class of adversaries which is both *stronger* than the computational adversary, and defined through a

[7] Nonetheless, some modularity can be achieved for computational soundness results. Indeed, in [CW11], the authors define a computational soundness notion, called deduction soundness, that is extendable: basically, if a symbolic model is shown to be deduction sound, then it can safely be extended with public data structure and asymmetric encryption.

greatest fix-point. This way, when we add new cryptographic primitives and hypotheses, we just need to add restrictions (corresponding to the new primitives) on the symbolic adversary. Because we consider a greatest fix-point, such restrictions can be designed independently from each other.

1.4.4 The Bana-Comon Model

This is the idea behind the Bana-Comon model [BC12], also known as the Computationally Complete Symbolic Attacker model. This is a first-order logic, in which messages as represented by terms. But instead of specifying the adversary by what he can do, as in the Dolev-Yao model, the adversary is defined negatively by what he *cannot do*, using a set of first-order axioms Ax. These axioms may reflect structural properties of the logic, implementation assumptions on the primitives (e.g. functional correctness), or cryptographic hypotheses on the primitives. We require that these axioms are computationally valid, under some cryptographic assumptions. More precisely, we identify the subset of the first-order models of our logic which correspond to computational models (basically, the interpretation domain is the set of polynomial-time probabilistic Turing machines); and we require that the axioms Ax are valid in any computational model where the cryptographic primitives satisfy our cryptographic assumptions.

Then, given a protocol and a security property, we can compute a formula ψ expressing the security of the protocol. Showing the unsatisfiability of the conjunction of the axioms Ax and the negation of ψ entails the security of the protocol.[8] Indeed, we know that there exists no adversary that can simultaneously satisfy the axioms Ax and break the security property. Since our axioms are computationally valid, we deduce that the security property ψ holds in all computational models: the protocol is secure.

Conversely, if the conjunction of the axioms and the negation of the security property are satisfiable, it means that there exists an attacker breaking the security property and satisfying all the axioms. But because we only required computational *soundness* of the axioms, and not computational *completeness*, this attacker may not be a computational attacker.

Comparison with Other Models This model has several advantages over the three other approaches presented so far. First, it gives strong security guarantees, as security in the Bana-Comon model implies computational security. Second, this model is simpler than the computational model: there is no probabilities and no security games, only first-order formulas. Third, because the security of a protocol amounts to the unsatisfiability of a set of first-order formulas, we believe that this model is more amenable to automatic verification than the computational model. Fourth, it does not allow for *implicit assumptions*. For example, if the security of a protocol relies on the fact that the first projection of a nonce can (almost) never be confused with an agent's name, then we need to add an axiom stating that this is the case. Otherwise, the security proof cannot be completed. Proving a protocol in the Bana-Comon model requires to make precise and *explicit* assumptions on the protocol implementation. Finally, it is more modular than the computational soundness approach, as axioms for cryptographic hypothesis can be designed and proved valid independently from each other.

A inherent drawback of the Bana-Comon approach is that it is only valid for protocols with a finite number of sessions: we may only consider protocols with no unbounded replication. Still, it is possible to show that a protocol is secure for any *constant but arbitrarily large* number of sessions. E.g. if ψ_n is a formula encoding the security of n sessions of a protocol, then it is sufficient to show that for every n, $\mathsf{Ax} \wedge \neg\psi_n$ is unsatisfiable. Typically, such a proof is done by induction over n. Note that security for any constant number of sessions does not imply security for a number of sessions that depends on the security parameter. Nonetheless, most attacks do not require polynomially-many sessions. Another drawback of the Bana-Comon approach is that it is not quantitative. Security is asymptotic, and we do not obtain an upper-bound on the advantage of an adversary, as in the concrete security approach [BKR00]. This is not an inherent restriction of the logic, as we believe that an upper-bound on the adversary advantage can be inferred from a proof. But bounds obtained using this method would probably be far from optimal, and by consequence of little use.

Trace Properties The Bana-Comon model introduced in [BC12] is for trace properties only. This model has been used in [BAS12] to prove the security of the Needham-Schroeder-Lowe protocol [Low95]

[8] In the Bana-Comon model of [BC12], we must also add some ground formulas encoding the conditions under which the protocol is executable.

(NSL). In [BHO13], the authors gives an alternative and simpler semantics for the logic based on Kripke structures, and use it to prove the correction of key-usability axioms (e.g. for KDM-CCA$_2$). Finally, in [CCS13] the authors study the automation of proofs in this model. They show that the problem of checking the satisfiability of a set of clauses corresponding to axioms the Bana-Comon model for trace properties is decidable in polynomial-time, using a Horn clause saturation procedure. A variant of this decision procedure is implemented in the SCARY tool [Sce15], which has been used to prove the security of standard cryptographic protocols (e.g. NSL) for a small number of sessions, and found a new attack on Andrew Secure RPC protocol [Sat89].

Equivalence Properties Many crucial security properties, such as strong secrecy or privacy, are inherently equivalence properties, and therefore out of the scope of [BC12]. To be able to prove such properties, Bana and Comon proposed a new model for equivalence properties in [BCL14]. This logic has only one predicate symbol $\sim$, which stands for the computational indistinguishability relation. That is, given two terms u and v, which are symbolic representations of bit-string distributions, the formula $u \sim v$ holds if no adversary can distinguish between the two distributions, except with a negligible advantage. While this model relies on the same ideas as in [BC12] (a symbolic representation of protocol messages and an axiomatization of what the adversary cannot do), its formulas and axioms are very different. We believe this logic is simpler than the trace logic: it has a simple first-order semantics, only one predicate, and more intuitive axioms. Nonetheless, proofs of non-toy protocols in the Bana-Comon model for equivalence properties are challenging, as we will see in Chapter 4.

In [BCL14], Bana and Comon designed a small set of axioms, including axioms for IND-CPA and *Key Privacy* (KP) cryptographic assumptions. They illustrated their method on a simple example, by showing the privacy of the Private Authentication protocol with a decoy message [AF04].

1.5 Limitations of the State of the Art

Ideally, we would like to have a model that provides strong security guarantees and that is amenable to automated proof search. Moreover, this model should support equivalence properties, as these are more expressive than trace properties.

As we saw, the two oldest and most established attacker models, namely the symbolic and computational models, fail to achieve these properties simultaneously. Indeed, while there exist several tools to automate proofs in the symbolic model, its security guarantees are not strong enough, as it only consider attacks that can be executed using the capabilities that the prover granted to the adversary. As for the computational model, it is indeed more realistic and offers strong guarantees, but the level of proof automation achieved by current tools in this model is not satisfactory, either because they often fail to automatically find proofs, or because they require users to do extensive manual proofs.

We mentioned the computational soundness approach, which tries to get the best of both worlds by proving that, in some cases, security in the symbolic model implies computational security. This allows to use an automated tool in the symbolic model to prove that a protocol is computationally secure. Unfortunately, computational soundness results come at a high cost, as they usually make strong assumptions on the protocol implementations and on the cryptographic primitives, which limit their applicability. Moreover, this approach scales poorly because of its lack of modularity: to add a new cryptographic primitive, one often need to show a new computational soundness result, with new assumptions.

We presented an alternative approach, due to Bana and Comon. Because we are interested in equivalence properties, we discard the Bana-Comon model for trace properties [BC12], and focus on the equivalence model in [BCL14]. This model looks promising: first, security in the Bana-Comon model implies computational security; then, this approach is modular, since axioms can be designed and proved valid independently for every cryptographic primitive; finally, because this is a first-order logic, it may be amenable to automated deduction techniques, as it turned out to be the case in the Bana-Comon model for trace properties [CCS13]. But, when this thesis started, the Bana-Comon model usefulness remained to be shown: there was no case study of non-toy protocols, only a small set of axioms had been designed, and there was no support for proof automation.

Related Works Since the Start of This Thesis In parallel to this thesis, some works have been done using the Bana-Comon equivalence model. These works address some of the concerns we had on

the model, in particular about its applicability and its small set of axioms. In [BC16], the authors design axioms for several cryptographic hypothesis: asymmetric encryption (IND-CCA$_1$ and IND-CCA$_2$), signatures (EUF-CMA) and for the Decisional Diffie-Hellman assumption. With these axioms, they prove that the Diffie-Hellman key-exchange provides real-or-random secrecy [AFP05] of the shared key. They also prove several properties of the NSL protocol, including authentication and real-or-random secrecy of the shared nonces. In [SS16], the authors design and prove secure a key wrapping API. Interestingly, their proof is modular in the choice of the symmetric encryption used in the wrapping mechanism: the authors design intermediate axioms for the wrapping mechanism, prove the security of the wrapping API using these axioms, and show that both *randomized* and *deterministic* symmetric encryption schemes satisfy the intermediate axioms. This is a nice benefit of the Bana-Comon approach. Finally, in [BCE18], the authors analyze the vote privacy of the FOO voting protocol [FOO92]. First, they design axioms for blind signatures [Cha82]. Second, they found new attacks on the privacy of the FOO protocol, when the candidate identities or the messages signatures are of different lengths. These are typical examples of implicit implementation assumptions that can be found using the Bana-Comon approach. Then, under the proper assumptions, they prove that the FOO protocol provides vote privacy.

1.6 Contributions

The goal of this thesis was to develop techniques to formally verify equivalence properties of cryptographic protocols. Moreover, we wanted a method that provides strong security guarantees while being amenable to automated deduction techniques. The Bana-Comon model for equivalence properties seemed to be a good candidate, but its applicability to real-world protocols remained to be shown, and there was no support for proof automation in this model. We tried to address these two shortcomings in this thesis.

1.6.1 RFID Protocols

First, we completed the case study of two RFID authentication protocols, KCL [KCL07] and LAK [LAK06], in which a reader is trying to authenticate a tag. An RFID tag is a small cryptographic device which has low computing capabilities. Therefore, tags do not rely on advanced and complex cryptographic primitives to achieve authentication. e.g. the two examples we consider use only hash functions, the xor operator and pairs. This makes them good candidates for a first application of the Bana-Comon approach, as we only need to design axioms for a small set of functions. In particular, we designed axioms for hash functions and for the xor operator.[9] Axioms for the xor include a uniform distribution axiom, as well as functional correctness axioms for associativity, commutativity, unit rule and nilpotence. The hash functions axioms are more interesting, as they depend on the cryptographic assumptions we make. We designed two axiom schemas, for the *Collision Resistance* and *Pseudo-Random Function* assumptions.

Case Study As RFID tags may be carried all the time by their users, it is crucial that they provide some form of privacy. There exists many definitions of privacy, e.g. [HPVP11, Vau07, JW09]. We chose the notion of Privacy from Juels and Weis [JW09] because it is simple and game-based, and translated it into the Bana-Comon model. Using this, we studied the KCL and LAK protocols. These two protocols are known to be insecure (privacy attacks can be found in [VDR08, HBD16]). Therefore, we designed fixed versions of these protocols, KCL$^+$ and LAK$^+$. Then, depending on the implementation assumptions we make (e.g. the cryptographic hypotheses on the hash function), we either provide an attack or a security proof. Specifically, under the appropriate assumptions, we prove Privacy for two tags and six interactions for LAK$^+$, and for any number of interactions for KCL$^+$. The latter proof is by induction on the number of interactions. We reuse this proof technique later on a much more complex protocol, the AKA$^+$ protocol.

1.6.2 The AKA Protocol

Next, we studied the AKA protocol. More precisely, we studied its 5G version, the 5G-AKA protocol, as it is described in the 3GPP draft [TS318]. We presented a simplified version of this protocol in Section 1.2. Mobile phone users often carry their phone with them everywhere, and could be easily and thoroughly

[9] Remark that the Bana-Comon model can handle the xor operator without difficulties, contrary to the symbolic model [BP05].

tracked through them. Therefore, it is important that mobile network protocols, such as the 5G-AKA protocol, provide some privacy. Previous versions of this protocol, the 3G-AKA and 4G-AKA protocol, are vulnerable to a famous linkability attack, the IMSI-catcher attack [Str07]. This is a major attack, as it is reliable (the attack always work), cheap to deploy, and large scale (every mobile phone in range of a rogue antenna can be tracked). The 5G version of the AKA protocol allows the mobile phone to hide its permanent identity using an asymmetric encryption scheme, which prevents the IMSI catcher attack.[10] But this is not enough for privacy: we show that several known privacy attacks against the previous versions of the protocol still apply to the 5G version [FOR16, AMR+12, BHP+17], except for the IMSI catcher attack. While studying these attacks, we found a privacy attack against another protocol, the PRIV-AKA protocol. This protocol is a significantly modified version of AKA, which is designed and claimed unlinkable in [FOR16]. Our attack is new, and consists in permanently de-synchronizing the mobile phone from its service provider. The fact that a user is de-synchronized can be detected by an attacker, which leads to a unlinkability attack.

Fixing the Protocol We then proposed a fixed version of the 5G-AKA protocol, called AKA$^+$. We designed this protocol to provide better privacy guarantees than the 5G-AKA protocol, while using the same cryptographic primitives and satisfying the same constraints (as much as possible). We then study its privacy. Here, we do not use Juel and Weis's Privacy, but the unlinkability property, which is a stronger notion of privacy inspired from [HPVP11] and Vaudenay's unlinkability [Vau07]. Our protocol does not satisfy this property: there is an attack. Actually, we believe that under the design constraint of the 5G-AKA protocol, unlinkability cannot be achieved. Still, we are able to prove that our protocol provides a weaker property, called σ-unlinkability. This is a new property which we designed. Basically, a protocol is σ-unlinkability if it is unlinkable for *some* scenarios of the standard unlinkability. This property is parametric in the set of scenarios that must be considered. If this set is empty, we have nothing to prove, and the protocol provides no privacy guarantees. If this set contains all possible scenarios, then the protocol satisfies the standard unlinkability property. By considering sets of scenarios between these two extremes, we can have a fine-grained quantification of a protocol's privacy. As in the RFID case study, we express this property using labelled transition systems.

Cryptographic Assumptions and Axioms To prove that the AKA$^+$ protocol satisfies the σ-unlinkability property, we have to design new axioms. First, the AKA$^+$ protocol has to assume that the hash functions are *jointly pseudo random functions*, i.e. that they are simultaneously computationally indistinguishable from random functions. Therefore, we introduce new axioms for the joint PRF assumptions, as well as the joint *Collision Resistance* and joint *Unforgeability against Chosen-Message Attacks* assumptions. We also design axiom schemas for the standard *Unforgeability against Chosen-Message Attacks* assumption.

Security Proofs Using these axioms, we prove that the 5G-AKA protocol satisfies the σ-unlinkability property. This proof is for any number of agents and sessions which does not depend on the security parameter. As for the KCL$^+$ protocol, this proof is by induction on the number of interactions between the adversary and the agents, but is much more involved. First, we show several *necessary acceptance conditions*. These are correspondence properties giving necessary conditions for a message to be accepted at some point of the protocol execution. Typically, such a property states that a message can only be accepted if the adversary honestly forwarded some messages. Using these conditions, we prove that the AKA$^+$ protocol provides mutual authentication between the mobile phone and the service provider. Then, we refine the acceptance conditions to obtain *acceptance characterizations*, i.e. necessary and *sufficient* conditions for a message to be accepted. Finally, we prove that the AKA$^+$ protocol is σ-unlinkable.

1.6.3 Deciding Indistinguishability

Our last contribution is the design of a complete and terminating strategy for a fragment of the Bana-Comon indistinguishability logic. We identify a set of axioms Ax which is both expressive enough to complete proofs of concrete formulas, and computationally sound under the appropriate cryptographic

[10] Unfortunately, the 3GPP consortium made usage of an asymmetric encryption optional. Therefore, the next generation of mobile phones may continue to be vulnerable to the IMSI-catcher attack.

assumption (IND-CCA$_2$). Then, we show that the satisfiability problem of this fragment is decidable. More precisely, given a ground formula $\vec{u} \sim \vec{v}$, we can decide whether $\mathsf{Ax} \wedge \vec{u} \not\sim \vec{v}$ is unsatisfiable:

Input: A ground formula $\vec{u} \sim \vec{v}$.
Question: Is $\mathsf{Ax} \wedge \vec{u} \not\sim \vec{v}$ unsatisfiable?

All the axioms in Ax are Horn clauses, therefore to show the unsatisfiability of $\mathsf{Ax} \wedge \vec{u} \not\sim \vec{v}$ we use resolution with a negative strategy (which is complete, see [CL73]). A proof by resolution with a negative strategy can be seen as a proof tree where each node is indexed by the axiom of Ax used at this resolution step. By consequence, we see axioms in Ax as inference rules and look for a derivation of the goal $\vec{u} \sim \vec{v}$. Our proof search incrementally builds a partial proof tree whose root is $\vec{u} \sim \vec{v}$, trying to close all branches by applying a unitary inference rule (i.e. a rule with no premise). The only unitary axioms in Ax are cryptographic axioms, which reflect the cryptographic hypothesis made on the security primitives (e.g. that the encryption function hides its content, as in $\{0\}_{\mathsf{pk}} \sim \{1\}_{\mathsf{pk}}$). We use a specialized handling of equalities in the axioms: we have a set of equalities R, which includes functional correctness equalities, and properties of the $\mathsf{if\_then\_else\_}$, such as:

$$\pi_1(\langle x\,,\, y\rangle) = x \qquad\qquad \mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ y)\ \mathsf{else}\ z = \mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ z$$

We then introduce its congruence closure $=_R$, and have a rewriting axiom:

$$\frac{t \sim u}{s \sim u}\ R \quad \text{whenever} \quad s =_R t$$

Actually, since $s =_R t$ is a side-condition ($=_R$ is not a predicate of the logic), this is not a single axiom, but a recursive countable set of axioms (i.e. an axiom schema). This axiom is problematic, as it allows to rewrite a term s into any R-equal term t, which can be arbitrarily large. This is the main obstacle to achieve decidability which we had to overcome. We sketch, in a high-level fashion, how we did it. First, we design a particular ordered strategy for our logic. An ordered strategy restrict the proof search space, by requiring that inference rules are applied in a specific order. We show that our ordered strategy is complete through several commutation lemmas. Our strategy ensures that R rules all occur at the beginning of the proof. Moreover, the other axioms are such that bounding the initial R applications bound the rest of the derivation. To bound the initial rewritings, we identify several proof cuts which introduce unbounded sub-terms, and find proof cut eliminations to remove them. Unfortunately, most of these proof cut eliminations are not *local* rewriting of the proof-tree, but are *global*, which makes the proof cut eliminations lemmas non-trivial. Finally, we prove that cut-free proofs are of bounded size. This yields a decision procedure for our satisfiability problem.

1.7 Outline of the Thesis

We give the outline of this thesis. We present the Bana-Comon indistinguishability logic in Chapter 2. The model we used is basically the model of [BCL14], with a small extension to allow for protocols with an arbitrary number of sessions (using infinite LTS). In this chapter, we also present most of the axioms we designed during this thesis. In Chapter 3, we present our RFID protocols case study. The larger case study of the AKA protocol is in Chapter 4. The most theoretical part of this thesis, the decision procedure, is described in Chapter 5. Finally, we conclude in Chapter 6.

Chapter 2

The Model

In this chapter, we present our version of the Bana-Comon first-order logic for indistinguishability. We give a general way of modeling security protocols and properties using labelled transition systems, and show how the computational semantics of these protocols relate to the logic. Basically, given an interpretation $\mathcal{M}_{\mathsf{c}}$ of functions as Turing machines, a protocol is secure in $\mathcal{M}_{\mathsf{c}}$ if and only if a countable set of formulas $(\phi_i)_{i\in\mathbb{N}}$ of the logic are valid in $\mathcal{M}_{\mathsf{c}}$ (seen as a first-order model). To use this, we design a set of formulas Ax, called *axioms*, which state what the adversary *cannot* do. Then, for every i, we show that the conjunction of the axioms Ax and the negation of ϕ_i is unsatisfiable. We deduce that the protocol is secure for any implementation $\mathcal{M}_{\mathsf{c}}$ of the protocol's functions satisfying the axioms Ax.

The Bana-Comon Logic The Bana-Comon logic for indistinguishability was introduced in [BCL14]. This is a sorted first-order logic, in which terms represent messages of the protocol sent over the network. For example, the term $\langle \mathsf{A}, \mathsf{n}\rangle$ represents a message which comprises two parts: an agent name A (which is a constant function symbol), and a name n (taken in the set of names $\mathcal{N}$), representing a random uniform sampling in $\{0,1\}^\eta$ (where η is the security parameter). A key idea in the logic is to use special adversarial function symbols $g_0, g_1, \cdots \in \mathcal{G}$ to represent the adversary's inputs. Morally, these function symbols are uninterpreted, which allows to model the fact that the adversary can do any polynomial-time computation. These adversarial function symbols receive as input the current knowledge of the adversary ϕ (the frame), which is simply the sequence of all messages sent over the network since the protocol started (since messages are modeled by terms, ϕ is a sequence of terms). For example, $g(\langle \mathsf{A}, \mathsf{n}\rangle)$ represents anything the adversary can compute after having intercepted the message $\langle \mathsf{A}, \mathsf{n}\rangle$. More generally, if ϕ is the current frame, then $g(\phi)$ represents any message that can be computed by the adversary at that point of the protocol execution.

In order to be able to represent messages of the protocol by terms, the control-flow of the protocol needs to be internalized in the logic. This is done by encoding tests of the protocol agents by boolean terms, and branching using the $\mathsf{if\_then\_else\_}$ function symbol. For example, imagine a protocol where some agent A behaves as follows: first A waits for a message from the network; then, after receiving a message x, A checks whether this message is equal to some secret value secret; if this is the case, A outputs its identity ID_{A}, otherwise it outputs an constant error message Error. This is modeled by the term:

$$\mathsf{if}\ \mathsf{eq}(g(\phi), \mathsf{secret})\ \mathsf{then}\ \mathrm{ID}_{\mathsf{A}}\ \mathsf{else}\ \mathsf{Error}$$

where $\mathsf{eq}(\_,\_)$ is a function symbol representing the equality check, ID_{A} and Error are constant function symbols and, we recall, $g(\phi)$ is a term representing the input from the network.

Formulas of the logic are built using the usual Boolean connectives and FO quantifiers, and a single predicate, $\sim$, which stands for indistinguishability. The semantics of the logic is the usual first-order semantics: each sort is interpreted as a domain and function symbols and predicates are interpreted as, respectively, functions and subsets of the appropriate domains. Still, since we want to interpret sequences of terms representing executions of protocols, we are particularly interested in *computational models*, in which terms are interpreted as *probabilistic polynomial-time Turing machines* (PPTMs), and $\sim$ is interpreted as computational indistinguishability. Intuitively, given an implementation of the protocol functions $\mathcal{I}$ and an adversary $\mathcal{A}$ (which is a PPTM), we obtain a computational model as follows:

- Protocol function symbols are interpreted using the protocol implementation $\mathcal{I}$. Basically, the implementation $\mathcal{I}$ associates to every protocol function symbol f its implementation $\mathcal{I}(f)$, which is a polynomial-time Turing machine.
- Names in $\mathcal{N}$ are interpreted as uniform random samplings in $\{0,1\}^\eta$.
- Finally, we let the adversary $\mathcal{A}$ choose the interpretation of every adversarial function symbol $g \in \mathcal{G}$. Since $\mathcal{A}$ is polynomially-bounded, the interpretation $\mathcal{A}(g)$ of g is also polynomially-bounded.

Basically, a computational model $\mathcal{M}_\mathsf{c}$ corresponds to the interaction of a given adversary $\mathcal{A}$ with an implementation of the protocol functions $\mathcal{I}$.

Protocols as Labelled Transition Systems As in [BCL14], we do not assume an input language (such as the applied pi-calculus [ABF18]) for protocols. Instead, protocols are defined using a *labelled transition systems* (LTS). The transitions of the LTS correspond to the actions available to the adversary, and the nodes of the LTS record the static component of the protocol state, i.e. the part of the state that does not depend on the random samplings or conditional branching of the protocol agents. Our definition of protocols as LTS differ from [BCL14] in several ways:

- In the original paper by Bana and Comon, protocols are finite LTS, which cannot model protocols with an unbounded number of sessions and agents. This restriction stems from the fact that the Bana-Comon logic cannot soundly model the security of protocols whose number of sessions is a function of the security parameter. Still, using the Bana-Comon logic, it is possible to prove that a protocol is secure for an arbitrary number of sessions, as long as it is independent from the security parameter. Therefore, we model protocols using potentially infinite LTS. To ensure that from any attacker we can extract a winning attacker against a finite fixed trace of the LTS, we require that the LTS is *finitely branching*.
- In [BCL14], the transitions of the LTS are guarded using conditional terms. Bana and Comon then show how to compute, from any protocol P, a protocol $\mathsf{fold(P)}$ where each node has at most one out-going transition with no guard. Basically, the guard checks are pushed inside the protocol message, using a technique similar to the one used in [CB13]. We decided to bypass this step, and not to use guards in the LTS transitions. Of course, messages can still be guarded by including tests directly in the protocol terms (as in a folded protocol).
- The Bana-Comon logic is well-suited to prove stateful protocols, which is something hard to do in other formalisms. By consequence, we chose to include state updates in the LTS transitions. This allows us to easily model protocols such as LAK [LAK06], KCL [KCL07] or AKA [TS318].

Protocol Executions Given a computational model $\mathcal{M}_\mathsf{c}$ interpreting a protocol P's function symbols and the adversarial functions, we define the *computational execution* of P by letting the adversary choose the transition to execute at every step i of the protocol. This is done using special adversarial function symbols $\mathsf{to}_i \in \mathcal{G}$, which models the adversary unknown choice of action.

Similarly, we define the fix-trace execution of P, where we fix the sequence of actions (the action trace) to execute in advance, instead of letting the adversary choose the next action on the fly. Because protocols are finitely branching, and because we only consider attacker against a finite, though arbitrary long, sequences of actions, we can show that the adaptive and fix-trace semantics of protocol are related: there is no winning adversary against P if, for every trace of action τ, there is no winning adversary against the execution of P with actions τ.

Once the action trace τ is fixed, it is very easy to build a formula ϕ_τ of the logic representing the execution of the protocol with trace τ. By consequence, we have reduced the problem of showing the security of a protocol P in a computational model $\mathcal{M}_\mathsf{c}$ to the problem of proving that, for every τ, the formula ϕ_τ is valid in $\mathcal{M}_\mathsf{c}$.

Axioms Of course, any non-trivial protocol will not be secure in any computational model $\mathcal{M}_\mathsf{c}$. E.g., any real-world protocol is probably not secure if the encryption function symbol is interpreted as the function that always returns the plain-text. By consequence, we are going to show that a protocol is secure in some class of models. We do this by restricting the models that have to be considered using axioms Ax, where an axiom is a formula of the logic stating something that the adversary *cannot* do. Axioms are of two kinds:

- *Structural axioms* are properties that are valid in all computational models. For example, the function application axiom:

$$\frac{\vec{u} \sim \vec{v}}{f(\vec{u}) \sim f(\vec{v})}\ \mathsf{FA}$$

states that to show that two terms $f(\vec{u})$ and $f(\vec{v})$ are indistinguishable, it is sufficient to show that the arguments $\vec{u}$ and $\vec{v}$ are indistinguishable.

Since, eventually, we only care about computational models, adding such axioms is always safe.

- *Implementation axioms* are not valid in all computational models, and instead reflect assumptions on the protocol's implementation. For example, the axiom:

$$\{m\}^{\mathsf{n_e}}_{\mathsf{pk}} \sim \{\mathbf{0}(m)\}^{\mathsf{n_e}}_{\mathsf{pk}}$$

states that no adversary can distinguish between the encryption of a message m using public key pk and encryption randomness $\mathsf{n_e}$, and the encryption of length of m zeros. While this axiom is not valid in general, we will show that it is valid (under some syntactic side-conditions on m) in any computational model where the function symbol $\{\_\}_\_$ is interpreted as an encryption satisfying some cryptographic properties (here IND-CCA$_1$).

Because of the syntactic side-condition on m, this axiom is actually an axiom schema, i.e. a recursive infinite set of axioms. We design such axiom schema for four usual cryptographic assumptions:

  - Indistinguishability against Chosen-Ciphertexts Attacks (IND-CCA$_1$).
  - Collision-Resistance under Hidden-Key attacks (CR-HK).
  - Unforgeability against Chosen-Message Attacks (EUF-CMA).
  - Pseudo Random Functions (PRF).

Outline We present some preliminary definitions in Section 2.1. In Section 2.2, we give the syntax of the logic. In Section 2.3, we present the first-order logic and computational semantics. In Section 2.4, we define protocols as labelled transition systems, give their semantics and prove the soundness theorem relating protocol executions and the logic. In Section 2.5, we present the structural axioms we designed, show their soundness, and give axiomatizations of several standard protocol function symbols (encryptions, pairs, xor and boolean tests). Finally, in Section 2.6 we translate several cryptographic assumptions into axiom schemata, and show their soundness.

2.1 Preliminaries

We write vectors using an arrow, as in $\vec{w}$. Given a vector $\vec{w}$, we let $|\vec{w}|$ be its length, and for every $1 \leq i \leq |\vec{w}|$, $\vec{w}_i$ is the i-th element of $\vec{w}$. Given a function $f : \mathcal{A} \mapsto \mathcal{B}$, we let $\mathsf{dom}(f)$ be its *domain* $\mathcal{A}$, and $\mathsf{codom}(f)$ be its *co-domain* $\mathcal{B}$. For any *random variable* f from a probability space Ω to a measurable space $\mathcal{A}$, we let $[w \in \Omega : f(w)]$ denote the *distribution* over $\mathcal{A}$ induced by f.

Words and Languages Given a finite or infinite set of symbols Σ, called an *alphabet*, we let Σ^* be the set of finite words over Σ, and Σ^ω the set of infinite words. The concatenation of two words w_1 and w_2 is denoted by $w_1 \cdot w_2$, and we let ϵ be the empty word. Given a symbol $a \in \Sigma$, we let a^ω be the only infinite word such that $a \cdot a^\omega = a^\omega$. Finally, for every word w, we let $|w|$ stands for the length of w.

Probability Measure on Infinite Tapes We consider Turing machines over the alphabet $\Sigma = \{0, 1\}$. In our model, we use *infinite* random tapes. We give here the definition of a standard probability measure μ_ω on infinite tapes Σ^ω. Basically, we see infinite tapes as binary representations of real numbers in the interval $[0, 1[$, and we use the Lebesgue measure on $[0, 1[$. Formally, let $\leq_{\mathsf{lex}}$ be the lexicographic ordering on Σ^ω such that $0^\omega < 1 \cdot 0^\omega$. Let $w_0, w_1 \in \Sigma^*$ such that $w_0 \cdot 0^\omega \leq_{\mathsf{lex}} w_1 \cdot 0^\omega$. The cylinder $\mathcal{C}^{w_1}_{w_0}$ is the subset of Σ^ω defined by:

$$\{w \mid w_0 \cdot 0^\omega \leq_{\mathsf{lex}} w \leq_{\mathsf{lex}} w_1 \cdot 0^\omega\}$$

One can easily check that finite disjoint unions of cylinders are a ring of subsets of Σ^ω (i.e. non-empty, closed under finite union and closed under relative complement).

We define the function μ_0 from finite disjoint unions of cylinders to $\mathbb{R}$ as follows:

$$\forall \text{ cylinder } \mathcal{C}^{w_1}_{w_0}.\, \mu_0\left(\mathcal{C}^{w_1}_{w_0}\right) = 0.w_1 - 0.w_0 \qquad \forall (\mathcal{C}_i)_{i\in I}.\, \mu_0\left(\bigcup_{i\in I}\mathcal{C}_i\right) = \sum_{i\in I}\mu_0\left(\mathcal{C}_i\right)$$

μ_0 is a pre-measure (i.e. μ_0 is σ-additive and $\mu_0(\emptyset) = 0$). Moreover, $\mu_0(\Sigma^\omega) = 1$. Therefore, using Carathéodory's extension theorem we can extend μ_0 into a measure μ_ω on the σ-algebra generated by the sets of finite disjoint unions of cylinders. Moreover, because $\mu_0(\Sigma^\omega) = 1$, this measure is unique and is a probability measure.

2.2 Syntax

2.2.1 Syntax of the Logic

Sorts and Types The Bana-Comon indistinguishability logic is a sorted logic, with only two sorts term and bool. For every $n \in \mathbb{N}$, we let Types_n be the set of types of n-ary functions:

$$\mathsf{Types}_n = \{(\mathsf{d}_1 \times \cdots \times \mathsf{d}_n \to \mathsf{d}_{n+1}) \mid \forall i, \mathsf{d}_i \in \{\mathsf{term}, \mathsf{bool}\}\}$$

Terms Let $\mathcal{F}$ be a set of function symbols, and $\mathsf{arity} : \mathcal{F} \mapsto \mathbb{N}$ their arity. We let $f_{/\mathsf{a}} \in \mathcal{F}$ denote the fact that $f \in \mathcal{F}$ and $\mathsf{arity}(f) = \mathsf{a}$. Let $\mathcal{N}$ be a countable set of names (representing random samplings) and $\mathcal{X}$ a countable set of variables. All names in $\mathcal{N}$ have sort term, and every variable in $\mathcal{X}$ comes with a sort. We assume that there is infinitely many variables of each sort. Finally, every function symbol $f_{/a} \in \mathcal{F}$ has at least one type, and we let $\mathsf{types}(f)$ be the types of f. We require that $\mathsf{types}(f) \subseteq \mathsf{Types}_n$, and that for every $(\mathsf{d}_1, \ldots, \mathsf{d}_n) \in \{\mathsf{term}, \mathsf{bool}\}^n$, there exists at most one $\mathsf{d} \in \{\mathsf{term}, \mathsf{bool}\}$ such that $(\mathsf{d}_1 \times \cdots \times \mathsf{d}_n \to \mathsf{d}) \in \mathsf{types}(f)$.

The set $\mathcal{F}$ of function symbols comprises a countable set of adversarial function symbols $\mathcal{G}$ (representing the adversary computations), and a set of protocol function symbols $\mathcal{F}_\mathsf{p}$. We require that $\mathcal{G}$ contains an infinite number of function symbols of arity n for every $n \in \mathbb{N}$. We also ask that $\mathcal{F}_\mathsf{p}$ contains at least the function symbols $\mathbf{0}_{/0}$, $\mathsf{true}_{/0}$, $\mathsf{false}_{/0}$, $\mathsf{len}_{/1}$, $\mathsf{eq}_{/2}$ and $\mathsf{if\_then\_else\_}_{/3}$, with the following types:

$$\mathbf{0}_{/0} :\to \mathsf{term} \qquad \mathsf{true}_{/0}, \mathsf{false}_{/0} :\to \mathsf{bool} \qquad \mathsf{eq}_{/2} : \mathsf{term}^2 \to \mathsf{bool} \qquad \mathsf{len}_{/1} : \mathsf{term} \to \mathsf{term}$$

$$\mathsf{if\_then\_else\_}_{/3} : \begin{cases} \mathsf{bool} \times \mathsf{bool}^2 \to \mathsf{bool} \\ \mathsf{bool} \times \mathsf{term}^2 \to \mathsf{term} \end{cases}$$

We let $\mathcal{F}_{\setminus \mathsf{if}}$ be $\mathcal{F}$ without the $\mathsf{if\_then\_else\_}$ function symbol, and for any subset $\mathcal{S}$ of $\mathcal{F}$, $\mathcal{N}$ and $\mathcal{X}$, we let $\mathcal{T}(\mathcal{S})$ be the set of terms built upon $\mathcal{S}$ (we require that terms are well-typed). Given a term t, the type of t is its larger type, where bool is smaller than term.

Given a term t, we let $\mathsf{st}(t)$ be the set of subterms of t and $\mathsf{var}(t) = \mathsf{st}(t) \cap \mathcal{X}$ be its variables. Given a set of variables $\mathcal{X}$, a substitution θ over $\mathcal{X}$ is a function from the set of variables $\mathcal{X}$ to some set of terms $\mathcal{T}(\mathcal{S})$. We sometimes use a post-fix notation for substitutions: given a term t and a substitution θ, we let $t\theta$ denote the application of θ to t.

Example 2.1. We re-use the example from the introduction. The term:

$$\mathsf{if}\ \mathsf{eq}(g(\phi), \mathsf{secret})\ \mathsf{then}\ \textsc{id}_\mathsf{A}\ \mathsf{else}\ \mathsf{Error}$$

can be used to model the output of an agent that checks if its input $g(\phi)$ is equal to a secret value secret (which can be a constant function symbol, a name in $\mathcal{N}$, or a more complex term). If this is the case, the agent outputs its identity $\textsc{id}_\mathsf{A}$, and otherwise it sends an error message Error. □

Example 2.2. We give an example of modeling of a full protocol, the AKA$^-$ protocol described in Figure 1.1. First, the signature: we use a constant function symbol $\textsc{id}_\mathsf{A}$ for the user's identity, the public/private key functions $\mathsf{pk}(\_), \mathsf{sk}(\_)$, asymmetric encryption and decryption $\{\_\}_\_$, $\mathsf{dec}(\_, \_)$, symmetric encryption and decryption $\mathsf{senc}(\_, \_)$, $\mathsf{sdec}(\_, \_)$, the pair $\langle \_\ , \_ \rangle$, projections π_1, π_2, the greater-than

test $\mathsf{geq}(\_,\_)$, the successor $\_+1$ and the error messages $\mathsf{UnknownId}, \mathsf{error}$. We give their types:

$$\mathsf{UnknownId}, \mathsf{error}, \mathrm{ID}_\mathsf{A} \ : \to \mathsf{term} \qquad \mathsf{geq}(\_,\_) \ : \ \mathsf{term}^2 \to \mathsf{bool} \qquad \{\_\}^{\_}_{\_} \ : \ \mathsf{term}^3 \to \mathsf{term}$$

$$\mathsf{pk}(\_),\ \mathsf{sk}(\_),\ \pi_1(\_),\ \pi_2(\_),\ (\_+1) \ : \ \mathsf{term} \to \mathsf{term}$$

$$\mathsf{dec}(\_,\_),\ \mathsf{senc}(\_,\_),\ \mathsf{sdec}(\_,\_),\ \langle \_\,,\,\_\rangle \ : \ \mathsf{term}^2 \to \mathsf{term}$$

The public/private key pair take as argument the seed used in the key generation. Corresponding public/private keys are keys with the same random, e.g. the public key $\mathsf{pk}(\mathsf{n})$ corresponds to the private key $\mathsf{sk}(\mathsf{n})$. The asymmetric encryption takes the encryption randomness as an extra parameter: $\{\mathrm{ID}_\mathsf{A}\}^{\mathsf{n_e}}_{\mathsf{pk}_\mathrm{N}}$ is the encryption of ID_A using public key pk_N and randomness $\mathsf{n_e}$.

We let $\mathsf{pk}_\mathrm{N} \equiv \mathsf{pk}(\mathsf{n}_\mathrm{N})$ and $\mathsf{sk}_\mathrm{N} \equiv \mathsf{sk}(\mathsf{n}_\mathrm{N})$, where $\mathsf{n}_\mathrm{N} \in \mathcal{N}$, be the public/private key of the service provider. The shared long-term symmetric key k is a name in $\mathcal{N}$. Then, the initial message from the user to the service provider is simply the term $\{\mathrm{ID}_\mathsf{A}\}^{\mathsf{n_e}}_{\mathsf{pk}_\mathrm{N}}$, where $\mathsf{n_e} \in \mathcal{N}$.

When receiving an input x, the service provider retrieves the encrypted identity and checks whether it is equal to the stored identity using the test $\mathsf{eq}(\mathsf{dec}(\mathsf{x}, \mathsf{sk}_\mathrm{N}), \mathrm{ID}_\mathsf{A})$. If the test succeed, the network sends the encryption of a random nonce $\mathsf{n} \in \mathcal{N}$ and of the current value of the sequence number SQN_S (represented by the term $\sigma^{\mathsf{in}}(\mathrm{SQN}_\mathsf{S})$). If the test fails, it sends the error message $\mathsf{UnknownId}$. This yields the message:

$$t_\mathsf{S}[\mathsf{x}] \equiv \mathsf{if}\ \mathsf{eq}(\mathsf{dec}(\mathsf{x}, \mathsf{sk}_\mathrm{N}), \mathrm{ID}_\mathsf{A})\ \mathsf{then}\ \mathsf{senc}(\langle \mathsf{n}\,,\, \sigma^{\mathsf{in}}(\mathrm{SQN}_\mathsf{S})\rangle, \mathsf{k}) \\ \mathsf{else}\ \mathsf{UnknownId}$$

It also updates the sequence number if the test is successful. The updated sequence number is represented by the term $\sigma(\mathrm{SQN}_\mathsf{S})$ given below:

$$\sigma(\mathrm{SQN}_\mathsf{S}) \equiv \mathsf{if}\ \mathsf{eq}(\mathsf{dec}(\mathsf{x}, \mathsf{sk}_\mathrm{N}), \mathrm{ID}_\mathsf{A})\ \mathsf{then}\ \sigma^{\mathsf{in}}(\mathrm{SQN}_\mathsf{S}) + 1 \\ \mathsf{else}\ \sigma^{\mathsf{in}}(\mathrm{SQN}_\mathsf{S})$$

When getting an answer y, the user decrypts the message using the key k, which yield $t_{\mathsf{dec}} \equiv \mathsf{sdec}(\mathsf{y}, \mathsf{k})$. It then extracts the service provider sequence number from t_{dec} using $\pi_2(t_{\mathsf{dec}})$, and checks that it is larger than its sequence number $\sigma^{\mathsf{in}}(\mathrm{SQN}_\mathsf{A})$ using the term:

$$\mathsf{accept} \equiv \mathsf{geq}(\pi_2(t_{\mathsf{dec}}), \sigma^{\mathsf{in}}(\mathrm{SQN}_\mathsf{A}) + 1)$$

Finally, the user's answer and the updated sequence number are represented by the terms:

$$\begin{aligned} t_\mathsf{A} \ &\equiv\ \mathsf{if}\ \mathsf{accept}\ \mathsf{then}\ \mathsf{senc}(\pi_1(t_{\mathsf{dec}}), \mathsf{k})\ \mathsf{else}\ \mathsf{error} \\ \sigma(\mathrm{SQN}_\mathsf{A}) \ &\equiv\ \mathsf{if}\ \mathsf{accept}\ \mathsf{then}\ \pi_2(t_{\mathsf{dec}})\ \mathsf{else}\ \sigma^{\mathsf{in}}(\mathrm{SQN}_\mathsf{A}) \end{aligned}$$ □

Formulas For every integer $n \in \mathbb{N}$, we have one predicate symbol $\sim_n$ of arity $2n$, which represents the equivalence between two vectors of terms of length n. We require that pairs of terms at matching positions (i.e. at position i and $n+i$) have the same sort. Formally, $\sim_n$ has the following types:

$$\sim_n\colon \{\mathcal{L} \times \mathcal{L} \mid \mathcal{L} \in \{\mathsf{bool}, \mathsf{term}\}^n\}$$

For every $\mathcal{L} \in \{\mathsf{bool}, \mathsf{term}\}^n$ and terms $u_1, \ldots, u_n, v_1, \ldots, v_n$ of sort $\mathcal{L}^2$, $\sim_n (u_1, \ldots, u_n, v_1, \ldots, v_n)$ is an atomic formula. From now on, we will use an infix notation for $\sim_n$, writing $u_1, \ldots, u_n \sim_n v_1, \ldots, v_n$ instead of $\sim_n (u_1, \ldots, u_n, v_1, \ldots, v_n)$. Moreover, we omit the index n when it is not necessary.

Formulas are obtained using atomic formulas, $\top$, $\bot$, the Boolean connectives $\wedge, \vee, \neg, \to$ and the first-order quantifiers $\forall, \exists$.

Example 2.3. For example, the formula:

$$\mathsf{if}\ g()\ \mathsf{then}\ \mathsf{n}_0\ \mathsf{else}\ \mathsf{n}_1 \sim \mathsf{n}$$

states that sampling from n_0 or n_1, depending on the branch chosen by the adversarial function $g()$, is equivalent to sampling from a single name n.

As a second example, we can express the fact that the first message of the AKA$^-$ protocol in Figure 1.1 guarantees the user anonymity: the formula $\{\mathrm{ID}_\mathsf{A}\}^{\mathsf{n_e}}_{\mathsf{pk}_\mathrm{N}} \sim \{\mathrm{ID}_\mathsf{B}\}^{\mathsf{n_e}}_{\mathsf{pk}_\mathrm{N}}$ states that users A and B first messages are indistinguishable. □

2.2.2 Positions and Contexts

We formally define the standard notions of positions of a term and of contexts. We also use the notion of if-context, which is a context that uses only the if_then_else_ function symbol.

Definition 2.1. A *position* is a word in $\mathbb{N}^*$. The value of a term t at a position p, denoted by $(t)_{|p}$, is the partial function defined inductively as follows:

$$\begin{array}{lcl} (t)_{|\epsilon} & = & t \\ (f(u_0,\ldots,u_{n-1}))_{|i.p} & = & \begin{cases} (u_i)_{|p} & \text{if } i < n \\ \text{undefined} & \text{otherwise} \end{cases} \end{array}$$

We say that a position is *valid* in t if $(t)_{|p}$ is defined. The set of positions of a term is the set of positions which are valid in t, and is denoted $\mathsf{pos}(t)$.

Definition 2.2. A context $D[]_{\vec{x}}$ (written D when there is no confusion) on a signature $\mathcal{S}$ is a term in:

$$\mathcal{T}(\mathcal{S}, \{[]_y \mid y \in \vec{x}\})$$

where $\vec{x}$ are distinct special variables called holes. A one-holed context is a context with one hole (in which case we write $D[]$ where [] is the only variable).

For all contexts $D[]_{\vec{x}}, C_0, \ldots, C_{n-1}$ with $|\vec{x}| = n$, we let $D[(C_i)_{i<n}]$ be the context $D[]_{\vec{x}}$ in which we substitute, for every $0 \leq i < n$, all occurrences of the hole $[]_{x_i}$ by C_i.

If-Contexts Often, we want to distinguish between holes that contain "internal" conditionals, and holes that contain terms appearing at the leaves. To do this we introduce the notion of if-context. An if-context $D[]_{\vec{x} \diamond \vec{y}}$ is a context using only the if_then_else_ function symbol and two sets of holes variables: $\vec{x}$ is for conditionals and $\vec{y}$ is for leaves.

Definition 2.3. For all distinct variables $\vec{x}, \vec{y}$, an if-context $D[]_{\vec{x} \diamond \vec{y}}$ is a context in:

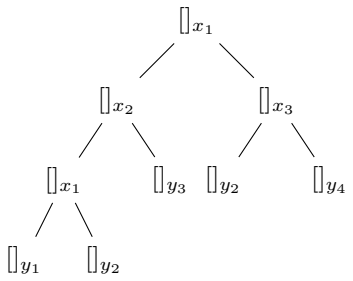
$$\mathcal{T}(\mathsf{if\_then\_else\_}, \{[]_z \mid z \in \vec{x} \cup \vec{y}\})$$

such that for all position p, $D_{|p} \equiv \mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v$ implies:

- $b \in \{[]_z \mid z \in \vec{x}\}$
- $u, v \notin \{[]_z \mid z \in \vec{x}\}$

Example 2.4. Let $\vec{x} = x_1, x_2, x_3$ and $\vec{y} = y_1, y_2, y_3, y_4$, we give below two representations of the same if-context $D[]_{\vec{x} \diamond \vec{y}}$ (the term on the left, and the labelled tree on the right):

$$\mathsf{if}\ []_{x_1}\ \mathsf{then}\ \begin{pmatrix} \mathsf{if}\ []_{x_2}\ \mathsf{then}\ \mathsf{if}\ []_{x_1}\ \mathsf{then}\ []_{y_1}\ \mathsf{else}\ []_{y_2} \\ \qquad\qquad \mathsf{else}\ []_{y_3} \end{pmatrix} \\ \mathsf{else}\ (\mathsf{if}\ []_{x_3}\ \mathsf{then}\ []_{y_2}\ \mathsf{else}\ []_{y_4})$$



□

2.3 Semantics

2.3.1 Sorted First-order Semantics

We use the classical semantics for sorted first-order logic: every sort is interpreted by some domain and function symbols and predicates are interpreted as, respectively, functions of the appropriate domains and relations on these domains.

Model Formally, an model $\mathcal{M}$ is a tuple $(\mathcal{D}_{\mathsf{term}}, \mathcal{D}_{\mathsf{bool}}, [\![\ ]\!]_{\mathsf{n}}, [\![\ ]\!]_{\mathsf{f}}, [\![\ ]\!]_{\mathsf{p}})$, where the domains $\mathcal{D}_{\mathsf{term}}$ and $\mathcal{D}_{\mathsf{bool}}$ are for terms of sort, respectively, term and bool. We require that:

- The name interpretation $[\![\ ]\!]_{\mathsf{n}}$ associates to every name $n \in \mathcal{N}$ a member of $\mathcal{D}_{\mathsf{term}}$.
- The function symbol interpretation $[\![\ ]\!]_{\mathsf{f}}$ associates to every $f_{/n} \in \mathcal{F}$ and type $\mathsf{s} = (\mathsf{d}_1, \ldots, \mathsf{d}_n \to \mathsf{d}_{n+1}) \in \mathsf{types}(f)$ a function:

$$[\![f]\!]^{\mathsf{s}}_{\mathsf{f}} : \mathcal{D}_1 \times \cdots \times \mathcal{D}_n \to \mathcal{D}_{n+1} \qquad \text{where, for every } i,\ \mathcal{D}_i = \begin{cases} \mathcal{D}_{\mathsf{term}} & \text{if } \mathsf{d}_i = \mathsf{term} \\ \mathcal{D}_{\mathsf{bool}} & \text{if } \mathsf{d}_i = \mathsf{bool} \end{cases}$$

- The predicate interpretation $[\![\ ]\!]^{\mathcal{L}}_{\mathsf{p}}$ associates to every predicate $\sim_n$ and $\mathcal{L} \in \{\mathsf{bool}, \mathsf{term}\}^n$ a subset of:

$$(\mathcal{D}_1 \times \cdots \times \mathcal{D}_n)^2$$

Term Interpretation We then define inductively the interpretation of a term in $\mathcal{M}$. Let σ a valuation from $\mathcal{X}$ to the appropriate domains. We define $[\![\_]\!]^{\sigma}_{\mathcal{M}}$ as follows:

- For any $x \in \mathcal{X}$, $[\![x]\!]^{\sigma}_{\mathcal{M}} = \sigma(x)$.
- For every $\mathsf{n} \in \mathcal{N}$, $[\![\mathsf{n}]\!]^{\sigma}_{\mathcal{M}} = [\![\mathsf{n}]\!]_{\mathsf{n}}$.
- Let $f \in \mathcal{F}$ and $u_1, \ldots, u_n$ be terms with types $\mathsf{d}_1, \ldots, \mathsf{d}_n$ such that $\mathsf{s} = (\mathsf{d}_1 \times \cdots \times \mathsf{d}_n \to \mathsf{d}_{n+1}) \in \mathsf{types}(f)$. Then:
$$[\![f(u_1, \ldots, u_n)]\!]^{\sigma}_{\mathcal{M}} = [\![f]\!]^{\mathsf{s}}_{\mathsf{f}}\,([\![u_1]\!]^{\sigma}_{\mathcal{M}}, \ldots, [\![u_n]\!]^{\sigma}_{\mathcal{M}})$$
If the function symbol f has a unique type s, we omit it and write $[\![f]\!]_{\mathsf{f}}$ instead of $[\![f]\!]^{\mathsf{s}}_{\mathsf{f}}$.

Because we have a sorted logic, the fact that $[\![\_]\!]^{\sigma}_{\mathcal{M}}$ is well-defined on term is not immediate from the definition, and must be shown.

Proposition 2.1. *For every model $\mathcal{M}$, $[\![\_]\!]^{\sigma}_{\mathcal{M}}$ is a total function from $\mathcal{T}(\mathcal{F}, \mathcal{N}, \mathcal{X})$ to $\mathcal{D}_{\mathsf{term}} \cup \mathcal{D}_{\mathsf{bool}}$. Moreover, for every term t, if t is of sort term (resp. bool), then $[\![t]\!]^{\sigma}_{\mathcal{M}}$ is a member of $\mathcal{D}_{\mathsf{term}}$ (resp. $\mathcal{D}_{\mathsf{bool}}$).*

Proof. We show this by structural induction on the term. For the function symbol case, we rely on the fact that for every $f \in \mathcal{F}$ and $(\mathsf{d}_1, \ldots, \mathsf{d}_n) \in \{\mathsf{term}, \mathsf{bool}\}^n$, there exists at most on $\mathsf{d} \in \{\mathsf{term}, \mathsf{bool}\}$ such that $(\mathsf{d}_1 \times \cdots \times \mathsf{d}_n \to \mathsf{d}_{n+1}) \in \mathsf{types}(f)$. ■

If t is a ground term (i.e. $\mathsf{var}(t) = \emptyset$) then its interpretation is independent from the valuation σ. In that case, we omit σ and write $[\![t]\!]_{\mathcal{M}}$.

Formula Interpretation We extend $[\![\_]\!]^{\sigma}_{\mathcal{M}}$ to interpret formulas in $\{\mathsf{True}, \mathsf{False}\}$ as follows:

- $[\![\bot]\!]^{\sigma}_{\mathcal{M}} = \mathsf{False}$ and $[\![\top]\!]^{\sigma}_{\mathcal{M}} = \mathsf{True}$.
- $[\![\neg\phi]\!]^{\sigma}_{\mathcal{M}} = \mathsf{True}$ iff $[\![\phi]\!]^{\sigma}_{\mathcal{M}} = \mathsf{False}$.
- $[\![\phi \wedge \psi]\!]^{\sigma}_{\mathcal{M}} = \mathsf{True}$ iff $[\![\phi]\!]^{\sigma}_{\mathcal{M}} = [\![\psi]\!]^{\sigma}_{\mathcal{M}} = \mathsf{True}$.
- $[\![\phi \vee \psi]\!]^{\sigma}_{\mathcal{M}} = [\![\neg(\neg\phi \wedge \neg\psi)]\!]^{\sigma}_{\mathcal{M}}$.
- $[\![\phi \to \psi]\!]^{\sigma}_{\mathcal{M}} = [\![\neg\phi \vee \psi)]\!]^{\sigma}_{\mathcal{M}}$.
- Let x be a variable of sort dom. Then $[\![\exists x.\phi]\!]^{\sigma}_{\mathcal{M}} = \mathsf{True}$ iff there exists $v \in \mathcal{D}_{\mathsf{dom}}$ such that $[\![\phi]\!]^{\sigma'}_{\mathcal{M}} = \mathsf{True}$ where:
$$\forall y \in \mathcal{X}, \sigma'(y) = \begin{cases} v & \text{if } y = x \\ \sigma(y) & \text{otherwise} \end{cases}$$
- $[\![\forall x.\phi]\!]^{\sigma}_{\mathcal{M}} = [\![\neg\exists x.\neg\phi]\!]^{\sigma}_{\mathcal{M}}$.
- Let $n \in \mathbb{N}$ and $\mathcal{L} \in \{\mathsf{bool}, \mathsf{term}\}^n$. For every terms $u_1, \ldots, u_n, v_1, \ldots, v_n$ of sort $\mathcal{L}^2$:
$$[\![u_1, \ldots, u_n \sim_n v_1, \ldots, v_n]\!]^{\sigma}_{\mathcal{M}} \text{ iff } ([\![u_1]\!]^{\sigma}_{\mathcal{M}}, \ldots, [\![u_n]\!]^{\sigma}_{\mathcal{M}}, [\![v_1]\!]^{\sigma}_{\mathcal{M}}, \ldots, [\![v_n]\!]^{\sigma}_{\mathcal{M}}) \in [\![\sim_n]\!]^{\mathcal{L}}_{\mathsf{p}}$$

A formula ϕ is valid in $\mathcal{M}$, denoted by $\models_{\mathcal{M}} \phi$, if and only if for every valuation σ, $[\![\phi]\!]^{\sigma}_{\mathcal{M}} = \mathsf{True}$. Finally we say that a formula is valid, denoted by $\models \phi$, if and only if ϕ is valid in all models, i.e. for all $\mathcal{M}$ we have $\models_{\mathcal{M}} \phi$.

2.3.2 Computational Models

We focus on a particular class of such models, the *computational models* (introduced in [BCL14]). Morally, a computational model corresponds to the interaction of a protocol implementation, which associates to every protocol function symbol in $\mathcal{F}_{\mathsf{p}}$ its implementation, with an polynomial-time adversary, which we let interpret the adversarial function symbols in $\mathcal{G}$. A computational model is a model where we interpret terms as *probabilistic polynomial-time Turing Machines* (PPTMs) and the predicates $\sim$ as computational indistinguishability. Formally, a model $\mathcal{M}_{\mathsf{c}} = (\mathcal{D}_{\mathsf{term}}, \mathcal{D}_{\mathsf{bool}}, [\![\ ]\!]_{\mathsf{n}}, [\![\ ]\!]_{\mathsf{f}}, [\![\ ]\!]_{\mathsf{p}})$ is a computational model if:

- $\mathcal{D}_{\mathsf{term}}$ is the set of *deterministic* polynomial-time Turing machines equipped with an input tape, which is also the working tape, and two additional infinite read-only tapes ρ_1, ρ_2, the *random tapes*, used for random samplings.[1] We need two random tapes to prevent the adversary from seeing the protocol random samplings (such as a secret key's random seed). More specifically, the tape ρ_1 is for the protocol random samplings, while ρ_2 is for the adversary random samplings. This will appear in the restrictions on the function symbols interpretations below. The machines in $\mathcal{D}_{\mathsf{term}}$ must run in polynomial-time with respect to the length of the *input tape* only.
- bool is the restriction of term to machines that return only the bit-strings 0 or 1.
- A name $\mathsf{n} \in \mathcal{N}$ is interpreted as a machine that, on input (w, ρ_1, ρ_2) where w is of length η, extracts a word of length η from the first random tape ρ_1. Furthermore we require that different names extract disjoint parts of ρ_1.
- The interpretation of the function symbols $\mathsf{true}_{/0}$, $\mathsf{false}_{/0}$, $\mathsf{len}_{/1}$, $\mathbf{0}_{/1}$, $\mathsf{eq}_{/2}$ and $\mathsf{if\_then\_else\_}_{/3}$ by $[\![\_]\!]^{\mathsf{s}}_{\mathsf{f}}$ is fixed, and the expected one:
  - $[\![\mathsf{true}]\!]^{\mathsf{s}}_{\mathsf{f}}$ (resp. $[\![\mathsf{false}]\!]^{\mathsf{s}}_{\mathsf{f}}$) is the machine that, on input (m, ρ_1, ρ_2), returns the bit-string 1 (resp. 0).
  - $[\![\mathsf{len}]\!]_{\mathsf{f}}$ (resp. $[\![\mathbf{0}]\!]^{\mathsf{s}}_{\mathsf{f}}$) is the function that, given a machine m, returns the machine M such that, on any input (w, ρ_1, ρ_2), $\mathsf{M}(w, \rho_1, \rho_2)$ returns the lengths of $m(w, \rho_1, \rho_2)$ in binary (resp. the bit-string $0^{|m|}$).
  - The interpretation of eq is the same for its two types. $[\![\mathsf{eq}]\!]^{\mathsf{s}}_{\mathsf{f}}$ is a function that takes as input two machines (m_1, m_2) and returns a polynomial-time machine M such that:

$$\mathsf{M}(w, \rho_1, \rho_2) = \begin{cases} 1 & \text{if } m_1(w, \rho_1, \rho_2) = m_2(w, \rho_1, \rho_2) \\ 0 & \text{otherwise} \end{cases}$$

  - The interpretation of $\mathsf{if\_then\_else\_}$ does not depend on the types of its arguments. The function $[\![\mathsf{if\_then\_else\_}]\!]_{\mathsf{f}}$ takes as input three machines (m_1, m_2, m_3) and returns a polynomial-time machine M such that:

$$\mathsf{M}(w, \rho_1, \rho_2) = \begin{cases} m_2(w, \rho_1, \rho_2) & \text{if } m_1(w, \rho_1, \rho_2) = 1 \\ m_3(w, \rho_1, \rho_2) & \text{otherwise} \end{cases}$$

- Let $f \in \mathcal{F}_{\mathsf{p}} \backslash \{\mathsf{if\_then\_else\_}, \mathsf{true}, \mathsf{false}, \mathsf{eq}, \mathsf{len}, \mathbf{0}\}$ be a protocol function symbol, and $\mathsf{s} = \mathsf{d}_1 \times \cdots \times \mathsf{d}_n \to \mathsf{d}_{n+1} \in \mathsf{types}(f)$. Then $[\![f]\!]^{\mathsf{s}}_{\mathsf{f}}$ is defined by a *deterministic* polynomial-time Turing machine $\mathcal{M}^f_c$ with n input tapes as follows: $[\![f]\!]^{\mathsf{s}}_{\mathsf{f}}$ is the function that, on input $(m_1, \ldots, m_n) \in \mathcal{D}_{\mathsf{d}_1} \times \cdots \times \mathcal{D}_{\mathsf{d}_n}$, returns a machine $[\![f]\!]^{\mathsf{s}}_{\mathsf{f}}(m_1, \ldots, m_n)$ in $\mathcal{D}_{\mathsf{d}_{n+1}}$ such that for every (w, ρ_1, ρ_2):

$$[\![f]\!]^{\mathsf{s}}_{\mathsf{f}}(m_1, \ldots, m_n)(w, \rho_1, \rho_2) = \mathcal{M}^f_c\left(m_1(w, \rho_1, \rho_2), \ldots, m_n(w, \rho_1, \rho_2)\right)$$

  This is just the composition of $\mathcal{M}^f_c$ with $(m_1, \ldots, m_n)$. Observe that $\mathcal{M}^f_c$ is deterministic and has no direct access to the random tapes ρ_1, ρ_2. Nonetheless, it can access to the random tapes through its argument $(m_1, \ldots, m_n)$. This ensures that all random samplings must appear explicitly in the terms.
- Let $g \in \mathcal{G}$ be an adversarial function symbol, and $\mathsf{s} = \mathsf{d}_1 \times \cdots \times \mathsf{d}_n \to \mathsf{d}_{n+1} \in \mathsf{types}(g)$. Adversarial function symbols are interpreted as protocol function symbols, except that they have an additional input tape for ρ_2. More precisely, $[\![g]\!]^{\mathsf{s}}_{\mathsf{f}}$ is characterized by a *deterministic* polynomial-time Turing

[1] We represent probabilistic Turing machine using deterministic Turing machine with explicit (infinite) random tapes.

machine $\mathcal{M}_c^g$ with $n+1$ input tapes as follows: $[\![g]\!]_{\mathsf{f}}^{\mathsf{s}}$ is the function that, on input $(m_1,\dots,m_n) \in \mathcal{D}_{\mathsf{d}_1} \times \cdots \times \mathcal{D}_{\mathsf{d}_n}$, returns a machine $[\![g]\!]_{\mathsf{f}}^{\mathsf{s}}(m_1,\dots,m_n)$ in $\mathcal{D}_{\mathsf{d}_{n+1}}$ such that for every (w,ρ_1,ρ_2):

$$[\![g]\!]_{\mathsf{f}}^{\mathsf{s}}(m_1,\dots,m_n)(w,\rho_1,\rho_2) = \mathcal{M}_c^g\left(m_1(w,\rho_1,\rho_2),\dots,m_2(w,\rho_1,\rho_2),\rho_2\right)$$

$\mathcal{M}_c^g$ has access to the random tape ρ_2, which allows the adversary to perform random samplings. It has no direct access to the protocol random tape ρ_1.

- Let $n \in \mathbb{N}$. The predicate $\sim_n$ is interpreted as *computational indistinguishability* $\approx_n$.
  Let $(\mathsf{d}_1,\dots,\mathsf{d}_n) \in \{\mathsf{bool},\mathsf{term}\}^n$, for every machines $m_1,\dots,m_n,m'_1,\dots,m'_n$ in $(\mathcal{D}_{\mathsf{d}_1},\dots,\mathcal{D}_{\mathsf{d}_n})^2$, we have $m_1,\dots,m_n \approx_n m'_1,\dots,m'_n$ iff for every PPTM $\mathcal{A}$, the following quantity is negligible in η:

$$\left|\begin{array}{ll} & \mathbf{Pr}\left(\rho_1,\rho_2 : \mathcal{A}(1^\eta,(m_i(1^\eta,\rho_1,\rho_2))_{1\le i\le n},\rho_2)=1\right) \\ - & \mathbf{Pr}\left(\rho_1,\rho_2 : \mathcal{A}(1^\eta,(m'_i(1^\eta,\rho_1,\rho_2))_{1\le i\le n},\rho_2)=1\right)\end{array}\right|$$

  where ρ_1 and ρ_2 are drawn using μ_ω among the set of infinite random tapes. Again, observe that $\mathcal{A}$ has direct access to ρ_2 but not ρ_1.

Syntactic Sugar We introduce a shorter notation for $[\![\_]\!]$, by writing:

$$[\![u]\!]_{\mathcal{M}_{\mathsf{c}}}^{\eta,\rho_1,\rho_2} \qquad \text{instead of} \qquad [\![u]\!]_{\mathcal{M}_{\mathsf{c}}}(1^\eta,\rho_1,\rho_2)$$

We also lift the definition of $[\![\_]\!]_{\mathcal{M}_{\mathsf{c}}}$ to tuples of terms. Let $\vec{u} = u_1,\dots,u_n$ be a vector of ground terms, then $[\![\vec{u}]\!]_{\mathcal{M}_{\mathsf{c}}}$ is such that for every (w,ρ_1,ρ_2):

$$[\![\vec{u}]\!]_{\mathcal{M}_{\mathsf{c}}}^{w,\rho_1,\rho_2} = \left([\![u_1]\!]_{\mathcal{M}_{\mathsf{c}}}^{w,\rho_1,\rho_2},\dots,[\![u_n]\!]_{\mathcal{M}_{\mathsf{c}}}^{w,\rho_1,\rho_2}\right)$$

The important point is that all $[\![u_i]\!]_{\mathcal{M}_{\mathsf{c}}}$ are evaluated using the same random tapes ρ_1,ρ_2.

Remark 2.1 (Bit-String Distributions)**.** Alternatively, we can see $[\![\_]\!]_{\mathcal{M}_{\mathsf{c}}}$ as interpreting vectors of terms as family of distributions of bit-strings vectors, indexed by the security parameter η. More precisely, let $\mathcal{M}_{\mathsf{c}}$ be a computational model and $\vec{u}$ be a vector of ground term $\vec{u}$. Then for all η, we let d_η be the distribution:

$$d_\eta = \left[\rho_1,\rho_2 : [\![\vec{u}]\!]_{\mathcal{M}_{\mathsf{c}}}^{\eta,\rho_1,\rho_2}\right]$$

Then, two families of distributions of bit-string vectors $(d_\eta)_\eta$ and $(d'_\eta)_\eta$, indexed by η, are indistinguishable if and only if for every PPTM $\mathcal{A}$, the following quantity is negligible in η:

$$\left|\mathbf{Pr}\left(\rho_1,\rho_2 :\ \mathcal{A}\left(1^\eta,d_\eta(\rho_1,\rho_2),\rho_2\right)=1\right) - \mathbf{Pr}\left(\rho_1,\rho_2 :\ \mathcal{A}\left(1^\eta,d'_\eta(\rho_1,\rho_2),\rho_2\right)=1\right)\right|$$

Observe that $\mathcal{A}$ and d_η (resp. d'_η) uses the same random tape, i.e. we correlate the distribution d_η (resp. d'_η) and the random samplings of the adversary $\mathcal{A}$. □

Example 2.5. Consider the first formula in Example 2.3. Let $\mathsf{n}_0,\mathsf{n}_1,\mathsf{n} \in \mathcal{N}$ and $g \in \mathcal{F}$ of arity 0. Then in every computational model $\mathcal{M}_{\mathsf{c}}$, we know that:

$$\mathcal{M}_{\mathsf{c}} \models \mathsf{if}\ g()\ \mathsf{then}\ \mathsf{n}_0\ \mathsf{else}\ \mathsf{n}_1 \sim \mathsf{n}$$

Indeed, the term on the left represents the message obtained by letting the adversary choose a branch, and then sampling from n_0 or n_1 accordingly, which is semantically equivalent to directly performing a random sampling, as done on the right.

We now focus on the second formula of Example 2.3, which we recall below:

$$\{\mathrm{ID}_{\mathsf{A}}\}_{\mathsf{pk}_{\mathrm{N}}}^{\mathsf{n}_{\mathsf{e}}} \sim \{\mathrm{ID}_{\mathsf{B}}\}_{\mathsf{pk}_{\mathrm{N}}}^{\mathsf{n}_{\mathsf{e}}} \tag{2.1}$$

This formula is not true in every computational model $\mathcal{M}_{\mathsf{c}}$. For example, if $\mathcal{M}_{\mathsf{c}}$ is such that:

- the encryption function symbol is interpreted as the function returning its first argument (hence for every u,v,w, $[\![\{u\}_v^w]\!]_{\mathcal{M}_{\mathsf{c}}} = [\![u]\!]_{\mathcal{M}_{\mathsf{c}}}$).
- ID_{A} and ID_{B} are interpreted as two distinct elements of $\mathcal{D}_{\mathsf{term}}$, e.g. 0 and 1.

Then the formula in (2.1) does not hold in $\mathcal{M}_{\mathsf{c}}$. Indeed, there is a simple distinguisher $\mathcal{A}$, returning true if and only if it receives 0 as input, which wins (2.1) with probability one. □

2.4 Protocol and Their Semantics

As in [BCL14], we model protocols as abstract labelled transition systems. There are two differences between our modeling and the one in [BCL14]. First, our LTS includes state updates, which allows us to model stateful protocols, such as LAK or AKA. Second, we consider infinite LTS instead of finite ones. This allows us to model protocols with an arbitrary number of agents and sessions. Note however that our soundness result will hold only for adversaries having a constant (w.r.t. the security parameter), though arbitrary large, number of interactions with the protocol.

2.4.1 Labelled Transition Systems

A protocol P is a tuple $(\mathcal{Q}, \mathcal{L}, \mathsf{Vars}_\sigma, q_\epsilon, \phi_\epsilon, \sigma_\epsilon, \delta)$ where:

- $\mathcal{Q}$, $\mathcal{L}$ and Vars_σ are the (possibly infinite) sets of, respectively, nodes, action labels and state variables. A node $q \in \mathcal{Q}$ records the static component of a protocol execution. Typically, this includes the number of agents and running sessions. Action labels are the actions available to the adversary (e.g. creating a new user, starting a new session, or sending a message). Finally, state variables are symbolic handles used to refer to the memory of the agents.
- $q_\epsilon \in \mathcal{Q}$ is the initial node.
- The initial frame ϕ_ϵ is a vector of terms in $\mathcal{T}(\mathcal{F}, \mathcal{N})$. It represents the initial adversarial knowledge.
- The initial state σ_ϵ is a *total* substitution from Vars_σ to $\mathcal{T}(\mathcal{F}, \mathcal{N})$.
- δ is the transition relation associating to every node q a finite (non-empty) set $\delta(q)$ of transitions of the form $(\alpha, t, \sigma^{\mathsf{up}}, q')$ where $\alpha \in \mathcal{L}$ is the transition label, t is a term of sort term representing the message output when executing the transition, the state update σ^{up} is a finite substitution from Vars_σ to terms and q' is a node. We require that:

$$\{t\} \cup \mathsf{codom}(\sigma^{\mathsf{up}}) \subseteq \mathcal{T}(\mathcal{F}, \mathcal{N}, \mathsf{Vars}_\sigma, \{\mathsf{x}_{\mathsf{in}}\})$$

  where $\mathsf{x}_{\mathsf{in}} \notin \mathsf{Vars}_\sigma$. That is, the outputted message t and the state update σ^{up} depend only on the current state (through Vars_σ) and the message that was inputted from the network (through x_{in}). For every $q \in \mathcal{Q}$ and $\alpha \in \mathcal{L}$, there must exists *at most one* transition in $\delta(q)$ labelled by α.
- There is a distinguished label $\mathsf{nop} \in \mathcal{L}$ such that for every $q \in \mathcal{Q}$, $(\mathsf{nop}, \mathbf{0}, \epsilon, q) \in \delta(q)$.

We say that a label α is enabled in a node q if there exists a transition in $\delta(q)$ labelled by α. We also let $\mathsf{enbl}_P(q)$ be the set of labels enabled in q. Observe that $\mathsf{enbl}_P(q)$ is never empty since it contains at least nop.

For all $q \in \mathcal{Q}$ and $\alpha \in \mathsf{enbl}_P(q)$, we let $\delta^{\mathsf{q}}(q, \alpha) = q'$ where q' is the unique member of $\mathcal{Q}$ such that $(\alpha, \_, \_, q') \in \delta$.

Example 2.6. As a first example, we present one session of the AKA$^-$ protocol given in Figure 1.1, which we started to model in Example 2.2. We recall that one session of the AKA$^-$ protocol between a user A and its service provider S comprises three messages, two from A and one from S. We use the nodes $\mathcal{Q}$ to record what messages have already been sent. Therefore, we let $\mathcal{Q} = \mathcal{Q}_\mathsf{A} \times \mathcal{Q}_\mathsf{S}$ where:

$$\mathcal{Q}_\mathsf{A} = \{\mathsf{started}, \mathsf{running}, \mathsf{done}\} \qquad \mathcal{Q}_\mathsf{S} = \{\mathsf{started}, \mathsf{done}\}$$

For example, we are in a node $(\mathsf{started}, \_)$ if A has not yet sent any message, and we are in a node $(\mathsf{running}, \_)$ if A already sent its first message, but not the second message.

At any point of the protocol execution, only two actions are available to the adversary: it can either interact with A or with S. Therefore, we let $\mathcal{L} = \{\mathsf{A}, \mathsf{S}\}$. The set of state variables is $\mathsf{Vars}_\sigma = \{\mathrm{SQN}_\mathsf{A}, \mathrm{SQN}_\mathsf{S}\}$, which are used to store the current value of the sequence numbers. Initially, the two agents have not sent any messages, the service provider public key is made available to the adversary, and we initialize the sequence numbers SQN_A and SQN_S to some constant values $\mathsf{sqn\text{-}init}_\mathsf{A}$ and $\mathsf{sqn\text{-}init}_\mathsf{S}$:

$$q_\epsilon = (\mathsf{started}, \mathsf{started}) \qquad \phi_\epsilon = \mathsf{pk}_{\mathrm{N}} \qquad \forall \mathsf{x} \in \mathsf{Vars}_\sigma,\ \sigma_\epsilon(\mathsf{x}) = \begin{cases} \mathsf{sqn\text{-}init}_\mathsf{A} & \text{if } \mathsf{x} = \mathrm{SQN}_\mathsf{A} \\ \mathsf{sqn\text{-}init}_\mathsf{S} & \text{if } \mathsf{x} = \mathrm{SQN}_\mathsf{S} \end{cases}$$

Finally, we describe the transition relations δ. For every $q \in \mathcal{Q}$, $(\alpha, t, \sigma^{\mathsf{up}}, q') \in \delta(q)$ when:

- if $q = (\mathsf{started}, q_{\mathsf{S}})$ and $\alpha = \mathsf{A}$ then we have the first user message:

$$q' = (\mathsf{running}, q_{\mathsf{S}}) \qquad t \equiv \{\mathrm{ID}_{\mathsf{A}}\}^{\mathsf{n_e}}_{\mathsf{pk}_{\mathrm{N}}} \qquad \sigma^{\mathsf{up}} = \epsilon$$

- if $q = (\mathsf{running}, q_{\mathsf{S}})$ and $\alpha = \mathsf{A}$, then we have the second (and last) user message:

$$q' = (\mathsf{done}, q_{\mathsf{S}}) \qquad t_{\mathsf{dec}} \equiv \mathsf{sdec}(\mathsf{x_{in}}, \mathsf{k}) \qquad \mathsf{accept} \equiv \mathsf{geq}(\pi_2(t_{\mathsf{dec}}), \mathrm{SQN}_{\mathsf{A}} + 1)$$

$$t \equiv \mathsf{if\ accept\ then\ senc}(\pi_1(t_{\mathsf{dec}}), \mathsf{k})\ \mathsf{else\ error} \qquad \sigma^{\mathsf{up}} = \mathrm{SQN}_{\mathsf{A}} \mapsto \mathsf{if\ accept\ then}\ \pi_2(t_{\mathsf{dec}})\ \mathsf{else}\ \mathrm{SQN}_{\mathsf{A}}$$

- if $q = (q_{\mathsf{A}}, \mathsf{started})$ and $\alpha = \mathsf{S}$, then we have the service provider message:

$$q' = (q_{\mathsf{A}}, \mathsf{done}) \qquad t \equiv \mathsf{if\ eq}(\mathsf{dec}(\mathsf{x_{in}}, \mathsf{sk}_{\mathrm{N}}), \mathrm{ID}_{\mathsf{A}})\ \mathsf{then\ senc}(\langle \mathsf{n}\,,\, \mathrm{SQN}_{\mathsf{S}} \rangle, \mathsf{k})\ \mathsf{else\ UnknownId}$$

$$\sigma^{\mathsf{up}} = \mathrm{SQN}_{\mathsf{S}} \mapsto \mathsf{if\ eq}(\mathsf{dec}(\mathsf{x_{in}}, \mathsf{sk}_{\mathrm{N}}), \mathrm{ID}_{\mathsf{A}})\ \mathsf{then}\ \mathrm{SQN}_{\mathsf{S}} + 1\ \mathsf{else}\ \mathrm{SQN}_{\mathsf{S}}$$

In this example, the LTS is finite, as we consider only one session of the protocol. □

Example 2.7. We now give a more complex example. We still model the AKA$^-$ protocol, but this time we consider a setting with an arbitrary number of users, each with a different identity. To reduce the complexity of the example, every user runs only one sessions. We let $\mathcal{S} = \{\mathrm{ID}_i \mid i \in \mathbb{N}\}$ be a set of identities, which are indexed by an integer i. The set of nodes $\mathcal{Q}$ is:

$$\mathcal{Q} = \left\{ \begin{pmatrix} (\mathrm{ID}_1 : b_1, \ldots, \mathrm{ID}_l : b_l), \\ (i_1 : a_1, \ldots, i_n : a_n) \end{pmatrix} \;\middle|\; \begin{array}{l} \forall 1 \leq i \leq l, \mathrm{ID}_i \in \mathcal{S} \ \wedge\ \forall 1 \leq j \leq n, i_j \in \{1, \ldots, l\} \\ \wedge\, b_1, \ldots, b_l, a_1, \ldots, a_n \in \{\mathsf{true}, \mathsf{false}\} \end{array} \right\}$$

Being in the state $(\mathrm{ID}_1 : b_1, \ldots, \mathrm{ID}_l : b_l), (i_1 : a_1, \ldots, i_n : a_n)$ should be interpreted as follows:

- There are l users, with identities $\mathrm{ID}_1, \ldots, \mathrm{ID}_l$. Moreover, for every i, b_i is true iff user ID_i has not sent its last message yet.
- There are n service provider sessions, where the j-th network session is communicating with user ID_{i_j} (hence i_j must be in $\{1, \ldots, l\}$). Moreover, for every j, a_j is true iff the service provider session i_j has not sent its last message yet.

The set of state variable Vars_σ contains, for every $\mathrm{ID} \in \mathcal{S}$, the variables $\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}$ and $\mathrm{SQN}^{\mathrm{ID}}_{\mathsf{S}}$ storing, respectively, the user and service provider version of the sequence number. Initially, there are no users and no service provider sessions. The initial frame contains only the service provider public key pk_{N}, and all sequence numbers are initialized using constants $\mathsf{sqn\text{-}init}^{\mathrm{ID}}_{\mathrm{U}}$ and $\mathsf{sqn\text{-}init}^{\mathrm{ID}}_{\mathsf{S}}$.

$$q_\epsilon = ((), ()) \qquad \phi^{\mathsf{in}}_\epsilon \equiv \mathsf{pk}_{\mathrm{N}} \qquad \forall \mathsf{x} \in \mathsf{Vars}_\sigma,\ \sigma_\epsilon(\mathsf{x}) = \begin{cases} \mathsf{sqn\text{-}init}^{\mathrm{ID}}_{\mathrm{U}} & \text{if } \mathsf{x} = \mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}} \\ \mathsf{sqn\text{-}init}^{\mathrm{ID}}_{\mathsf{S}} & \text{if } \mathsf{x} = \mathrm{SQN}^{\mathrm{ID}}_{\mathsf{S}} \end{cases}$$

The set of action labels is:

$$\mathcal{L} = \left\{ \mathtt{NewUser},\ \mathtt{NewNetwork}_i,\ \mathtt{UserMsg}_i,\ \mathtt{NetMsg}_j \mid i \in \mathbb{N} \right\}$$

We describe the available action labels in node $q = (\mathrm{ID}_1 : b_1, \ldots, \mathrm{ID}_l : b_l), (i_1 : a_1, \ldots, i_n : a_n)$. For every transition, we underline, in the new node q', the changes between q and q'.

- `NewUser` creates a new user with identity ID_{l+1}:

$$t \equiv \{\mathrm{ID}_{l+1}\}^{\mathsf{n}^{l+1}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} \qquad \sigma^{\mathsf{up}} = \epsilon \qquad q' = \Big((\mathrm{ID}_1 : b_1, \ldots, \mathrm{ID}_l : b_l, \underline{\mathrm{ID}_{l+1} : \mathsf{true}}), (i_1 : a_1, \ldots, i_n : a_n) \Big)$$

- For any $i \in \{1, \ldots, l\}$, $\mathtt{NewNetwork}_i$ creates a new service provider session communicating with ID_i:

$$t \equiv \mathsf{Nothing} \qquad \sigma^{\mathsf{up}} = \epsilon \qquad q' = ((\mathrm{ID}_1 : b_1, \ldots, \mathrm{ID}_l : b_l), (i_1 : a_1, \ldots, i_n : a_n, \underline{i : \mathsf{true}}))$$

- For every $i \in \{1,\dots,l\}$ such that b_i is true, $\mathtt{UserMsg}_i$ lets the adversary send a message to ID_i and get the user's output. Let $\mathrm{ID} = \mathrm{ID}_i$ and $t_{\mathsf{dec}} \equiv \mathsf{sdec}(\mathsf{x_{in}}, \mathsf{k}_{\mathrm{ID}})$, the action is defined by:

$$\mathsf{accept} \equiv \mathsf{geq}(\pi_2(t_{\mathsf{dec}}), \mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}} + 1) \qquad t \equiv \mathsf{if\ accept\ then\ senc}(\pi_1(t_{\mathsf{dec}}), \mathsf{k}_{\mathrm{ID}})\ \mathsf{else\ error}$$

$$\sigma^{\mathsf{up}} = \mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}} \mapsto \mathsf{if\ accept\ then}\ \pi_2(t_{\mathsf{dec}})\ \mathsf{else}\ \mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}$$

$$q' = \big((\mathrm{ID}_1 : b_1, \dots, \underline{\mathrm{ID}_i : \mathsf{false}}, \dots, \mathrm{ID}_l : b_l), (i_1 : a_1, \dots, i_n : a_n)\big)$$

  We set b_i to false in the node since ID_i sent its last message.
- For every $j \in \{1,\dots,n\}$, such that a_j is true, $\mathtt{NetMsg}_j$, lets the adversary send a message to the session i_j of the service provider and get its output. Let $\mathrm{ID} = \mathrm{ID}_{i_j}$, the action is defined by:

$$t \equiv \mathsf{if\ eq}(\mathsf{dec}(\mathsf{x_{in}}, \mathsf{sk}_{\mathrm{N}}), \mathrm{ID})\ \mathsf{then\ senc}(\langle \mathsf{n}^j\ , \mathrm{SQN}_{\mathsf{S}}^{\mathrm{ID}} \rangle, \mathsf{k}_{\mathrm{ID}})\ \mathsf{else\ UnknownId}$$

$$\sigma^{\mathsf{up}} = \mathrm{SQN}_{\mathsf{S}}^{\mathrm{ID}} \mapsto \mathsf{if\ eq}(\mathsf{dec}(\mathsf{x_{in}}, \mathsf{sk}_{\mathrm{N}}), \mathrm{ID})\ \mathsf{then}\ \mathrm{SQN}_{\mathsf{S}}^{\mathrm{ID}} + 1\ \mathsf{else}\ \mathrm{SQN}_{\mathsf{S}}^{\mathrm{ID}}$$

$$q' = \Big((\mathrm{ID}_1 : b_1, \dots, \mathrm{ID}_l : b_l), (i_1 : a_1, \dots, \underline{i_j : \mathsf{false}}, \dots, i_n : a_n)\Big)$$

  We set a_j to false in the state since the j-th session of the service provider sent its last message.

Remark that the set of available actions in a node $q \in \mathcal{Q}$ can be arbitrarily large, but remains finite. □

2.4.2 Computational Execution

We are now going to define what it means to execute a protocol. Let $\mathcal{M}_{\mathsf{c}}$ be a computational model, and $P = (\mathcal{Q}, \mathcal{L}, \mathsf{Vars}_\sigma, q_\epsilon, \phi_\epsilon, \sigma_\epsilon, \delta)$ be a protocol. We assume that $\mathcal{L}$ is equipped with an arbitrary non-ambiguous encoding $\widetilde{\cdot}$ into bit-strings. This is used to let the adversary choose the action it wants to execute, by writing the encoding of the action on its output tape.

When executing a protocol, at every step $i \in \mathbb{N}$, we let the adversary choose both the input message and the action to be executed. For this, we symbolically represent inputs and action choices using the reserved function symbols in $\mathcal{G}$ (which can be interpreted by any probabilistic polynomial-time function). More precisely:

- the i-th input is computed using the adversarial function symbols $g_i(\_)$.
- the i-th action to be executed is chosen using the adversarial function symbol $\mathsf{to}_i(\_)$. We do not force to_i to return only valid encodings of labels in $\mathcal{L}$. Instead, we interpret any invalid encoding as nop.

The function symbols $g_i(\_)$ and $\mathsf{to}_i(\_)$ are of arity $|\phi_\epsilon + i|$.

We define what is a symbolic frame and a symbolic state of P.

Definition 2.4. A symbolic frame is a finite sequence of terms in $\mathcal{T}(\mathcal{F}, \mathcal{N})$.

Definition 2.5. A symbolic state of P is a total function σ from variables Vars_σ to terms in $\mathcal{T}(\mathcal{F}, \mathcal{N})$.

Computational Trace Let $\eta \in \mathbb{N}$ and ρ_1, ρ_2 be two random tapes. The trace $\mathsf{trace}^P_{\mathcal{M}_{\mathsf{c}}}(1^\eta, \rho_1, \rho_2)$ of the execution of P with security parameter η on random tapes ρ_1, ρ_2 is the sequence:

$$\mathsf{trace}^P_{\mathcal{M}_{\mathsf{c}}}(1^\eta, \rho_1, \rho_2) = (q_i, \alpha_i, [\![\phi_i]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}}, [\![\sigma_i]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}})_{i \in \mathbb{N}}$$

where, for every $i \in \mathbb{N}$:

1. q_i is a node of $\mathcal{Q}$ and $q_0 = q_\epsilon$.
2. ϕ_i is a sequence of terms in $\mathcal{T}(\mathcal{F}, \mathcal{N})$ and $\phi_0 = \phi_\epsilon$.
3. σ_i is a symbolic state of P and $\sigma_0 = \sigma_\epsilon$.
4. α_i is such that $\widetilde{\alpha_i} = [\![\mathsf{to}_i(\phi_i)]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}}$ if $[\![\mathsf{to}_i(\phi_i)]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}}$ is a valid encoding of a label in $\delta^{\mathsf{q}}(q)$, and $\alpha_i = \mathsf{nop}$ otherwise. Let $(\alpha_i, t, \sigma^{\mathsf{up}}, q') \in \delta(q_i)$, then:

$$q_{i+1} = q' \qquad \theta_i = \sigma_i \cdot (\mathsf{x_{in}} \mapsto g_i(\phi_i)) \qquad \phi_{i+1} = \phi_i, t\theta_i$$

$$\forall \mathsf{x} \in \mathsf{Vars}_\sigma, \sigma_{i+1}(\mathsf{x}) = \begin{cases} \sigma^{\mathsf{up}}\theta_i & \text{if } \mathsf{x} \in \mathsf{dom}(\sigma^{\mathsf{up}}) \\ \sigma_i(\mathsf{x}) & \text{otherwise} \end{cases}$$

We also let $\widetilde{\phi}\text{-}\mathsf{trace}^P_{\mathcal{M}_\mathsf{c}}(\eta,\rho_1,\rho_2) = (\llbracket\phi_i\rrbracket^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}})_{i\in\mathbb{N}}$ and $\phi\text{-}\mathsf{s}\text{-}\mathsf{trace}^P_{\mathcal{M}_\mathsf{c}}(\eta,\rho_1,\rho_2) = (\phi_i)_{i\in\mathbb{N}}$.

Computational Execution A computational trace is with fixed η,ρ_1,ρ_2. The computational execution of P in $\mathcal{M}_\mathsf{c}$ is a sequence, indexed by the length i, of sequences, indexed by the security parameter η, of random variables representing the execution of the i-th first actions of P with security parameter η. Formally:

$$\mathsf{exec}^P_{\mathcal{M}_\mathsf{c}} = \left(\left(\left[\rho_1,\rho_2 \;:\; \widetilde{\phi}\text{-}\mathsf{trace}^P_{\mathcal{M}_\mathsf{c}}(\eta,\rho_1,\rho_2)(i)\right]\right)_{\eta\in\mathbb{N}}\right)_{i\in\mathbb{N}}$$

The order in which we introduce i,η,ρ_1,ρ_2 is important here. Since our model is only sound for a finite number of messages, we first introduce the length of the protocol execution i. Then, for a fixed length i, we see the execution of the i-th first messages of P as a sequences of random variables indexed by the security parameter (as we did in Remark 2.1).

It is not meaningful to compare protocols with different set of labels:

Definition 2.6. Two protocols P and Q are *compatible* if they have the same set of labels.

We can now state what it means for two protocol to be indistinguishable:

Definition 2.7. Let $\mathcal{M}_\mathsf{c}$ be a computational model. Two compatible protocols P and Q are indistinguishable in $\mathcal{M}_\mathsf{c}$, denoted by $P \approx_{\mathcal{M}_\mathsf{c}} Q$ iff for every $i \in \mathbb{N}$:

$$\mathsf{exec}^P_{\mathcal{M}_\mathsf{c}}(i) \approx_{\mathcal{M}_\mathsf{c}} \mathsf{exec}^Q_{\mathcal{M}_\mathsf{c}}(i)$$

2.4.3 Symbolic Execution

We now define a way of symbolically executing a protocol $P = (\mathcal{Q},\mathcal{L},\mathsf{Vars}_\sigma,q_\epsilon,\phi_\epsilon,\sigma_\epsilon,\delta)$. Instead of querying the computational model $\mathcal{M}_\mathsf{c}$ at every step to get the next action to be executed using $\llbracket\mathsf{to}_i(\phi_i)\rrbracket_{\mathcal{M}_\mathsf{c}}$, we fix the sequence of actions τ (the action trace) to be executed. For every action trace τ, this yields a symbolic frame of the same length. The symbolic execution of P at depth l is then the collection of all symbolic execution of P on any action trace of length l.

Action Traces $\mathcal{L}$ may be infinite, hence there may be an infinite number of action traces of length l. But because P is a finitely branching LTS, the symbolic execution of P at depth l contains a finite number of distinct symbolic frames. Therefore we can consider only a finite set of action traces to symbolically evaluates P at depth l, which we define below.

First, for every $q \in \mathcal{Q}$, we lift δ^q to any finite sequence of labels τ in the expected fashion, and we lift enbl_P to any finite sequence of labels τ:

$$\delta^\mathsf{q}(q,\epsilon) = q \qquad \delta^\mathsf{q}(q,\tau\cdot\alpha) = \delta^\mathsf{q}(\delta^\mathsf{q}(q,\tau),\alpha) \qquad \mathsf{enbl}_P(\tau) = \mathsf{enbl}_P(\delta^\mathsf{q}(q_\epsilon,\tau))$$

We now define the set of action traces of P of length l:

Definition 2.8. For every protocol P and $l \in \mathbb{N}$, we let $\mathsf{T}_l(P)$ be the set of P action traces of length l:

$$\mathsf{T}_0(P) = \{\epsilon\} \qquad \mathsf{T}_{l+1}(P) = \{\tau\cdot\alpha \mid \tau \in \mathsf{T}_l(P), \alpha \in \mathsf{enbl}_P(\tau)\}$$

For every protocol Q compatible with P, we also let $\mathsf{T}_l(P,Q) = \mathsf{T}_l(P) \cup \mathsf{T}_l(Q)$.

Proposition 2.2. *For every protocol P and for every $i \in \mathbb{N}$, $\mathsf{T}_i(P)$ is finite.*

Proof. This directly follows from the fact that P is a finitely branching LTS. ∎

Symbolic Executions We now define the symbolic execution of a protocol with a fixed action trace τ of length l. We do not require that τ be in $\mathsf{T}_l(P)$. Instead, as in a computational execution, we treat any invalid action label as nop.

Definition 2.9. Let $\tau = \alpha_0, \ldots, \alpha_{l-1}$ be a finite sequence of labels in $\mathcal{L}$. The symbolic trace $\mathsf{s\text{-}trace}^P(\tau)$ of the execution of $P = (\mathcal{Q}, \mathcal{L}, \mathsf{Vars}_\sigma, q_\epsilon, \phi_\epsilon, \sigma_\epsilon, \delta)$ on τ is the sequence:

$$\mathsf{s\text{-}trace}^P_\tau = (q_i, \phi_i, \sigma_i)_{0 \le i \le l}$$

where, for every $i \in \mathbb{N}$:

1. q_i is a node of $\mathcal{Q}$ and $q_0 = q_\epsilon$.
2. ϕ_i is a sequence of terms in $\mathcal{T}(\mathcal{F}, \mathcal{N})$ and $\phi_0 = \phi_\epsilon$.
3. σ_i is a symbolic state of P and $\sigma_0 = \sigma_\epsilon$.
4. Let $\alpha = \alpha_i$ if α_i is a valid encoding of a label in $\mathsf{enbl}_P(q)$, and $\alpha = \mathsf{nop}$ otherwise. Let $(\alpha, t, \sigma^{\mathsf{up}}, q') \in \delta(q_i)$, then:

$$q_{i+1} = q' \qquad \theta_i = \sigma_i \cdot (\mathsf{x_{in}} \mapsto g_i(\phi_i)) \qquad \phi_{i+1} = \phi_i, t\theta_i$$

$$\forall \mathsf{x} \in \mathsf{Vars}_\sigma, \sigma_{i+1}(\mathsf{x}) = \begin{cases} \sigma^{\mathsf{up}}\theta_i & \text{if } \mathsf{x} \in \mathsf{dom}(\sigma^{\mathsf{up}}) \\ \sigma_i(\mathsf{x}) & \text{otherwise} \end{cases}$$

We let $\phi\text{-}\mathsf{s\text{-}trace}^P_\tau = (\phi_i)_{0 \le i \le l}$ be the sequence of symbolic frames extracted from the symbolic trace $\mathsf{s\text{-}trace}^P(\tau)$ and $\phi^P_\tau = \phi_l$ be the final symbolic frame.

Soundness Theorem We can now state the soundness theorem linking computational and symbolic executions: given a computational model $\mathcal{M}_\mathsf{c}$ and two compatible protocols P and Q, if the formula $\phi^P_\tau \sim \phi^Q_\tau$ is valid in $\mathcal{M}_\mathsf{c}$ for every action trace $\tau \in \bigcup_l \mathsf{T}_l(P, Q)$, then P and Q are computational indistinguishable, i.e. $P \approx_{\mathcal{M}_\mathsf{c}} Q$.

Intuitively, this is because if $P \not\approx_{\mathcal{M}_\mathsf{c}} Q$, then there exist a depth l and an adversary $\mathcal{A}$ with a non-negligible advantage in distinguishing l steps of P from l steps of Q. Because $\mathsf{T}_l(P, Q)$ is finite, the adversary has only a finite number of choices of action trace. Moreover, the advantage of $\mathcal{A}$ against P and Q with an adaptive choice of action can be upper-bounded by the sum, over all possible choices of action trace τ, of the advantage of $\mathcal{A}$ against P and Q with fixed trace τ. Since $\mathcal{A}$'s advantage is non-negligible, and since the sum is finite, this implies that there exists some $\tau \in \mathsf{T}_l(P, Q)$ such that $\mathcal{A}$ has a non-negligible advantage in distinguishing P and Q with fixed trace τ.

Theorem 2.1. *Let P and Q be two compatible protocols and $\mathcal{M}_\mathsf{c}$ be a computational model. If, for every $l \in \mathbb{N}$ and $\tau \in \mathsf{T}_l(P, Q)$, we have $\mathcal{M}_\mathsf{c} \models \phi^P_\tau \sim \phi^Q_\tau$, then:*

$$P \approx_{\mathcal{M}_\mathsf{c}} Q$$

Proof. We prove this by contraposition. Assume that $P \not\approx_{\mathcal{M}_\mathsf{c}} Q$, by definition there exist a depth l and an adversary $\mathcal{A}$ such that $\mathcal{A}$ has a non-negligible probability of distinguishing $\mathsf{exec}^P_{\mathcal{M}_\mathsf{c}}(l)$ and $\mathsf{exec}^Q_{\mathcal{M}_\mathsf{c}}(l)$, i.e. the following quantity is not negligible:

$$\left| \begin{array}{l} \mathbf{Pr}\left(\rho_1, \rho_2 : \mathcal{A}(1^\eta, \widetilde{\phi}\text{-}\mathsf{trace}^P_{\mathcal{M}_\mathsf{c}}(\eta, \rho_1, \rho_2)(l), \rho_2) = 1\right) \\ - \ \mathbf{Pr}\left(\rho_1, \rho_2 : \mathcal{A}(1^\eta, \widetilde{\phi}\text{-}\mathsf{trace}^Q_{\mathcal{M}_\mathsf{c}}(\eta, \rho_1, \rho_2)(l), \rho_2) = 1\right) \end{array} \right| \tag{2.2}$$

Trace Events For every $\tau = \alpha_0, \ldots, \alpha_{l-1} \in \mathsf{T}_i(P, Q)$, we want to define the family of events $(E^\eta_\tau)_\eta$:

E^η_τ : "with security parameter η, the trace of actions chosen during the execution of l steps of P is τ"

To define E^η_τ, we just need to define, for every $0 \le i < l$, the event:

$E^\eta_{\tau,i}$: "with security parameter η, the i-th action chosen during the execution of P is α_i"

Let $\phi\text{-}\mathsf{s\text{-}trace}^P_\tau = (\phi_i)_{0 \le i \le l}$. We have two cases:

- If α_l is an enabled actions of P, $E^{\eta}_{\tau,l}$ is the event:

$$\left[\rho_1, \rho_2 : [\![\mathsf{to}_l(\phi_l)]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}} = \widetilde{\alpha_l}\right]$$

- If α_l is not enabled, it is treated as nop. Therefore we let $\mathsf{to}_l(\phi_l)$ be any value which is treated as nop by the computational semantics. $E^{\eta}_{\tau,l}$ is the event:

$$\left[\rho_1, \rho_2 : [\![\mathsf{to}_l(\phi_l)]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}} \notin \left\{\widetilde{\alpha} \mid \alpha \in \mathsf{enbl}_P(\tau_{|1,\ldots,l})\right\}\right]$$

In general, $E^{\eta}_{\tau,l}$ is the union of both cases above:

$$\left[\rho_1, \rho_2 : \begin{array}{ll} & \left(\alpha_l \in \mathsf{enbl}_P(\tau_{|1,\ldots,l}) \wedge [\![\mathsf{to}_l(\phi_l)]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}} = \widetilde{\alpha_l}\right) \\ \vee & \left(\alpha_l \notin \mathsf{enbl}_P(\tau_{|1,\ldots,l}) \wedge [\![\mathsf{to}_l(\phi_l)]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}} \notin \left\{\widetilde{\alpha} \mid \alpha \in \mathsf{enbl}_P(\tau_{|1,\ldots,l})\right\}\right) \end{array}\right]$$

For every τ and η, the following two random variables are the same:

- executing l steps of P in $\mathcal{M}_\mathsf{c}$ with security parameter η conditioned on E^{η}_{τ}.
- sampling from ϕ^P_τ in $\mathcal{M}_\mathsf{c}$ with security parameter η.

Formally we have:

$$\left[\rho_1, \rho_2 : \widetilde{\phi}\text{-}\mathsf{trace}^P_{\mathcal{M}_\mathsf{c}}(\eta, \rho_1, \rho_2)(l) \mid E^{\eta}_{\tau}\right] = \left[\rho_1, \rho_2 : [\![\phi^P_\tau]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}\right] \tag{2.3}$$

We also define the symmetrical events:

E'^{η}_{τ} : "with security parameter η, the trace of actions chosen during the execution of l steps of Q is τ"

Which satisfies:

$$\left[\rho_1, \rho_2 : \widetilde{\phi}\text{-}\mathsf{trace}^Q_{\mathcal{M}_\mathsf{c}}(\eta, \rho_1, \rho_2)(l) \mid E'^{\eta}_{\tau}\right] = \left[\rho_1, \rho_2 : [\![\phi^Q_\tau]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}\right] \tag{2.4}$$

Upper-Bounding $\mathcal{A}$'s Advantage To conclude, we just need to upper-bound $\mathcal{A}$'s advantage in distinguishing P and Q. Using (2.3), we get that for every η:

$$\mathbf{Pr}\left(\rho_1, \rho_2 : \mathcal{A}(1^\eta, \widetilde{\phi}\text{-}\mathsf{trace}^P_{\mathcal{M}_\mathsf{c}}(\eta, \rho_1, \rho_2)(l), \rho_2) = 1\right) = \sum_{\tau \in \mathsf{T}_i(P,Q)} \mathbf{Pr}\left(\rho_1, \rho_2 : \mathcal{A}(1^\eta, [\![\phi^P_\tau]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}, \rho_2) = 1\right) \times \mathbf{Pr}(E^{\eta}_{\tau})$$

Similarly, using (2.4) we get that:

$$\mathbf{Pr}\left(\rho_1, \rho_2 : \mathcal{A}(1^\eta, \widetilde{\phi}\text{-}\mathsf{trace}^Q_{\mathcal{M}_\mathsf{c}}(\eta, \rho_1, \rho_2)(l), \rho_2) = 1\right) = \sum_{\tau \in \mathsf{T}_i(P,Q)} \mathbf{Pr}\left(\rho_1, \rho_2 : \mathcal{A}(1^\eta, [\![\phi^Q_\tau]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}, \rho_2) = 1\right) \times \mathbf{Pr}(E'^{\eta}_{\tau})$$

Since $\mathbf{Pr}(E^{\eta}_{\tau}) \leq 1$, $\mathbf{Pr}(E'^{\eta}_{\tau}) \leq 1$ and bounding the absolute value of the sum by the sum of the absolute values, we get that (2.2) is upper-bounded by:

$$\sum_{\tau \in \mathsf{T}_i(P,Q)} \left|\mathbf{Pr}\left(\rho_1, \rho_2 : \mathcal{A}(1^\eta, [\![\phi^P_\tau]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}, \rho_2) = 1\right) - \mathbf{Pr}\left(\rho_1, \rho_2 : \mathcal{A}(1^\eta, [\![\phi^Q_\tau]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}, \rho_2) = 1\right)\right|$$

Since (2.2) is non-negligible, the quantity above is non-negligible. Using Proposition 2.2, we know that $\mathsf{T}_i(P,Q)$ is finite. A finite sum is non-negligible iff one of its terms is non-negligible. Hence there exists $\tau \in \mathsf{T}_i(P,Q)$ such that:

$$\left|\mathbf{Pr}\left(\rho_1, \rho_2 : \mathcal{A}(1^\eta, [\![\phi^P_\tau]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}, \rho_2) = 1\right) - \mathbf{Pr}\left(\rho_1, \rho_2 : \mathcal{A}(1^\eta, [\![\phi^Q_\tau]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}, \rho_2) = 1\right)\right|$$

is non-negligible, which implies that $\mathcal{M}_\mathsf{c} \not\models \phi^P_\tau \sim \phi^Q_\tau$. ■

Completeness We would like to state the converse statement, i.e. that if $P \approx_{\mathcal{M}_\mathsf{c}} Q$ then for every $l \in \mathbb{N}$ and $\tau \in \mathsf{T}_l(P,Q)$, $\mathcal{M}_\mathsf{c} \models \phi^P_\tau \sim \phi^Q_\tau$. This is quite intuitive: if there is an attack against P and Q for a fixed trace τ, then there is an adaptive attack against P and Q (we simply let the adversary pick the trace τ). The problem is that the adversary's action choices are "stored" in the computational model $\mathcal{M}_\mathsf{c}$, in the interpretation of the function symbols to_i. Therefore we cannot use the same computational model $\mathcal{M}_\mathsf{c}$. Formally:

Proposition 2.3. *For every compatible protocols P and Q that do not use the function symbols $\{\mathsf{to}_l \mid l \in \mathbb{N}\}$, for every computational model $\mathcal{M}_\mathsf{c}$, if there exists $l \in \mathbb{N}$ and $\tau \in \mathsf{T}_l(P,Q)$ such that:*

$$\mathcal{M}_\mathsf{c} \not\models \phi^P_\tau \sim \phi^Q_\tau$$

then there exists a computational model $\mathcal{M}_\mathsf{c}{}'$, which may differ from $\mathcal{M}_\mathsf{c}$ only on the interpretation of $\{\mathsf{to}_l \mid l \in \mathbb{N}\}$, such that $P \not\approx_{\mathcal{M}_\mathsf{c}{}'} Q$.

Proof. Take an adversary $\mathcal{A}$, a depth l and an action trace $\tau \in \mathsf{T}_l(P,Q)$ such that $\mathcal{A}$ has a non-negligible probability of distinguishing between ϕ^P_τ and ϕ^Q_τ, i.e.:

$$\left|\mathbf{Pr}\left(\rho_1,\rho_2 : \mathcal{A}(1^\eta, [\![\phi^P_\tau]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}, \rho_2) = 1\right) \; - \; \mathbf{Pr}\left(\rho_1,\rho_2 : \mathcal{A}(1^\eta, [\![\phi^Q_\tau]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}, \rho_2) = 1\right)\right|$$

is non-negligible. We simply let $\mathcal{M}_\mathsf{c}{}'$ be the computational model $\mathcal{M}_\mathsf{c}$ where we modify the interpretation of $\{\mathsf{to}_i \mid 0 \leq i < l\}$ by having $[\![\mathsf{to}_i]\!]$ be the machine that ignores its arguments and always returns $\widetilde{\alpha_i}$ where α_i is the i-th action in τ. Using the fact that P does not use the function symbols $\{\mathsf{to}_l \mid l \in \mathbb{N}\}$, we can check that for every η, ρ_1, ρ_2:

$$[\![\phi^P_\tau]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}} = \widetilde{\phi}\text{-}\mathsf{trace}^P_{\mathcal{M}_\mathsf{c}{}'}(\eta,\rho_1,\rho_2)(l)$$

Similarly:

$$[\![\phi^Q_\tau]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}} = \widetilde{\phi}\text{-}\mathsf{trace}^Q_{\mathcal{M}_\mathsf{c}{}'}(\eta,\rho_1,\rho_2)(l)$$

We deduce that $\mathcal{A}$ has a non-negligible probability of distinguishing the computational executions of P and Q in $\mathcal{M}_\mathsf{c}{}'$, i.e. $P \not\approx_{\mathcal{M}_\mathsf{c}{}'} Q$. ■

2.5 Axioms

When studying two protocols P and Q, we cannot just show that P and Q are indistinguishable in some fixed computational model $\mathcal{M}_\mathsf{c}$ for several reasons:

- First, we would need to fully specify the implementation of every protocol function symbols, e.g., how is the pair implemented, how projections behave on ill-formed inputs etc. Not only would this be very tedious, but it would also be a waste of time. For example, if there is no way for the adversary to have the protocols' agents try to compute projections of ill-formed pairs, then we do not need to know how projections handle bad inputs.
- This is too restrictive: if we show security in some model $\mathcal{M}_\mathsf{c}$, and we later decide to change the implementation of some functions, we need to redo the whole proof.
- Lastly, we do not know the implementation of the adversarial function symbols. We need to let them be any probabilistic polynomial-time Turing machine.

Instead, we use an axiomatic approach, restricting the models that have to be considered using *axioms*, which are formulas of the logic. Then, given a set of axioms Ax (potentially infinite) and two compatible protocols P and Q, if we show that for every l and $\tau \in \mathsf{T}_l(P,Q)$, the formula $\phi^P_\tau \not\sim \phi^Q_\tau$ is inconsistent with Ax, then we know that there exists no model $\mathcal{M}$ (hence no computational model either) satisfying Ax such that $\mathcal{M} \models \phi^P_\tau \not\sim \phi^Q_\tau$. Hence, using the soundness Theorem 2.1, we know that $P \approx_{\mathcal{M}_\mathsf{c}} Q$ in every computational model $\mathcal{M}_\mathsf{c}$ such that the axioms Ax hold.

Structural and Implementation Axioms We define the axioms that we will need later. Axioms can be of two kind:

- *Structural axioms* represent properties that hold in every computational model. This includes axioms such as the symmetry of $\sim$, or properties of the $\mathsf{if\_then\_else\_}$ function symbol (since its interpretation is fixed).
- *Implementation axioms* reflect implementation assumptions, such as the functional correctness of the pair and projections (e.g. $\pi_1(\langle u\,,\,v\rangle) = u$), or cryptographic assumptions on the security primitives (e.g. EUF-CMA or IND-CPA).

All our axioms are universally quantified Horn clauses or recursive schemata of ground Horn clauses. Therefore, to show the unsatisfiability of $\mathsf{Ax} \wedge \vec{u} \not\sim \vec{v}$ (where Ax is a given set of axioms and $\vec{u} \not\sim \vec{v}$ is a ground formula), we use resolution with a negative strategy (which is complete, see [CL73]). As all axioms are Horn clauses, a proof by resolution with a negative strategy can be seen as a proof tree where each node is indexed by the axiom of Ax used at this resolution step. Hence, axioms will be given as inference rules (where variables are implicitly universally quantified).

2.5.1 Structural Axioms

Almost all the axioms in this subsection have been introduced in the literature, see [BCL14, CK17, BC16].

Equality Computational indistinguishability is an equivalence relation (i.e. reflexive, symmetric and transitive). But we can observe that it is not a congruence. E.g. take a computational model $\mathcal{M}_{\mathsf{c}}$, we know that two names n and n' are indistinguishable (since they are interpreted as independent uniform random sampling in $\{0,1\}^\eta$), and n is indistinguishable from itself. Therefore:

$$\mathcal{M}_{\mathsf{c}} \models \mathsf{n} \sim \mathsf{n}' \qquad\qquad \mathcal{M}_{\mathsf{c}} \models \mathsf{n} \sim \mathsf{n}$$

But the formula $\mathsf{n}, \mathsf{n} \sim \mathsf{n}', \mathsf{n}$ is not valid in $\mathcal{M}_{\mathsf{c}}$. Indeed, there is a simple PPTM that can distinguish between n, n and n', n: simply test whether the two arguments are equal, if so return 1 and otherwise return 0. Then, with overwhelming probability, this machine will guess from which distribution its input was sampled from.

Even though $\sim$ is not a congruence, we can get a congruence from it: if $\mathsf{eq}(s,t) \sim \mathsf{true}$ holds in all models then, using the semantics of $\mathsf{eq}(\_,\_)$, in every computational model $\mathcal{M}_{\mathsf{c}}$, $[\![s]\!]$ and $[\![t]\!]$ are identical except for a negligible number of samplings. Hence all properties of equality hold: this relation is symmetric, reflexive, transitive and closed under function applications. Moreover, we can replace any occurrence of s by t in a formula without changing its semantics with respect to computational indistinguishability. We let $s = t$ be the shorthand for $\mathsf{eq}(s,t) \sim \mathsf{true}$, and we introduce the axioms:

$$\frac{}{u = u}\ \text{=-refl} \qquad \frac{v = u}{u = v}\ \text{=-sym} \qquad \frac{u = w \qquad w = v}{u = v}\ \text{=-trans}$$

$$\frac{u_0 = v_0 \qquad \dots \qquad u_n = v_n}{f(u_0,\dots,u_n) = f(v_0,\dots,v_n)}\ \text{=-subst} \quad (f \in \mathcal{F}) \qquad \frac{\vec{u}, t \sim \vec{v} \qquad s = t}{\vec{u}, s \sim \vec{v}}\ \mathsf{Equ}$$

Finally, we have equality axioms reflecting properties of the function symbols with a fixed interpretation, which are given in Figure 2.1. When writing equality axioms, we usually omit the over-line: e.g., we write $\mathsf{eq}(x,x)$ instead of $\overline{\mathsf{eq}(x,x)}$.

Most of these axioms deal with the $\mathsf{if\_then\_else\_}$ function symbols. We give a quick informal description: E_1 contains properties of zero and equality; E_2 and E_3 contain, respectively, the homomorphism properties and simplification rules of the $\mathsf{if\_then\_else\_}$; and E_4 allows to change the order in which conditional tests are performed.

Other Axioms We now give an informal description of the structural axioms given in Figure 2.2 that we have not introduced yet:

- Perm allows to change the terms order, using the same permutation π on both sides of $\sim$.

$$\frac{u_{\pi(1)},\dots,u_{\pi(n)} \sim v_{\pi(1)},\dots,v_{\pi(n)}}{u_1,\dots,u_n \sim v_1,\dots,v_n}\ \mathsf{Perm}$$

$$\mathsf{E}_1 \begin{cases} \mathbf{0}(\mathbf{0}(x)) = \mathbf{0}(x) \\ \mathsf{eq}(x,x) = \mathsf{true} \end{cases}$$

$$\mathsf{E}_2 \begin{cases} f(\vec{u}, \mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ y, \vec{v}) = \mathsf{if}\ b\ \mathsf{then}\ f(\vec{u},x,\vec{v})\ \mathsf{else}\ f(\vec{u},y,\vec{v}) & (f \in \mathcal{F}_{\setminus \mathsf{if}}) \\ \mathsf{if}\ (\mathsf{if}\ b\ \mathsf{then}\ a\ \mathsf{else}\ c)\ \mathsf{then}\ x\ \mathsf{else}\ y = \mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ x\ \mathsf{else}\ y)\ \mathsf{else}\ (\mathsf{if}\ c\ \mathsf{then}\ x\ \mathsf{else}\ y) \end{cases}$$

$$\mathsf{E}_3 \begin{cases} \mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ x = x \\ \mathsf{if}\ \mathsf{true}\ \mathsf{then}\ x\ \mathsf{else}\ y = x \\ \mathsf{if}\ \mathsf{false}\ \mathsf{then}\ x\ \mathsf{else}\ y = y \\ \mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ y)\ \mathsf{else}\ z = \mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ z \\ \mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ (\mathsf{if}\ b\ \mathsf{then}\ y\ \mathsf{else}\ z) = \mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ z \end{cases}$$

$$\mathsf{E}_4 \begin{cases} \mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ x\ \mathsf{else}\ y)\ \mathsf{else}\ z = \mathsf{if}\ a\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ z)\ \mathsf{else}\ (\mathsf{if}\ b\ \mathsf{then}\ y\ \mathsf{else}\ z) \\ \mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ (\mathsf{if}\ a\ \mathsf{then}\ y\ \mathsf{else}\ z) = \mathsf{if}\ a\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ y)\ \mathsf{else}\ (\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ z) \end{cases}$$

Figure 2.1: Equality Axioms $\mathsf{E}_1, \mathsf{E}_2, \mathsf{E}_3, \mathsf{E}_4$

$$\frac{}{u = u}\ =\text{-refl} \qquad \frac{v = u}{u = v}\ =\text{-sym} \qquad \frac{u = w \qquad w = v}{u = v}\ =\text{-trans} \qquad \frac{u_0 = v_0 \qquad \ldots \qquad u_n = v_n}{f(u_0,\ldots,u_n) = f(v_0,\ldots,v_n)}\ =\text{-subst}$$

$$\frac{\vec{u}, t \sim \vec{v} \qquad s = t}{\vec{u}, s \sim \vec{v}}\ \mathsf{Equ} \qquad \frac{u_{\pi(1)},\ldots,u_{\pi(n)} \sim v_{\pi(1)},\ldots,v_{\pi(n)}}{u_1,\ldots,u_n \sim v_1,\ldots,v_n}\ \mathsf{Perm} \qquad \frac{\vec{u}, s \sim \vec{v}, t}{\vec{u} \sim \vec{v}}\ \mathsf{Restr}$$

$$\frac{\vec{u}_1, \vec{v}_1 \sim \vec{u}_1, \vec{v}_2}{f(\vec{u}_1), \vec{v}_1 \sim f(\vec{u}_2), \vec{v}_2}\ \mathsf{FA} \qquad \frac{\vec{u}, s \sim \vec{v}, t}{\vec{u}, s, s \sim \vec{v}, t, t}\ \mathsf{Dup} \qquad \frac{}{\vec{u} \sim \vec{u}}\ \mathsf{Refl} \qquad \frac{\vec{v} \sim \vec{u}}{\vec{u} \sim \vec{v}}\ \mathsf{Sym}$$

$$\frac{\vec{u} \sim \vec{w} \qquad \vec{w} \sim \vec{v}}{\vec{u} \sim \vec{v}}\ \mathsf{Trans} \qquad \frac{\vec{u} \sim \vec{v}}{\vec{u}, \mathsf{n} \sim \vec{v}, \mathsf{n}'}\ \mathsf{Fresh} \quad \text{when } \mathsf{n} \notin \mathsf{st}(\vec{u}) \text{ and } \mathsf{n}' \notin \mathsf{st}(\vec{v})$$

$$\frac{}{\vec{u} \sim \vec{u}\alpha}\ \alpha\text{-equ} \quad \text{when } \alpha \text{ is an injective renaming of names in } \mathcal{N}$$

$$\frac{}{\mathsf{eq}(t,\mathsf{n}) = \mathsf{false}}\ =\text{-ind} \quad \text{when } \mathsf{n} \notin \mathsf{st}(t) \qquad \frac{\vec{u}, C\,[\mathsf{if}\ \mathsf{eq}(s,t)\ \mathsf{then}\ C_0[t]\ \mathsf{else}\ w] \sim \vec{v}}{\vec{u}, C\,[\mathsf{if}\ \mathsf{eq}(s,t)\ \mathsf{then}\ C_0[s]\ \mathsf{else}\ w] \sim \vec{v}}\ \mathsf{IFT}$$

$$\frac{\vec{w}, b, (u_i)_i \sim \vec{w}', b', (u'_i)_i \qquad \vec{w}, b, (v_i)_i \sim \vec{w}', b', (v'_i)_i}{\vec{w}, (\mathsf{if}\ b\ \mathsf{then}\ u_i\ \mathsf{else}\ v_i)_i \sim \vec{w}', (\mathsf{if}\ b'\ \mathsf{then}\ u'_i\ \mathsf{else}\ v'_i)_i}\ \mathsf{CS}$$

Conventions: π is a permutation of $\{1,\ldots,n\}$ and $f \in \mathcal{F}$.

Figure 2.2: Some Structural Axioms.

- Restr is a strengthening axiom, stating that to prove that $\vec{u} \sim \vec{v}$, it is sufficient to show the stronger property $\vec{u}, s \sim \vec{v}, t$.

$$\frac{\vec{u}, s \sim \vec{v}, t}{\vec{u} \sim \vec{v}}\ \mathsf{Restr}$$

- The function application axiom FA states that to prove that two images (by $f \in \mathcal{F}$) are indistinguishable, it is sufficient to show that the arguments are indistinguishable.

$$\frac{\vec{u}_1, \vec{v}_1 \sim \vec{u}_1, \vec{v}_2}{f(\vec{u}_1), \vec{v}_1 \sim f(\vec{u}_2), \vec{v}_2}\ \mathsf{FA}$$

- Dup states that giving twice the same value to an adversary is equivalent to giving it only once.

$$\frac{\vec{u}, s \sim \vec{v}, t}{\vec{u}, s, s \sim \vec{v}, t, t}\ \mathsf{Dup}$$

- Refl, Sym and Trans states that indistinguishability is a reflexive, symmetrical and transitive relation.

$$\frac{}{\vec{u} \sim \vec{u}}\ \mathsf{Refl} \qquad \frac{\vec{v} \sim \vec{u}}{\vec{u} \sim \vec{v}}\ \mathsf{Sym} \qquad \frac{\vec{u} \sim \vec{w} \qquad \vec{w} \sim \vec{v}}{\vec{u} \sim \vec{v}}\ \mathsf{Trans}$$

- Fresh states that giving a value uniformly sampled at random and independent from the rest of the distribution is useless. We guarantee that n is independent from $\vec{u}$ by requiring that n does not appear in $\vec{u}$'s subterms (and similarly for n' and $\vec{v}$). By consequence, this is not a universally quantified axiom. Instead, this is a recursive infinite set of axioms, one for each *ground* formula satisfying the side-condition.

$$\frac{\vec{u} \sim \vec{v}}{\vec{u}, \mathsf{n} \sim \vec{v}, \mathsf{n}'}\ \mathsf{Fresh} \quad \text{when } \mathsf{n} \notin \mathsf{st}(\vec{u}) \text{ and } \mathsf{n}' \notin \mathsf{st}(\vec{v})$$

- The α-equ axioms allow to rename all the names appearing in $\vec{u}$, using an injective renaming α:

$$\frac{}{\vec{u} \sim \vec{u}\alpha}\ \alpha\text{-equ} \quad \text{when } \alpha \text{ is an injective renaming of names in } \mathcal{N}$$

- =-ind is a axiom schema stating that, if t is independent from a uniform random sampling n, then t is never equal to n, except for a negligible number of samplings.

$$\frac{}{\mathsf{eq}(t, \mathsf{n}) = \mathsf{false}}\ =\text{-ind} \quad \text{when } \mathsf{n} \notin \mathsf{st}(t)$$

- The IFT axioms allows to replace a term s by a term t if it appears in the then branch of a $\mathsf{eq}(s,t)$ conditional. Again, this is an axiom schema.

$$\frac{\vec{u}, C\left[\mathsf{if}\ \mathsf{eq}(s,t)\ \mathsf{then}\ C_0[t]\ \mathsf{else}\ w\right] \sim \vec{v}}{\vec{u}, C\left[\mathsf{if}\ \mathsf{eq}(s,t)\ \mathsf{then}\ C_0[s]\ \mathsf{else}\ w\right] \sim \vec{v}}\ \mathsf{IFT}$$

- The CS axioms states that in order to show that:

$$\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v \sim \mathsf{if}\ b'\ \mathsf{then}\ u'\ \mathsf{else}\ v'$$

  it is sufficient to show that the then branches and the else branches are indistinguishable, *when giving to the adversary the value of the conditional* (i.e. b on the left and b' on the right). We can do better, by considering simultaneously several terms starting with the same conditional. We also allow some terms $\vec{w}$ and $\vec{w}'$ on the left and right to stay untouched.

$$\frac{\vec{w}, b, (u_i)_i \sim \vec{w}', b', (u'_i)_i \qquad \vec{w}, b, (v_i)_i \sim \vec{w}', b', (v'_i)_i}{\vec{w}, (\mathsf{if}\ b\ \mathsf{then}\ u_i\ \mathsf{else}\ v_i)_i \sim \vec{w}', (\mathsf{if}\ b'\ \mathsf{then}\ u'_i\ \mathsf{else}\ v'_i)_i}\ \mathsf{CS}$$

Remark 2.2. In the CS axioms, we need to give the conditional b to the adversary. For example, assume that $\mathcal{F}_\mathsf{p}$ contains two constant function symbols one and zero. Then, in every computational model $\mathcal{M}_\mathsf{c}$:

$$\mathcal{M}_\mathsf{c} \models \mathsf{zero} \sim \mathsf{zero} \qquad \mathcal{M}_\mathsf{c} \models \mathsf{one} \sim \mathsf{one}$$

But if $\mathcal{M}_\mathsf{c}$ is such that zero's interpretation is different from one's interpretation, then:

$$\mathcal{M}_\mathsf{c} \not\models \mathsf{if\ true\ then\ zero\ else\ one} \sim \mathsf{if\ false\ then\ zero\ else\ one}$$ □

The conjunction of the equality axioms in Figure 2.1 and the axioms in Figure 2.2 form the set of structural axioms $\mathsf{Ax}_{\mathsf{struct}}$.

Definition 2.10. We let $\mathsf{Ax}_{\mathsf{struct}}$ be the union of sets of axioms in Figure 2.1 and Figure 2.2.

Structural axioms are valid in all computational models.

Proposition 2.4. *The axioms $\mathsf{Ax}_{\mathsf{struct}}$ are valid in all computational models.*

Proof. The axioms in Figure 2.1 and the four first axioms of Figure 2.2 are all immediate properties of the function symbols interpretations in any computational model.

All the remaining axioms are proved using the same kind of argument: given a adversary breaking the conclusion, we show that there exists an adversary breaking one of the premises.

- The axioms $\mathsf{Perm}, \mathsf{Restr}, \mathsf{Dup}$ are very similar. We only detail the proof for one of them, Dup.
  Assume a winning adversary $\mathcal{A}$ against $\vec{u}, s, s \sim \vec{v}, t, t$, we can define an adversary $\mathcal{B}$ against $\vec{u}, s \sim \vec{v}, t$ has follows: on input $\vec{w}, x$, return $\mathcal{A}(\vec{w}, x, x)$. The advantage of $\mathcal{B}$ against $\vec{u}, s \sim \vec{v}, t$ is exactly the advantage of $\mathcal{A}$ against $\vec{u}, s, s \sim \vec{v}, t, t$, which is non-negligible by hypothesis.
- Assume that $s = t$ holds in every computational model, and that there exists a winning adversary $\mathcal{A}$ against $\vec{u}, s \sim \vec{v}$. Recall that $s = t$ is the formula $\mathsf{eq}(s,t) \sim \mathsf{true}$. Since eq is interpreted as actual equality, we know that in every computational model, $[\![s]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}} = [\![t]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}$ except for a negligible subset of random tapes ρ_1, ρ_2 of measure $m_\rho(\eta)$. By consequence, the advantage of $\mathcal{A}$ against $\vec{u}, t \sim \vec{v}$ is the advantage of $\mathcal{A}$ against $\vec{u}, s \sim \vec{v}$, plus or minus the negligible quantity $m_\rho(\eta)$. Hence $\mathcal{A}$ has a non-negligible advantage against $\vec{u}, t \sim \vec{v}$.
- For the FA axiom, assume a winning adversary $\mathcal{A}$ against $f(\vec{u}_1), \vec{v}_1 \sim f(\vec{u}_2), \vec{v}_2$. Let $\mathcal{M}^f_c$ be the Turing machine used in $\mathcal{M}_\mathsf{c}$ to define $[\![f]\!]^\mathsf{s}_{\mathcal{M}_\mathsf{c}}$ semantics, where s is the type of $\vec{u}_1$.
  We define an adversary $\mathcal{B}$ against $\vec{u}_1, \vec{v}_1 \sim \vec{u}_1, \vec{v}_2$ as follows: on input $\vec{w}_1, \vec{w}_2$, compute $\mathcal{M}^f_c(\vec{w}_1)$, storing the result in x, and then return $\mathcal{A}(x, \vec{w}_2)$ ($\mathcal{M}^f_c$ runs in polynomial-time, therefore $\mathcal{B}$ is still polynomial-time).
  The advantage of $\mathcal{B}$ against $\vec{u}_1, \vec{v}_1 \sim \vec{u}_1, \vec{v}_2$ is exactly the advantage of $\mathcal{A}$ against $f(\vec{u}_1), \vec{v}_1 \sim f(\vec{u}_2), \vec{v}_2$, which is non-negligible by hypothesis.
- Refl is obvious: no adversary can distinguish between two identical distributions.
- Sym follows from the fact that the definition of computational indistinguishability is symmetrical.
- For Trans, take a winning adversary $\mathcal{A}$ against $\vec{u} \sim \vec{v}$. Using the triangular inequality:

$$\begin{aligned}
&\left|\mathbf{Pr}\left(\rho_1,\rho_2 : \mathcal{A}(1^\eta, [\![\vec{u}]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}, \rho_2)\right) - \mathbf{Pr}\left(\rho_1,\rho_2 : \mathcal{A}(1^\eta, [\![\vec{v}]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}, \rho_2)\right)\right| \leq \\
&\quad \left|\mathbf{Pr}\left(\rho_1,\rho_2 : \mathcal{A}(1^\eta, [\![\vec{u}]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}, \rho_2)\right) - \mathbf{Pr}\left(\rho_1,\rho_2 : \mathcal{A}(1^\eta, [\![\vec{w}]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}, \rho_2)\right)\right| \\
&+ \left|\mathbf{Pr}\left(\rho_1,\rho_2 : \mathcal{A}(1^\eta, [\![\vec{w}]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}, \rho_2)\right) - \mathbf{Pr}\left(\rho_1,\rho_2 : \mathcal{A}(1^\eta, [\![\vec{v}]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}, \rho_2)\right)\right|
\end{aligned}$$

  Since the left quantity is non-negligible, one of the two quantities on the right must be non-negligible. This shows that $\mathcal{A}$ is winning against $\vec{u} \sim \vec{w}$ or $\vec{w} \sim \vec{v}$.
- For Fresh, given a winning adversary against $\vec{u}, \mathsf{n} \sim \vec{v}, \mathsf{n}'$, we build a winning adversary $\mathcal{B}$ against $\vec{u} \sim \vec{v}$: $\mathcal{B}$ simply samples the uniform random value itself before calling $\mathcal{A}$.
- For $=$-ind, using the independence, we can upper-bound the probability of collision by $1/2^\eta$, which is negligible.
- For the IFT axiom, we just need to observe that in every computational model $\mathcal{M}_\mathsf{c}$, the distributions:

$$[\![\vec{u}, C\,[\mathsf{if}\ \mathsf{eq}(s,t)\ \mathsf{then}\ C_0[s]\ \mathsf{else}\ w]]\!]_{\mathcal{M}_\mathsf{c}} \qquad \text{and} \qquad [\![\vec{u}, C\,[\mathsf{if}\ \mathsf{eq}(s,t)\ \mathsf{then}\ C_0[t]\ \mathsf{else}\ w]]\!]_{\mathcal{M}_\mathsf{c}}$$

  are the same except for a negligible number of samplings. Hence any winning adversary against the conclusion is a winning adversary against the premise.
- It only remain to show that CS is valid. Let $\mathcal{M}_\mathsf{c}$ be a computational mode, and $\mathcal{A}$ be a winning adversary against:

$$\underbrace{\vec{w}, (\mathsf{if}\ b\ \mathsf{then}\ u_i\ \mathsf{else}\ v_i)_i}_{\vec{s}} \sim \underbrace{\vec{w}', (\mathsf{if}\ b'\ \mathsf{then}\ u'_i\ \mathsf{else}\ v'_i)_i}_{\vec{t}}$$

  in $\mathcal{M}_\mathsf{c}$. We let $p_{\mathsf{l},\mathsf{l}}, p_{\mathsf{r},\mathsf{l}}, p_{\mathsf{l},\mathsf{r}}$ and $p_{\mathsf{r},\mathsf{r}}$ be the quantities:

$$\begin{aligned}
p_{\mathsf{l},\mathsf{l}} &: \mathbf{Pr}\big(\rho_1,\rho_2 : \mathcal{A}(1^\eta, [\![\vec{s}]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}, \rho_2) \wedge \ [\![b]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}\big) \\
p_{\mathsf{r},\mathsf{l}} &: \mathbf{Pr}\big(\rho_1,\rho_2 : \mathcal{A}(1^\eta, [\![\vec{s}]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}, \rho_2) \wedge \neg[\![b]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}\big) \\
p_{\mathsf{l},\mathsf{r}} &: \mathbf{Pr}\big(\rho_1,\rho_2 : \mathcal{A}(1^\eta, [\![\vec{t}]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}, \rho_2) \wedge \ [\![b']\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}\big) \\
p_{\mathsf{r},\mathsf{r}} &: \mathbf{Pr}\big(\rho_1,\rho_2 : \mathcal{A}(1^\eta, [\![\vec{t}]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}, \rho_2) \wedge \neg[\![b']\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}\big)
\end{aligned}$$

We define the adversary $\mathcal{B}_{\mathsf{t}}$ against $\vec{w}, b, (u_i)_i \sim \vec{w}', b', (u'_i)_i$:

$$\mathcal{B}_{\mathsf{t}}(\vec{x}, a, (m_i)_i) = \begin{cases} \mathcal{A}(\vec{x}, (m_i)_i) & \text{if } a = 1 \\ 1 & \text{otherwise} \end{cases}$$

Then:

$$\mathbf{Pr}\left(\rho_1, \rho_2 \,:\, \mathcal{B}_{\mathsf{t}}(1^\eta, [\![\vec{w}, b, (u_i)_i]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}}, \rho_2)\right) = p_{\mathsf{l},\mathsf{l}} + \mathbf{Pr}\left(\rho_1, \rho_2 \,:\, \neg[\![b]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}}\right) \tag{2.5}$$

$$\mathbf{Pr}\left(\rho_1, \rho_2 \,:\, \mathcal{B}_{\mathsf{t}}(1^\eta, [\![\vec{w}', b', (u'_i)_i]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}}, \rho_2)\right) = p_{\mathsf{l},\mathsf{r}} + \mathbf{Pr}\left(\rho_1, \rho_2 \,:\, \neg[\![b']\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}}\right) \tag{2.6}$$

Since we assumed $\vec{w}, b, (u_i)_i \sim \vec{w}', b', (u'_i)_i$ to hold in any computational model, we know that $b \sim b'$ also holds in any computational model. Since b and b' are of sort bool, and since $b \sim b'$ holds in $\mathcal{M}_{\mathsf{c}}$, we know that the two quantities below:

$$\mathbf{Pr}\left(\rho_1, \rho_2 \,:\, [\![b]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}}\right) \qquad\qquad \mathbf{Pr}\left(\rho_1, \rho_2 \,:\, [\![b']\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}}\right)$$

are equal except for a negligible number of samplings (otherwise, we could easily break $b \sim b'$ in $\mathcal{M}_{\mathsf{c}}$). Using this fact, (2.5) and (2.6) we deduce that $\mathcal{B}_{\mathsf{t}}$ advantage against $\vec{w}, b, (u_i)_i \sim \vec{w}', b', (u'_i)_i$ is $|p_{\mathsf{l},\mathsf{l}} - p_{\mathsf{l},\mathsf{r}}|$ (up-to a negligible quantity).
We define the adversary $\mathcal{B}_{\mathsf{e}}$ against $\vec{w}, b, (v_i)_i \sim \vec{w}', b', (v'_i)_i$:

$$\mathcal{B}_{\mathsf{t}}(\vec{x}, a, (m_i)_i) = \begin{cases} \mathcal{A}(\vec{x}, (m_i)_i) & \text{if } a = 0 \\ 1 & \text{otherwise} \end{cases}$$

Similarly, we can check that $\mathcal{B}_{\mathsf{e}}$ has an advantage $|p_{\mathsf{r},\mathsf{l}} - p_{\mathsf{r},\mathsf{r}}|$ (up-to a negligible quantity).
To conclude, we observe that the advantage of $\mathcal{A}$ against $\vec{s} \sim \vec{t}$ is $|p_{\mathsf{l},\mathsf{l}} + p_{\mathsf{r},\mathsf{l}} - p_{\mathsf{l},\mathsf{r}} - p_{\mathsf{r},\mathsf{r}}|$, which is upper-bounded by $|p_{\mathsf{l},\mathsf{l}} - p_{\mathsf{l},\mathsf{r}}| + |p_{\mathsf{r},\mathsf{l}} - p_{\mathsf{r},\mathsf{r}}|$. It follows that $\mathcal{B}_{\mathsf{t}}$ or $\mathcal{B}_{\mathsf{e}}$ has a non-negligible advantage of winning against their respective formulas. ■

2.5.2 Implementation Axioms

Implementation axioms are axioms that are not valid in all computational models. When studying the security of a protocol, implementation axioms are what allow the prover to put *requirements* on the protocol concrete implementations. For example, we can require that the first projection of a pair is equal to the first element of the pair, or that distinct constant function symbols representing agents names are never equal. Then, if we can show that the conjunction of the structural axioms, the implementation axioms and the negation of the security property hold, we know that the protocol is secure in any computational model where the implementation axioms hold.

Of course, we use different implementation axioms for different protocols. Still, we give some examples of frequent axioms in the section.

Pair and Asymmetric Encryption As a first example, we consider pairs $\langle\,,\,\rangle_{/2}$, projections $\pi_{1/1}$, $\pi_{2/1}$, public key $\mathsf{pk}_{/1}$, private key $\mathsf{sk}_{/1}$ and asymmetric encryption and decryption $\{\}_{/3}$ and $\mathsf{dec}_{/2}$. Encryptions are of the form $\{u\}^{\mathsf{n_e}}_{\mathsf{pk(n)}}$ where u is the plain-text, $\mathsf{pk(n)}$ is the public key (where $\mathsf{n} \in \mathcal{N}$ is the random seed used during the key generation), and $\mathsf{n_e} \in \mathcal{N}$ is the explicit encryption randomness.

We then have the axioms $\mathsf{Ax}_{\langle\,,\rangle}$ for pairs and $\mathsf{Ax}_{\mathsf{dec}}$ for encryptions:

$$\mathsf{Ax}_{\langle\,,\rangle} \,:\, \pi_i(\langle x_1\,,\,x_2\rangle) = x_i \quad (\text{where } i \in \{1,2\}) \qquad\qquad \mathsf{Ax}_{\mathsf{dec}} \,:\, \mathsf{dec}(\{x\}^z_{\mathsf{pk}(y)}, \mathsf{sk}(y)) = x$$

Xor Axiomatization Assume that $\oplus_{/2}, 0_{/0} \in \mathcal{F}$. We want $\oplus$ and 0 to have some of the properties of, respectively, bit-wise xor and the bit-string containing η zeros. First, we have the usual ACUN (associativity, commutativity, unit and nilpotence) axioms:

$$x \oplus (y \oplus z) = (x \oplus y) \oplus z \qquad x \oplus y = y \oplus x \qquad 0 \oplus x = x \qquad x \oplus x = 0$$

While the actual bit-wise xor[2] satisfy the axioms above, these axioms are valid in other computational models. For example, we can interpret the $\oplus$ function symbol as the function that, on every inputs, return η zeroes. Or, more plausibly, we can interpret $\oplus$ has the function that only return bit-strings of length exactly η (padding or truncating its inputs if necessary).

To study protocols relying on the xor (e.g. KCL, LAK or AKA), we need the following axiom:

$$\frac{\vec{u}, \mathsf{n} \sim \vec{v} \qquad \mathsf{len}(t) = \mathsf{len}(\mathsf{n})}{\vec{u}, t \oplus \mathsf{n} \sim \vec{v}} \oplus\text{-ind} \quad \text{when } \mathsf{n} \notin \mathsf{st}(\vec{u}, \vec{v}, t) \tag{2.7}$$

Basically, this axioms states that the xor of a term t and an uniform random value n is indistinguishable from an uniform random value, as long as t and n are independent and of the same length. The fact that t and n are independent is checked by requiring that n does not appear in t in the syntactic side-condition $\mathsf{n} \notin \mathsf{st}(\vec{u}, \vec{v}, t)$ (therefore this is an infinite schema of ground axioms). We define the set of axioms $\mathsf{Ax}_{\oplus}$:

Definition 2.11. $\mathsf{Ax}_{\oplus}$ is the conjunction of the ACUN axioms and the axiom schema $\oplus$-ind in (2.7).

Booleans It is often convenient, or necessary, to add functions symbols for boolean operations:

$$\mathsf{and}, \mathsf{or}, \mathsf{imply}, \mathsf{equiv} \,:\, \mathsf{bool}^2 \to \mathsf{bool} \qquad\qquad \mathsf{neg} \,:\, \mathsf{bool} \to \mathsf{bool}$$

We can also add an axiom linking the if_then_else_ function symbol with the boolean function symbols whenever the then and else branches are of sort bool:

$$\mathsf{if}\ a\ \mathsf{then}\ b\ \mathsf{else}\ c = \mathsf{or}\,(\mathsf{and}(a, b), \mathsf{and}(\mathsf{neg}(a), c)) \tag{2.8}$$

where a, b and c are variables of sort bool. Instead of adding multiple axioms allowing to reason on terms with boolean function symbols, we use a single axiom schema stating that if two terms, seen as formulas in first-order logic with equality, are equivalent, then they are equal:

$$\frac{t_\phi \text{ and } t_\psi \text{ are encoding of } \phi \text{ and } \psi \qquad \phi \Leftrightarrow \psi \text{ valid in FO}(=)}{t_\phi = t_\psi} \tag{2.9}$$

Again, this is a recursive schema of ground axioms: t_ϕ and t_ψ are ground terms.

Example 2.8. For example, we can obtain De Morgan's laws:

$$\mathsf{neg}(\mathsf{and}(a, b)) = \mathsf{or}(\mathsf{neg}(a), \mathsf{neg}(b)) \qquad\qquad \mathsf{neg}(\mathsf{or}(a, b)) = \mathsf{and}(\mathsf{neg}(a), \mathsf{neg}(b))$$

If we consider first-order logic with equality and injectivity of the pair function symbols, we obtain:

$$\mathsf{imply}\,(\mathsf{neg}(\mathsf{eq}(\langle u\,,\, v\rangle\,, \langle s\,,\, t\rangle)), \mathsf{or}(\mathsf{neg}(\mathsf{eq}(u, s)), \mathsf{neg}(\mathsf{eq}(v, t)))) = \mathsf{true}$$ □

This allow to push part of the reasoning outside the Bana-Comon logic, into some standard logic, without having to fix the way we reason in the outer logic: the proof that ϕ and ψ are equivalent takes place in the meta-logic. In practice, we use a logic with more axioms than FO(=). For example, in the study of the AKA protocol, we will need to do reasoning about conjunctions of inequalities between integer sequence numbers, for which we need, e.g., properties of orderings. We define the set of axioms $\mathsf{Ax}_{\mathsf{bool}}$:

Definition 2.12. $\mathsf{Ax}_{\mathsf{bool}}$ is the conjunction of the axioms in (2.8) and (2.9).

Notations The prefix notation for boolean terms is cumbersome to use. Therefore, we introduce infix notations for and, or, imply, equiv, neg, eq:

$$\dot{\wedge},\ \dot{\vee},\ \dot{\to},\ \dot{\leftrightarrow},\ \dot{\neg},\ \doteq$$

We use the usual precedence, e.g. $a \,\dot{\vee}\, b \,\dot{\wedge}\, c$ is $a \,\dot{\vee}\, (b \,\dot{\wedge}\, c)$. For every boolean term b, when there is no confusion, we write b instead of $b \sim \mathsf{true}$.

[2] When xoring bit-strings of different lengths, the shorter bit-string is padded with 0s to the length of the longer one.

While it may seems that we need to be careful not to confuse $\doteq$ and $=$, this is actually not the case. Indeed, the formula $a \doteq b$ is, by definition, the formula $\mathsf{eq}(a,b) \sim \mathsf{true}$, which is also the formula $a = b$. Moreover, the following two rules are admissible using the axioms in Figure 2.1 and 2.2:

$$\frac{a \doteq b}{(a \doteq b) \doteq \mathsf{true}} \qquad \frac{(a \doteq b) \doteq \mathsf{true}}{a \doteq b}$$

We give the derivations below:

$$\dfrac{\dfrac{\dfrac{\dfrac{a \doteq b}{a \doteq b \sim \mathsf{true}}}{(a \doteq b), \mathsf{true} \sim \mathsf{true}, \mathsf{true}}\,\mathsf{FA}}{(a \doteq b) \doteq \mathsf{true} \sim \mathsf{true} \doteq \mathsf{true}}\,\mathsf{FA} \qquad \dfrac{}{(\mathsf{true} \doteq \mathsf{true}) = \mathsf{true}}\,\text{=-refl}}{\dfrac{(a \doteq b) \doteq \mathsf{true} \sim \mathsf{true}}{(a \doteq b) \doteq \mathsf{true}}}\,\mathsf{Equ} \qquad \dfrac{\dfrac{}{\mathsf{true} \sim \mathsf{true}}\,\mathsf{Refl} \qquad (a \doteq b) \doteq \mathsf{true}}{\dfrac{a \doteq b \sim \mathsf{true}}{a \doteq b}}\,\mathsf{Equ}$$

Ax-Interpretation Instead of proving that a protocol is secure in any computational model, we are going to prove that it is secure in any computational model satisfying some implementation assumptions (such as $\mathsf{Ax}_{\mathsf{dec}}, \mathsf{Ax}_{\oplus}$ ...). To make the distinction between implementation axioms, which only restrict the interpretations of the function in $\mathcal{F}_{\mathsf{p}}$, and other axioms, which restrict all function symbols interpretations, including function symbols in $\mathcal{G}$, we introduce the notion of Ax-interpretation.

Definition 2.13. For every set of axioms Ax, an Ax-interpretation $\mathcal{I}_{\mathsf{c}}$ for $\mathcal{F}_{\mathsf{p}}$ is a computational model $\mathcal{I}_{\mathsf{c}}$ on signature $\mathcal{F}_{\mathsf{p}}$ such that $\mathcal{I}_{\mathsf{c}} \models \mathsf{Ax}$.

Given an adversary $\mathcal{A}$ and an Ax-interpretation $\mathcal{I}_{\mathsf{c}}$ on $\mathcal{F}_{\mathsf{p}}$, we can lift $\mathcal{I}_{\mathsf{c}}$ to a full computational model on signature $(\mathcal{F}_{\mathsf{p}}, \mathcal{G})$ by letting $\mathcal{A}$ interpret all functions in $\mathcal{G}$.

2.6 Cryptographic Assumptions and Axioms

We now explain how we translated several cryptographic assumptions into axioms. Before starting, we introduce some notations used to define side-conditions of cryptographic axioms.

Definition 2.14. For every ground terms $\vec{u}, \vec{v}$, we let $\mathsf{fresh}(\vec{u}; \vec{v})$ hold if and only if no term in $\vec{u}$ is a subterm of a term in $\vec{v}$, i.e.:

$$\{u \mid u \in \vec{u}\} \cap \mathsf{st}(\vec{v}) = \emptyset$$

Definition 2.15. Let $s, \vec{u}$ be ground terms and $C_{\vec{x},\cdot}$ be a context with one distinguished hole variable $\cdot$ such that the hole variable $\cdot$ appears exactly once in $C_{\vec{x},\cdot}$. We let $s \sqsubseteq_{C_{\vec{x},\cdot}} \vec{u}$ holds whenever s appears in $\vec{u}$ only in subterms of the form $C[\vec{w}, s]$. Formally:

$$\forall u \in \vec{u}, \forall p \in \mathsf{pos}(u), u_{|p} \equiv s \rightarrow \exists \vec{w} \in \mathcal{T}(\mathcal{F}, \mathcal{N}), \exists q \in \mathsf{pos}(u) \text{ s.t. } q \leq p \wedge u_{|q} \equiv C[\vec{w}, s]$$

We generalize this to n contexts $C_1, \ldots, C_n$, by allowing s to appear only as subterm of one of the C_is. Formally, we let $s \sqsubseteq_{C_1,\ldots,C_n} \vec{u}$ if and only if:

$$\forall u \in \vec{u}, \forall p \in \mathsf{pos}(u), u_{|p} \equiv s \rightarrow \exists \vec{w} \in \mathcal{T}(\mathcal{F}, \mathcal{N}), \exists q \in \mathsf{pos}(u), \exists 1 \leq i \leq n \text{ s.t. } q \leq p \wedge u_{|q} \equiv C_i[\vec{w}, s]$$

Example 2.9. Two examples:

- $\mathsf{n} \sqsubseteq_{\mathsf{pk}(\cdot),\mathsf{sk}(\cdot)} \vec{u}$ states that the nonce n appears only in terms of the form $\mathsf{pk}(\mathsf{n})$ or $\mathsf{sk}(\mathsf{n})$ in $\vec{u}$.
- $\mathsf{sk}(\mathsf{n}) \sqsubseteq_{\mathsf{dec}(\_,\cdot)} \vec{u}$ states that the secret key $\mathsf{sk}(\mathsf{n})$ appears only in decryption position in $\vec{u}$. □

2.6.1 The CCA$_1$ Axioms

In the computational model, the security of a cryptographic primitive is expressed through a game between a challenger and an attacker (which is a PPTM) that tries to break the primitive.

We informally recall the IND-CCA$_2$ game (for Indistinguishability against Chosen Ciphertexts Attacks, see [BDPR98]). First, the challenger computes a public/private key pair $(\mathsf{pk}(\mathsf{n}), \mathsf{sk}(\mathsf{n}))$ (using a nonce n of length η uniformly sampled), and sends $\mathsf{pk}(\mathsf{n})$ to the attacker. The adversary has access to two oracles:

- A left-right oracle $\mathcal{O}^b_{\mathsf{LR}}(\mathsf{n})$ that takes two messages m_0, m_1 of the same length as input and returns $\{m_b\}^{\mathsf{n}_r}_{\mathsf{pk}(\mathsf{n})}$, where b is an internal random bit uniformly drawn at the beginning by the challenger and n_r is a fresh nonce.
- A decryption oracle $\mathcal{O}_{\mathsf{dec}}(\mathsf{n})$ that, given m, returns $\mathsf{dec}(m, \mathsf{sk}(\mathsf{n}))$ if m was not submitted to the $\mathcal{O}_{\mathsf{LR}}$ oracle yet, and length of m zeros otherwise.

Remark that the two oracles have a shared memory. The advantage $\mathrm{Adv}^{\mathrm{CCA2}}_{\mathcal{A}}(\eta)$ of an adversary $\mathcal{A}$ against this game is the probability for $\mathcal{A}$ to guess the bit b:

$$\left|\ \mathbf{Pr}\big(\mathsf{n} : \mathcal{A}^{\mathcal{O}^1_{\mathsf{LR}}(\mathsf{n}),\mathcal{O}_{\mathsf{dec}}(\mathsf{n})}(1^\eta) = 1\big) - \mathbf{Pr}\big(\mathsf{n} : \mathcal{A}^{\mathcal{O}^0_{\mathsf{LR}}(\mathsf{n}),\mathcal{O}_{\mathsf{dec}}(\mathsf{n})}(1^\eta) = 1\big)\ \right|$$

An encryption scheme is IND-CCA$_2$ if the advantage $\mathrm{Adv}^{\mathrm{CCA2}}_{\mathcal{A}}(\eta)$ of any adversary $\mathcal{A}$ is negligible in η. The IND-CCA$_1$ game is the restriction of this game where the adversary cannot call $\mathcal{O}_{\mathsf{dec}}$ after having called $\mathcal{O}_{\mathsf{LR}}$. An encryption scheme is IND-CCA$_1$ if $\mathrm{Adv}^{\mathsf{CCA1}}_{\mathcal{A}}(\eta)$ is negligible for any adversary $\mathcal{A}$.

In this section, we only present axioms for the simpler IND-CCA$_1$ game. We will present axioms for the IND-CCA$_2$ game later, in Chapter 5, Section 5.4.

The CCA$_1^s$ Axioms We define first a set of axioms CCA$_1^s$:

Definition 2.16. We let CCA$_1^s$ be the set of axioms:

$$\frac{\mathsf{len}(s) = \mathsf{len}(t)}{\vec{u}, \{s\}^{\mathsf{n}_\mathsf{e}}_{\mathsf{pk}(\mathsf{n})} \sim \vec{u}, \{t\}^{\mathsf{n}_\mathsf{e}}_{\mathsf{pk}(\mathsf{n})}}\ \mathrm{CCA}_1^s \qquad \text{when } \begin{cases} \mathsf{fresh}(\mathsf{n}_\mathsf{e}; \vec{u}, s, t) \\ \mathsf{n} \sqsubseteq_{\mathsf{pk}(\cdot),\mathsf{sk}(\cdot)} \vec{u}, s, t \ \wedge\ \mathsf{sk}(\mathsf{n}) \sqsubseteq_{\mathsf{dec}(\_,\cdot)} \vec{u}, s, t \end{cases}$$

This set of axioms CCA$_1^s$ is very similar to the one used in [BCL14]. The only difference is that in [BCL14], the length equality requirement is not a premise of the axiom. Instead, if the length are not equal they return a error message. We found our version of the axiom simpler to use.[3]

We have the following soundness property:

Proposition 2.5. *The* CCA$_1^s$ *axioms are valid in any computational model where* $(\{\}, \mathsf{dec}, \mathsf{pk}, \mathsf{sk})$ *is interpreted as an* IND-CCA$_1$ *secure encryption scheme.*

Proof. The proof is by contradiction, and is given below.

We assume that there is a computational model $\mathcal{M}_\mathsf{c}$ where the encryption scheme is IND-CCA$_1$ secure, and such that there is an instance $\vec{u}, \{s\}^{\mathsf{n}_\mathsf{e}}_{\mathsf{pk}(\mathsf{n})} \sim \vec{v}, \{t\}^{\mathsf{n}_\mathsf{e}}_{\mathsf{pk}(\mathsf{n})}$ of the axioms CCA$_1^s$ which is not valid. We deduce that there exists an attacker $\mathcal{A}$ that can distinguish between the left and right terms, i.e. the following quantity is non-negligible:

$$\left|\mathbf{Pr}\big(\rho_1, \rho_2 : \mathcal{A}(1^\eta, [\![\vec{u}, \{s\}^{\mathsf{n}_\mathsf{e}}_{\mathsf{pk}(\mathsf{n})}]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}, \rho_2) = 1\big) - \mathbf{Pr}\big(\rho_1, \rho_2 : \mathcal{A}(1^\eta, [\![\vec{u}, \{t\}^{\mathsf{n}_\mathsf{e}}_{\mathsf{pk}(\mathsf{n})}]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}, \rho_2) = 1\big)\right| \quad (2.10)$$

Using $\mathcal{A}$, we can build an adversary $\mathcal{B}$ with a non-negligible advantage against the IND-CCA1 game. First, $\mathcal{B}$ samples a vector of bit-strings $\vec{u}_s, s_s, t_s$ from $[\![\vec{u}, s, t]\!]_{\mathcal{M}_\mathsf{c}}$, querying the decryption oracle whenever it needs to compute a subterm of the from $\mathsf{dec}(\_, \mathsf{sk}(\mathsf{n}))$. Remark that the syntactic side-conditions:

$$\mathsf{n} \sqsubseteq_{\mathsf{pk}(\cdot),\mathsf{sk}(\cdot)} \vec{u}, s, t \qquad\qquad \mathsf{sk}(\mathsf{n}) \sqsubseteq_{\mathsf{dec}(\_,\cdot)} \vec{u}, s, t$$

guarantee that this is always possible. Afterward, $\mathcal{B}$ queries the left-or-right oracle with (s_s, t_s) to get a value a. Here, we need the side-condition $\mathsf{fresh}(\mathsf{n}_\mathsf{e}; \vec{u}, s, t)$ to guarantee that the random value n_e has not been sampled by $\mathcal{B}$. Indeed, the value n_e is sampled by the challenger, and is not available to $\mathcal{B}$. If the challenger internal bit b is 0 then $\vec{u}_s, a$ has been sampled from $[\![\vec{u}, \{s\}^{\mathsf{n}_\mathsf{e}}_{\mathsf{pk}(\mathsf{n})}]\!]_{\mathcal{M}_\mathsf{c}}$, and if the challenger internal bit is 1 then $\vec{u}_s, a$ has been sampled from $[\![\vec{u}, \{t\}^{\mathsf{n}_\mathsf{e}}_{\mathsf{pk}(\mathsf{n})}]\!]_{\mathcal{M}_\mathsf{c}}$:

$$\vec{u}_s, a \xleftarrow{\$} \begin{cases} [\![\vec{u}, \{s\}^{\mathsf{n}_\mathsf{e}}_{\mathsf{pk}(\mathsf{n})}]\!]_{\mathcal{M}_\mathsf{c}} & \text{if } b = 0 \\ [\![\vec{u}, \{t\}^{\mathsf{n}_\mathsf{e}}_{\mathsf{pk}(\mathsf{n})}]\!]_{\mathcal{M}_\mathsf{c}} & \text{if } b = 1 \end{cases}$$

Finally, $\mathcal{B}$ returns $\mathcal{A}(\vec{u}_s, a)$. It is easy to check that the advantage of $\mathcal{B}$ against the IND-CCA$_1$ game is exactly the advantage of $\mathcal{A}$ against $\vec{u}, \{s\}^{\mathsf{n}_\mathsf{e}}_{\mathsf{pk}(\mathsf{n})} \sim \vec{v}, \{t\}^{\mathsf{n}_\mathsf{e}}_{\mathsf{pk}(\mathsf{n})}$ This advantage is the quantity in (2.10), which we assumed non-negligible. Hence $\mathcal{B}$ is winning against the IND-CCA$_1$ game. Contradiction. ■

[3] The two formulations should be equivalent provided that you have the CS and Equ axioms, and that you can do basic reasoning on lengths (though we did not prove it).

The CCA$_1$ Axioms We define the axioms CCA$_1$, which are more convenient to use than CCA$_1^s$. Basically, CCA$_1$ is the axiom CCA$_1^s$ where we applied transitivity to have different terms $\vec{u}$, $\vec{v}$ on each side.

Definition 2.17. We let CCA$_1$ be the set of axioms:

$$\frac{\vec{u}, \mathsf{len}(s) \sim \vec{v}, \mathsf{len}(t)}{\vec{u}, \{s\}^{\mathsf{n_e}}_{\mathsf{pk(n)}} \sim \vec{v}, \{t\}^{\mathsf{n'_e}}_{\mathsf{pk(n')}}}\ \text{CCA}_1 \qquad \text{when} \begin{cases} \mathsf{fresh}(\mathsf{n_e}, \mathsf{n'_e}; \vec{u}, \vec{v}, s, t) \\ \vec{u} \equiv \mathsf{pk(n)}, \_ \ \wedge\ \vec{v} \equiv \mathsf{pk(n')}, \_ \\ \mathsf{n} \sqsubseteq_{\mathsf{pk}(\cdot),\mathsf{sk}(\cdot)} \vec{u}, s \ \wedge\ \mathsf{sk(n)} \sqsubseteq_{\mathsf{dec}(\_,\cdot)} \vec{u}, s \\ \mathsf{n'} \sqsubseteq_{\mathsf{pk}(\cdot),\mathsf{sk}(\cdot)} \vec{v}, t \ \wedge\ \mathsf{sk(n')} \sqsubseteq_{\mathsf{dec}(\_,\cdot)} \vec{v}, t \end{cases}$$

We have the following soundness theorem:

Proposition 2.6. *The* CCA$_1$ *axioms are valid in any computational model where* $(\{\}, \mathsf{dec}, \mathsf{pk}, \mathsf{sk})$ *is interpreted as an* IND-CCA$_1$ *secure encryption scheme.*

Proof. We are going to give a direct derivation of the axioms CCA$_1$, using rules that are valid in all computational models where $(\{\}, \mathsf{dec}, \mathsf{pk}, \mathsf{sk})$ is interpreted as an IND-CCA$_1$ secure encryption scheme. The derivation mostly rely on the Trans and the CCA$_1^s$ axioms. First, we use transitivity to split the goal $\vec{u}, \{s\}^{\mathsf{n_e}}_{\mathsf{pk(n)}} \sim \vec{v}, \{t\}^{\mathsf{n'_e}}_{\mathsf{pk(n')}}$ into three sub-goals, by replacing the plain-texts with zeros:

$$\overbrace{\vec{u}, \{s\}^{\mathsf{n_e}}_{\mathsf{pk(n)}} \sim \underbrace{\vec{u}, \{\mathbf{0}(\mathsf{len}(s))\}^{\mathsf{n_e}}_{\mathsf{pk(n)}} \sim \vec{v}, \{\mathbf{0}(\mathsf{len}(t))\}^{\mathsf{n'_e}}_{\mathsf{pk(n')}}}}^{} \sim \vec{v}, \{t\}^{\mathsf{n'_e}}_{\mathsf{pk(n')}}$$

We deal with the left and right sub-goals using the CCA$_1^s$ axioms. We deal with the length equality constraint of the CCA$_1^s$ axioms using the axioms:

$$\mathsf{len}(t) = \mathsf{len}(\mathbf{0}(\mathsf{len}(t))) \qquad\qquad \mathsf{len}(s) = \mathsf{len}(\mathbf{0}(\mathsf{len}(s)))$$

which are valid in any computational model, using the fact that len interpretation is fixed. Finally, for the middle sub-goal, we deconstruct the terms using the FA rule and then apply Dup and Fresh. Putting everything together:

$$\dfrac{\dfrac{\overline{\mathsf{len}(s) = \mathsf{len}(\mathbf{0}(\mathsf{len}(s)))}}{\vec{u}, \{s\}^{\mathsf{n_e}}_{\mathsf{pk(n)}} \sim \vec{u}, \{\mathbf{0}(\mathsf{len}(s))\}^{\mathsf{n_e}}_{\mathsf{pk(n)}}}\ \text{CCA}_1^s \qquad \boxed{\vec{u}, \{\mathbf{0}(\mathsf{len}(s))\}^{\mathsf{n_e}}_{\mathsf{pk(n)}} \sim \vec{v}, \{t\}^{\mathsf{n'_e}}_{\mathsf{pk(n')}}}}{\vec{u}, \{s\}^{\mathsf{n_e}}_{\mathsf{pk(n)}} \sim \vec{v}, \{t\}^{\mathsf{n'_e}}_{\mathsf{pk(n')}}}\ \text{Trans}$$

$$\dfrac{\boxed{\vec{u}, \{\mathbf{0}(\mathsf{len}(s))\}^{\mathsf{n_e}}_{\mathsf{pk(n)}} \sim \vec{v}, \{\mathbf{0}(\mathsf{len}(t))\}^{\mathsf{n'_e}}_{\mathsf{pk(n')}}} \qquad \dfrac{\overline{\mathsf{len}(t) = \mathsf{len}(\mathbf{0}(\mathsf{len}(t)))}}{\vec{v}, \{\mathbf{0}(\mathsf{len}(t))\}^{\mathsf{n'_e}}_{\mathsf{pk(n')}} \sim \vec{v}, \{t\}^{\mathsf{n'_e}}_{\mathsf{pk(n')}}}\ \text{CCA}_1^s}{\boxed{\vec{u}, \{\mathbf{0}(\mathsf{len}(s))\}^{\mathsf{n_e}}_{\mathsf{pk(n)}} \sim \vec{v}, \{t\}^{\mathsf{n'_e}}_{\mathsf{pk(n')}}}}\ \text{Trans}$$

$$\dfrac{\dfrac{\dfrac{\overline{\vec{u}, \mathsf{len}(s) \sim \vec{v}, \mathsf{len}(t)}}{\vec{u}, \mathsf{len}(s), \mathsf{n_e} \sim \vec{v}, \mathsf{len}(t), \mathsf{n'_e}}\ \text{Fresh}}{\vec{u}, \mathsf{len}(s), \mathsf{pk(n)}, \mathsf{n_e} \sim \vec{v}, \mathsf{len}(t), \mathsf{pk(n')}, \mathsf{n'_e}}\ \text{Dup}}{\boxed{\vec{u}, \{\mathbf{0}(\mathsf{len}(s))\}^{\mathsf{n_e}}_{\mathsf{pk(n)}} \sim \vec{v}, \{\mathbf{0}(\mathsf{len}(t))\}^{\mathsf{n'_e}}_{\mathsf{pk(n')}}}}\ \text{FA}^3$$

■

2.6.2 The CR-HK Axioms

We now give the axioms we designed for keyed-hash function satisfying the Collision Resistance assumption. The idea is that, if a hash function $\mathsf{H}(\cdot, \mathsf{k})$ is collision-resistant, then no polynomial-time adversary can find distinct messages having the same image by $\mathsf{H}(\cdot, \mathsf{k})$. Formally:

Definition 2.18 (CR-HK [GB01])**.** A hash function H is said to be *collision resistant under hidden-key attacks* iff for any PPTM $\mathcal{A}$ with oracle access to the keyed hash function, the following quantity:

$$\mathbf{Pr}\left(\mathsf{k} : \mathcal{A}^{\mathcal{O}_{\mathsf{H}(\cdot,\mathsf{k})}}(1^\eta) = \langle m_1\,,\, m_2\rangle, m_1 \neq m_2 \text{ and } \mathsf{H}(m_1,\mathsf{k}) = \mathsf{H}(m_2,\mathsf{k})\right)$$

is negligible, where k is drawn uniformly at random in $\{0,1\}^\eta$.

We translate this game in the logic as follows:

Definition 2.19. We let CR be the set of axioms:

$$\frac{}{\mathsf{H}(m_1,\mathsf{k}) \doteq \mathsf{H}(m_2,\mathsf{k}) \;\dot{\rightarrow}\; m_1 \doteq m_2}\ \text{CR} \qquad \text{when } \mathsf{k} \sqsubseteq_{\mathsf{H}(\_,\cdot)} m_1, m_2$$

Remark 2.3. We need the implication here, we cannot simply state that, when the terms m_1 and m_2 are distinct, we have:

$$(\mathsf{H}(m_1,\mathsf{k}) \doteq \mathsf{H}(m_2,\mathsf{k})) = \mathsf{false} \tag{2.11}$$

For instance, take $m_1 = g(u)$ and $m_2 = g(u')$ where u, u' are distinct and g is an attacker's function symbol. Then, even though m_1 and m_2 are syntactically distinct, the function symbol g can be interpreted, e.g., as a function that discards its argument and always returns the same value. In such a case, the computational interpretations of m_1 and m_2 are identical, and the formula in (2.11) is not valid. □

Proposition 2.7. *The* CR *axioms are valid in any computational model where the function symbol H is interpreted as a* CR-HK *keyed hash function.*

Proof. Let b be the following boolean term:

$$\mathsf{H}(m_1,\mathsf{k}) \doteq \mathsf{H}(m_2,\mathsf{k}) \;\dot{\rightarrow}\; m_1 \doteq m_2$$

Let $\mathcal{M}_\mathsf{c}$ be a computational model such that H is interpreted by as collision-resistant keyed hash function, and assume that there exists an adversary $\mathcal{A}$ such that:

$$\left|\mathbf{Pr}\left(\rho_1,\rho_2 : \mathcal{A}(1^\eta, [\![b]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}, \rho_2)\right) - \mathbf{Pr}\left(\rho_1,\rho_2 : \mathcal{A}(1^\eta, [\![\mathsf{true}]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}, \rho_2)\right)\right|$$

is non-negligible. Since b is a boolean term, $[\![b]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}} \in \{0,1\}$, hence the existence of $\mathcal{A}$ is equivalent to:

$$\mathbf{Pr}\left(\rho_1,\rho_2 : [\![b]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}} = 0\right) \text{ is non-negligible} \tag{2.12}$$

We are going to define an adversary $\mathcal{B}$ against the CR-HK game. Since the only occurrences of k in m_1 and m_2 are as second argument of H, the adversary $\mathcal{B}$ can sample two value a_1 and b_2 from, respectively, $[\![m_1]\!]_{\mathcal{M}_\mathsf{c}}$ and $[\![m_2]\!]_{\mathcal{M}_\mathsf{c}}$ (names different from k are uniformly sampled by $\mathcal{B}$, and subterms of the form $\mathsf{H}(u,\mathsf{k})$ are computed by calling the hash oracle). The adversary $\mathcal{B}$ returns $\langle a_1\,,\, a_2\rangle$. Then:

$$\begin{aligned}
&\mathbf{Pr}\left(\mathsf{k} : \mathcal{B}^{\mathcal{O}_{\mathsf{H}(\cdot,\mathsf{k})}}(1^\eta) = \langle x_1\,,\, x_2\rangle, x_1 \neq x_2 \text{ and } \mathsf{H}(x_1,\mathsf{k}) = \mathsf{H}(x_2,\mathsf{k})\right)\\
=\ &\mathbf{Pr}\left(\rho_1,\rho_2 : [\![m_1]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}} \neq [\![m_2]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}} \wedge [\![\mathsf{H}(m_1,\mathsf{k})]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}} = [\![\mathsf{H}(m_2,\mathsf{k})]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}\right)\\
=\ &\mathbf{Pr}\left(\rho_1,\rho_2 : \neg\left([\![\mathsf{H}(m_1,\mathsf{k})]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}} = [\![\mathsf{H}(m_2,\mathsf{k})]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}} \rightarrow [\![m_1]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}} = [\![m_2]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}\right)\right)\\
=\ &\mathbf{Pr}\left(\rho_1,\rho_2 : \neg[\![\mathsf{H}(m_1,\mathsf{k}) \doteq \mathsf{H}(m_2,\mathsf{k}) \dot{\rightarrow} m_1 \doteq m_2]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}}\right)\\
=\ &\mathbf{Pr}\left(\rho_1,\rho_2 : [\![\mathsf{H}(m_1,\mathsf{k}) \doteq \mathsf{H}(m_2,\mathsf{k}) \dot{\rightarrow} m_1 \doteq m_2]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_\mathsf{c}} = 0\right)
\end{aligned}$$

which by hypothesis (2.12) is non-negligible. ■

2.6.3 EUF-MAC Axioms

A Mac schema is a pair $(\mathsf{Mac}\_(\_), \mathsf{Verify}(\_,\_,\_))$ where Mac creates symmetric signatures of messages, and Verify checks that some message has a valid signature. For every η, they must satisfy the following soundness relation:

$$\forall k \in \{0,1\}^\eta, \forall m \in \{0,1\}^*.\ \mathsf{Verify}(m, \mathsf{Mac}_\mathsf{k}(m), \mathsf{k}) = \mathsf{true}$$

Moreover, $\mathsf{Mac}_\mathsf{k}(\cdot)$ must be computationally unforgeable, even when letting the adversary have access to a Mac oracle $\mathcal{O}_{\mathsf{Mac}_\mathsf{k}(\cdot)}$. To successively forge a Mac, the adversary must find a pair (m,σ) such that $\mathsf{Verify}(m,\sigma,\mathsf{k})$ and m was never queried to the oracle $\mathcal{O}_{\mathsf{Mac}_\mathsf{k}(\cdot)}$. Formally:

Definition 2.20. A Mac schema $(\mathsf{Mac}, \mathsf{Verify})$ is *unforgeable against chosen-message attacks* (EUF-CMA) iff for every PPTM $\mathcal{A}$, the following quantity:

$$\mathbf{Pr}\left(\mathsf{k} : \mathcal{A}^{\mathcal{O}_{\mathsf{Mac}_{\mathsf{k}}(\cdot)}}(1^\eta) = \langle m\,,\,\sigma\rangle\,, m \text{ not queried to } \mathcal{O}_{\mathsf{Mac}_{\mathsf{k}}(\cdot)} \text{ and } \mathsf{Verify}(m,\sigma,\mathsf{k})\right)$$

is negligible, where k is drawn in $\{0,1\}^\eta$.

We explain how we translate this cryptographic assumption in the logic. Given two terms m, s where m is a message and s is a (candidate) forgery of a Mac of m, if s is a valid forgery (i.e. $\mathsf{Verify}(m,s,\mathsf{k})$ holds) then s must be a honestly generated Mac. Moreover, the set of honest Macs is simply the set of all subterms of m and s which are of the form $\mathsf{Mac}_{\mathsf{k}}(\_)$. This motivates the following definition:

Definition 2.21. We let $\mathsf{set\text{-}mac}_{\mathsf{k}}(u)$ be the set of Maced terms, using key k, in u:

$$\mathsf{set\text{-}mac}_{\mathsf{k}}(u) = \{m \mid \mathsf{Mac}_{\mathsf{k}}(m) \in \mathsf{st}(u)\}$$

We can now give the EUF-MAC axioms:

Definition 2.22. We let EUF-MAC be the set of axioms:

$$\frac{}{\mathsf{Verify}(m,s,\mathsf{k}) \;\dot{\rightarrow}\; \dot{\bigvee}_{u\in S}\, s \doteq \mathsf{Mac}_{\mathsf{k}}(u)}\ \text{EUF-MAC} \qquad \text{when } \begin{cases} \mathsf{k} \sqsubseteq_{\mathsf{Mac}.(\_)} s, m \\ S = \mathsf{set\text{-}mac}_{\mathsf{k}}(s,m) \end{cases}$$

For these axioms to be valid, we need the Mac schema to be such that every message has exactly one valid Mac. Formally, we require that:

$$\forall k \in \{0,1\}^\eta, \forall m \in \{0,1\}^*.\ \mathsf{Verify}(m,s,\mathsf{k}) = \mathsf{true} \;\rightarrow\; s = \mathsf{Mac}_{\mathsf{k}}(m) \tag{2.13}$$

Proposition 2.8. *The* EUF-MAC *axioms are valid in any computational model where* $(\mathsf{Mac}, \mathsf{Verify})$ *is interpreted as an* EUF-CMA *secure function and satisfies* (2.13).

Proof. We assume that there is a computational model $\mathcal{M}_{\mathsf{c}}$ where $(\mathsf{Mac}, \mathsf{Verify})$ is interpreted as an EUF-CMA secure function. Moreover, we assume that there is an instance:

$$\frac{}{\mathsf{Verify}(m,s,\mathsf{k}) \;\dot{\rightarrow}\; \dot{\bigvee}_{u\in S}\, s \doteq \mathsf{Mac}_{\mathsf{k}}(u)}\ \text{EUF-MAC}$$

of the EUF-MAC axioms which is not valid in $\mathcal{M}_{\mathsf{c}}$, where $S = \mathsf{set\text{-}mac}_{\mathsf{k}}(s,m)$. Therefore we know that the following quantity is non-negligible:

$$\mathbf{Pr}\left(\rho_1,\rho_2 \,:\, [\![\mathsf{Verify}(m,s,\mathsf{k})]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}} \wedge \neg [\![\dot{\bigvee}_{u\in S}\, s \doteq \mathsf{Mac}_{\mathsf{k}}(u)]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}}\right)$$

Or, equivalently, the following quantity is non-negligible:

$$\mathbf{Pr}\left(\rho_1,\rho_2 \,:\, [\![\mathsf{Verify}(m,s,\mathsf{k})]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}} \wedge \bigwedge_{u\in S} [\![s]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}} \neq [\![\mathsf{Mac}_{\mathsf{k}}(u)]\!]^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}}\right) \tag{2.14}$$

Using $\mathcal{M}_{\mathsf{c}}$, we can build an adversary $\mathcal{A}$ against the EUF-CMA game. The adversary $\mathcal{A}$ simply samples two values a_s, a_m from $[\![s]\!]_{\mathcal{M}_{\mathsf{c}}}$ and $[\![m]\!]_{\mathcal{M}_{\mathsf{c}}}$, by sampling all the subterms of s and m in a bottom-up fashion. The adversary calls the Mac oracle $\mathcal{O}_{\mathsf{Mac}_{\mathsf{k}}(\cdot)}$ whenever he needs to sample a value from a subterm of the form $\mathsf{Mac}_{\mathsf{k}}(\_)$. Remark that the side-condition $\mathsf{k} \sqsubseteq_{\mathsf{Mac}.(\_)} s, m$ ensures that this is always possible. Then $\mathcal{A}$ returns a_s, a_m. Using the property (2.13), we know that m was never queried to the Mac oracle. Hence, the advantage of $\mathcal{A}$ against the EUF-CMA game is exactly the quantity (2.14). It follows that $\mathcal{A}$ has a non-negligible probability of winning the game. Contradiction. ■

P-EUF-MAC$_s$ Axioms We can refine the unforgeability axioms EUF-MAC using a finite partition of the outcomes, which is quite useful in proofs.

Definition 2.23. A finite family of conditionals $(b_i)_{i\in I}$ is a valid CS partition under some axioms Ax if the following formula is valid in every computational model satisfying the axioms Ax:

$$\left(\dot{\bigvee}_i\, b_i \;\dot{\wedge}\; \dot{\bigwedge}_{i\neq j}\, b_i \neq b_j\right) = \mathsf{true}$$

Definition 2.24. For every terms b, t, we let $[b]t$ be the term $\mathsf{if}\ b\ \mathsf{then}\ t\ \mathsf{else}\ \mathsf{defaut}$, where defaut is a constant function symbol of types term and bool.

We can have a more precise axiom, by considering a valid CS partition $(b_i)_{i\in I}$ and applying the EUF-MAC axiom once for each element of the partition.

Definition 2.25. We let P-EUF-MAC$_s$ be the set of axioms:

$$\frac{}{\mathsf{Verify}(m,s,\mathsf{k}) \dot{\rightarrow} \dot{\bigvee}_{i\in I} b_i \mathrel{\dot{\wedge}} \dot{\bigvee}_{u\in S_i} s \doteq \mathsf{Mac}_{\mathsf{k}}(u)} \quad \text{when} \begin{cases} \mathsf{k} \sqsubseteq_{\mathsf{Mac}.(\_)} s, m \\ (b_i)_{i\in I} \text{ is a valid } \mathsf{CS} \text{ partition} \\ \text{There exists } (s_i, m_i)_{i\in I} \text{ s.t. for every } i \in I \\ \quad [b_i]s_i = [b_i]s \wedge [b_i]m_i = [b_i]m \\ \quad S_i = \mathsf{set\text{-}mac}_{\mathsf{k}}(s_i, m_i) \end{cases}$$

Proposition 2.9. *The* P-EUF-MAC$_s$ *axioms are valid in any computational model where* $(\mathsf{Mac}, \mathsf{Verify})$ *is interpreted as an* EUF-CMA *secure function and satisfies* (2.13).

Proof. To show this, we prove that the P-EUF-MAC$_s$ axioms are a logical consequences of the axioms EUF-MAC and the axioms in Figure 2.1 and 2.2. The proof is pretty straightforward:

$$\begin{aligned}
\mathsf{Verify}(m,s,\mathsf{k}) &\rightarrow \dot{\bigvee}_{i\in I} b_i \mathrel{\dot{\wedge}} \mathsf{Verify}(m,s,\mathsf{k}) && \text{(Since } (b_i)_{i\in I} \text{ is a valid CS partition)} \\
&\dot{\rightarrow} \dot{\bigvee}_{i\in I} b_i \mathrel{\dot{\wedge}} \mathsf{Verify}(m_i,s_i,\mathsf{k}) && \text{(Since } [b_i]s_i = [b_i]s \text{ and } [b_i]m_i = [b_i]m) \\
&\dot{\rightarrow} \dot{\bigvee}_{i\in I} b_i \mathrel{\dot{\wedge}} \dot{\bigvee}_{u\in S_i} s_i \doteq \mathsf{Mac}_{\mathsf{k}}(u) && \text{(Using EUF-MAC for every } i \in I) \\
&\dot{\rightarrow} \dot{\bigvee}_{i\in I} b_i \mathrel{\dot{\wedge}} \dot{\bigvee}_{u\in S_i} s \doteq \mathsf{Mac}_{\mathsf{k}}(u) && \blacksquare
\end{aligned}$$

P-EUF-MAC Axioms We can further refine the unforgeability axioms, by noticing that Macs appearing only in boolean conditionals can be ignored. For this, we let $\mathsf{strict\text{-}st}(u)$ be the set of subterms of u appearing outside u's conditionals. The definition is by structural induction on u.

Definition 2.26. For every u, we let $\mathsf{strict\text{-}st}(u)$ be the set of subterms of u appearing outside conditionals:

$$\begin{aligned}
\mathsf{strict\text{-}st}(\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v) &= \{\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v\} \cup \mathsf{strict\text{-}st}(u) \cup \mathsf{strict\text{-}st}(v) \\
\mathsf{strict\text{-}st}(f(\vec{u})) &= \{f(\vec{u})\} \cup \textstyle\bigcup_{u\in\vec{u}} \mathsf{strict\text{-}st}(u) && (\forall f \in \mathcal{F} \backslash \{\mathsf{if_then_else_}\})
\end{aligned}$$

We define the set of strict Mac subterms of a term u:

Definition 2.27. We let $\mathsf{strict\text{-}set\text{-}mac}_{\mathsf{k}}(u)$ be the set of mac-ed terms under key k in u appearing outside a conditional:

$$\mathsf{strict\text{-}set\text{-}mac}_{\mathsf{k}}(u) = \{m \mid \mathsf{Mac}_{\mathsf{k}}(m) \in \mathsf{strict\text{-}st}(u)\}$$

We give the axioms:

Definition 2.28. We let P-EUF-MAC be the set of axioms:

$$\frac{}{\mathsf{Verify}(m,s,\mathsf{k}) \dot{\rightarrow} \dot{\bigvee}_{i\in I} b_i \mathrel{\dot{\wedge}} \bigvee_{u\in S_i} s \doteq \mathsf{Mac}_{\mathsf{k}}(u)} \quad \text{when} \begin{cases} \mathsf{k} \sqsubseteq_{\mathsf{Mac}.(\_)} s, m \\ (b_i)_{i\in I} \text{ is a valid } \mathsf{CS} \text{ partition} \\ \text{There exists } (s_i, m_i)_{i\in I} \text{ s.t. for every } i \in I \\ \quad [b_i]s_i = [b_i]s \wedge [b_i]m_i = [b_i]m \\ \quad S_i = \mathsf{strict\text{-}set\text{-}mac}_{\mathsf{k}}(s_i, m_i) \end{cases}$$

Proposition 2.10. *The* P-EUF-MAC *axioms are valid in any computational model where* $(\mathsf{Mac}, \mathsf{Verify})$ *is interpreted as an* EUF-CMA *secure function and satisfies* (2.13).

Proof. First, we are going to show that the following axioms are a logical consequences of the axioms EUF-MAC and the structural axioms $\mathsf{Ax}_{\mathsf{struct}}$.

$$\overline{\mathsf{Verify}(m,s,\mathsf{k}) \;\dot{\to}\; \bigvee_{u\in S} s \doteq \mathsf{Mac}_{\mathsf{k}}(u)} \quad \text{when} \begin{cases} \mathsf{k} \sqsubseteq_{\mathsf{Mac}.(\_)} s,m \\ S \equiv \mathsf{strict\text{-}set\text{-}mac}_{\mathsf{k}}(s,m) \end{cases} \tag{2.15}$$

Assuming the axioms above are valid, it is easy to conclude by repeating the proof of Proposition 2.9, using the axioms above instead of EUF-MAC.

To show that the axioms in (2.15) are admissible, we are going to pull out all conditionals using the properties of the if_then_else_ function symbols. This yields a term of the form $C[\vec{\beta}\diamond\vec{e}]$ where the terms $\vec{e}$ are of the form $\mathsf{Verify}(u',s',\mathsf{k})$. Then, we apply the EUF-MAC axioms to every $e\in\vec{e}$. Finally, we rewrite back the conditionals. To be able to do this last step, we need, when we pulled out the conditionals, to remember which conditional appeared where. We do this by replacing a conditional b with either true_b or false_b, where the lower-script b is a label that we attach to the term.

This motivates the following definition: for every boolean term b, we let $\mathsf{Val}_b = \{\mathsf{true}_b, \mathsf{false}_b\}$. We extend this to vector of conditionals by having $\mathsf{Val}_{u_0,\dots,u_l} = \mathsf{Val}_{u_0}\times\cdots\times\mathsf{Val}_{u_l}$. Basically, for every vector of conditionals $\vec{\beta}$, choosing a vector of terms $\vec{\nu}\in\mathsf{Val}_{\vec{\beta}}$ correspond to choosing a valuation of $\vec{\beta}$.

We start showing the validity of (2.15). Let $\vec{\beta}$ be the set of conditionals appearing in s,m, and C be an if-context such that:

$$\mathsf{Verify}(m,s,\mathsf{k}) \leftrightarrows C\Big[\vec{\beta}\diamond\big(\mathsf{Verify}(m[\vec{\nu}/\vec{\beta}],s[\vec{\nu}/\vec{\beta}],\mathsf{k})\big)_{\vec{\nu}\in\mathsf{Val}_{\vec{\beta}}}\Big]$$

where $t[\vec{u}/\vec{v}]$ denotes the substitution of every occurrence of $\vec{v}$ by $\vec{u}$ in t. For every $\vec{\nu}\in\mathsf{Val}_{\vec{\beta}}$, let $S_{\vec{\nu}} = \mathsf{set\text{-}mac}_{\mathsf{k}}(s[\vec{\nu}/\vec{\beta}],m[\vec{\nu}/\vec{\beta}])$. By applying EUF-MAC to every $\mathsf{Verify}(m[\vec{\nu}/\vec{\beta}],s[\vec{\nu}/\vec{\beta}],\mathsf{k})$ we get:

$$\mathsf{Verify}(m,s,\mathsf{k}) \;\dot{\to}\; C\Big[\vec{\beta}\diamond\big(\dot{\bigvee}_{u\in S_{\vec{\nu}}} s[\vec{\nu}/\vec{\beta}]\doteq\mathsf{Mac}_{\mathsf{k}}(u)\big)_{\vec{\nu}\in\mathsf{Val}_{\vec{\beta}}}\Big]$$

Since any conditional of $s[\vec{\nu}/\vec{\beta}]$ or $m[\vec{\nu}/\vec{\beta}]$ is of the form true_x or false_x for some label x, we know that:

$$S_{\vec{\nu}} = \mathsf{set\text{-}mac}_{\mathsf{k}}(s[\vec{\nu}/\vec{\beta}],m[\vec{\nu}/\vec{\beta}]) = \mathsf{strict\text{-}set\text{-}mac}_{\mathsf{k}}(s[\vec{\nu}/\vec{\beta}],m[\vec{\nu}/\vec{\beta}])$$

Moreover, we can check that:

$$\mathsf{strict\text{-}set\text{-}mac}_{\mathsf{k}}(s[\vec{\nu}/\vec{\beta}],m[\vec{\nu}/\vec{\beta}]) = (\mathsf{strict\text{-}set\text{-}mac}_{\mathsf{k}}(s,m))\,[\vec{\nu}/\vec{\beta}]$$

Let $S = \mathsf{strict\text{-}set\text{-}mac}_{\mathsf{k}}(s,m)$, we just showed that $S_{\vec{\nu}} = S[\vec{\nu}/\vec{\beta}]$. Hence:

$$\begin{aligned} C\Big[\vec{\beta}\diamond\big(\dot{\bigvee}_{u\in S_{\vec{\nu}}} s[\vec{\nu}/\vec{\beta}]\doteq\mathsf{Mac}_{\mathsf{k}}(u)\big)_{\vec{\nu}\in\mathsf{Val}_{\vec{\beta}}}\Big] &\;\dot{\to}\; C\Big[\vec{\beta}\diamond\big(\dot{\bigvee}_{u\in S[\vec{\nu}/\vec{\beta}]} s[\vec{\nu}/\vec{\beta}]\doteq\mathsf{Mac}_{\mathsf{k}}(u)\big)_{\vec{\nu}\in\mathsf{Val}_{\vec{\beta}}}\Big] \\ &\;\dot{\to}\; C\Big[\vec{\beta}\diamond\big(\big(\dot{\bigvee}_{u\in S} s\doteq\mathsf{Mac}_{\mathsf{k}}(u)\big)[\vec{\nu}/\vec{\beta}]\big)_{\vec{\nu}\in\mathsf{Val}_{\vec{\beta}}}\Big] \\ &\;\dot{\to}\; \dot{\bigvee_{u\in S}}\, s\doteq\mathsf{Mac}_{\mathsf{k}}(u) \end{aligned}$$

■

CR-KEY$_{\neq}$ Axioms Finally, we have an axiom stating that two macs generated with distinct random keys cannot be equal.

Definition 2.29. We let CR-KEY$_{\neq}$ be the set of axioms:

$$\overline{\mathsf{Mac}_{\mathsf{k}}(u)\doteq\mathsf{Mac}_{\mathsf{k}'}(v)\dot{\to}\mathsf{false}} \quad \text{when} \begin{cases} \mathsf{k},\mathsf{k}' \sqsubseteq_{\mathsf{Mac}.(\_)} u,v \\ \mathsf{k},\mathsf{k}'\in\mathcal{N}, \mathsf{k}\not\equiv\mathsf{k}' \end{cases}$$

Proposition 2.11. *The* CR-KEY$_{\neq}$ *axioms are valid in any computational model where* $(\mathsf{Mac},\mathsf{Verify})$ *is interpreted as an* EUF-CMA *secure function.*

We now give the proof of the above proposition.[4]

[4]This proof is due to Bruno Blanchet.

Proof. Assume that there exists a computational model $\mathcal{M}_{\mathsf{c}}$ and an instance:

$$\frac{}{\mathsf{Mac}_{\mathsf{k}}(u) \doteq \mathsf{Mac}_{\mathsf{k}'}(v) \dashrightarrow \mathsf{false}}$$

of the CR-KEY$_{\neq}$ axioms which is not valid in $\mathcal{M}_{\mathsf{c}}$. Then we know that $[\![\mathsf{Mac}_{\mathsf{k}}(u)]\!]$ and $[\![\mathsf{Mac}_{\mathsf{k}'}(v)]\!]$ coincide on a non-negligible number of samplings. W.l.o.g. we assume that $|u|+|v|$ is minimal among all instances of the axioms that are not valid. First, remark that:

$$\begin{aligned} \mathsf{Mac}_{\mathsf{k}}(u) \doteq \mathsf{Mac}_{\mathsf{k}'}(v) &\dashrightarrow \mathsf{Verify}(v, \mathsf{Mac}_{\mathsf{k}}(u), \mathsf{k}') \\ &\dashrightarrow \bigvee_{v' \in S} v \doteq v' \qquad \text{(where } S = \mathsf{set\text{-}mac}_{\mathsf{k}'}(v, u)\text{)} \end{aligned}$$

using the EUF-MAC and CR axioms.[5] Hence:

$$\mathsf{Mac}_{\mathsf{k}}(u) \doteq \mathsf{Mac}_{\mathsf{k}'}(v) \dashrightarrow \bigvee_{v' \in S} \mathsf{Mac}_{\mathsf{k}}(u) \doteq \mathsf{Mac}_{\mathsf{k}'}(v')$$

Since v' is a strict subset of u or v, we know that for every $v' \in S$, $|u|+|v'| < |u|+\max(|u|,|v|)$. Since S is a finite set, and since $\mathsf{Mac}_{\mathsf{k}}(u) \doteq \mathsf{Mac}_{\mathsf{k}'}(v)$ is valid for a non-negligible number of samplings, we know that there exists some $v' \in S$ such that $\mathsf{Mac}_{\mathsf{k}}(u) \doteq \mathsf{Mac}_{\mathsf{k}'}(v')$ is valid for a non-negligible number of samplings.

By the same reasoning:

$$\begin{aligned} \mathsf{Mac}_{\mathsf{k}}(u) \doteq \mathsf{Mac}_{\mathsf{k}'}(v) &\dashrightarrow \mathsf{Verify}(u, \mathsf{Mac}_{\mathsf{k}'}(v), \mathsf{k}) \\ &\dashrightarrow \bigvee_{u' \in S'} \mathsf{Mac}_{\mathsf{k}}(u') \doteq \mathsf{Mac}_{\mathsf{k}'}(v) \qquad \text{(where } S' = \mathsf{set\text{-}mac}_{\mathsf{k}}(u, v)\text{)} \end{aligned}$$

where, for every $u' \in S'$, $|u'| + |v| < |v| + \max(|u|,|v|)$. Again, since S' is finite, there exists $u' \in S'$ such that $\mathsf{Mac}_{\mathsf{k}}(u') \doteq \mathsf{Mac}_{\mathsf{k}'}(v)$ is valid for a non-negligible number of samplings.

Therefore we can always pick u', v' such that $|u'| + |v'| < |u| + |v|$ and $\mathsf{Mac}_{\mathsf{k}}(u') \doteq \mathsf{Mac}_{\mathsf{k}'}(v')$ is valid for a non-negligible number of samplings (if $|u| \leq |v|$, we take $u' = u$ and $v' \in S$, and if $|v| \leq |u|$, we take $u' \in S'$ and $v' = v$). Contradiction. ■

2.6.4 PRF Axioms

We now present the axioms we designed for keyed hash functions satisfying the *Pseudo Random Function* (PRF) assumption. Informally, a keyed hash function $\mathsf{H}(\cdot, \mathsf{k})$ is a PRF if its outputs are computationally indistinguishable from the outputs of a random function. Formally:

Definition 2.30 (PRF [Gol01, GGM86])**.** Let $\mathsf{H}(\cdot,\cdot) : \{0,1\}^* \times \{0,1\}^\eta \to \{0,1\}^\eta$ be a keyed hash function. The function H is a *Pseudo Random Function* iff, for any PPTM adversary $\mathcal{A}$ with access to an oracle $\mathcal{O}_f$:

$$|\mathbf{Pr}(k : \mathcal{A}^{\mathcal{O}_{\mathsf{H}(\cdot,k)}}(1^\eta) = 1) - \mathbf{Pr}(g : \mathcal{A}^{\mathcal{O}_{g(\cdot)}}(1^\eta) = 1)|$$

is negligible, where:

- k is drawn uniformly in $\{0,1\}^\eta$.
- g is drawn uniformly in the set of all functions from $\{0,1\}^*$ to $\{0,1\}^\eta$.

Here are the axioms:

Definition 2.31. We let PRF be the set of axioms:

$$\frac{}{\begin{array}{l} \vec{u}, \mathsf{if} \ \dot{\bigvee}_{i \in I} m \doteq m_i \ \mathsf{then}\ \mathbf{0}\ \mathsf{else}\ \mathsf{H}(m, \mathsf{k}) \\ \sim \ \vec{u}, \mathsf{if} \ \dot{\bigvee}_{i \in I} m \doteq m_i \ \mathsf{then}\ \mathbf{0}\ \mathsf{else}\ \mathsf{n} \end{array}} \quad \text{when} \begin{cases} \mathsf{fresh}(\mathsf{n}; \vec{u}, m) \\ \mathsf{k} \sqsubseteq_{\mathsf{H}(\_,\cdot)} \vec{u}, m \\ \{m_i \mid i \in I\} = \{u \mid \mathsf{H}(u, \mathsf{k}) \in \mathsf{st}(\vec{u}, m)\} \end{cases}$$

Proposition 2.12. *The* PRF *axioms are valid in any computational model where* H *is interpreted as a* PRF *function.*

[5] Which is valid, as an EUF-CMA secure function is also CR-HK. We give a proof of (a generalization of) this in Section 4.7.2.

Proof. Consider a computational model $\mathcal{M}_{\mathsf{c}}^0$ where H is interpreted as a PRF function, and an instance of the axiom schema which is not valid in $\mathcal{M}_{\mathsf{c}}^0$:

$$\frac{}{\begin{array}{l}\vec{u}, \mathsf{if}\ \dot{\bigvee}_{i\in I} m \doteq m_i\ \mathsf{then}\ \mathbf{0}\ \mathsf{else}\ \mathsf{H}(m,\mathsf{k}) \\ \sim\ \vec{u}, \mathsf{if}\ \dot{\bigvee}_{i\in I} m \doteq m_i\ \mathsf{then}\ \mathbf{0}\ \mathsf{else}\ \mathsf{n}\end{array}}$$

Let $\vec{h} \equiv (\mathsf{H}(m_i,\mathsf{k}))_{i\in I}$ and $\vec{v}[], b[]$ be contexts such that $\vec{v}[\vec{h}] \equiv \vec{u}$, $b[\vec{h}] \equiv \bigvee_{i\in I} \mathsf{eq}(m, m_i)$ and such that $\mathsf{k} \notin \mathsf{st}(\vec{v}, b)$. To get a contradiction, we just have to show that:

$$\mathbf{Pr}\big(\rho_1,\rho_2 : \mathcal{A}\big(\llbracket \vec{v}[\vec{h}], [b[\vec{h}]]\mathsf{H}(m,\mathsf{k})\rrbracket^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}^0}\big) = 1\big) \approx \mathbf{Pr}\big(\rho_1,\rho_2 : \mathcal{A}\big(\llbracket \vec{v}[\vec{h}], [b[\vec{h}]]\mathsf{n}\rrbracket^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}^0}\big) = 1\big) \tag{2.16}$$

Let $\mathcal{M}_{\mathsf{c}}$ be an extension of $\mathcal{M}_{\mathsf{c}}^0$ where we added two function symbols $g, g' \in \mathcal{F}_{\mathsf{p}}$ which are interpreted as random functions. Observe that $\mathcal{M}_{\mathsf{c}}$ is not a computational model, because we require that function in $\mathcal{F}_{\mathsf{p}}$ are interpreted as *deterministic* polynomial-time functions. Still, $\mathcal{M}_{\mathsf{c}}$ is a first-order model. Moreover, $\mathcal{M}_{\mathsf{c}}$ and $\mathcal{M}_{\mathsf{c}}^0$'s interpretations coincide on terms which do not use g and g'. Hence, to prove (2.16) it is sufficient to show that:

$$\mathbf{Pr}\big(\rho_1,\rho_2 : \mathcal{A}\big(\llbracket \vec{v}[\vec{h}], [b[\vec{h}]]\mathsf{H}(m,\mathsf{k})\rrbracket^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}}\big) = 1\big) \approx \mathbf{Pr}\big(\rho_1,\rho_2 : \mathcal{A}\big(\llbracket \vec{v}[\vec{h}], [b[\vec{h}]]\mathsf{n}\rrbracket^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}}\big) = 1\big) \tag{2.17}$$

Let $\vec{r} \equiv (g(m_i))_{i\in I}$. It is straightforward to check that, thanks to the PRF assumption of H, we can replace all subterms of the form $H(\mathsf{x},\mathsf{k})$ by $g(\mathsf{x})$ on the left:

$$\mathbf{Pr}\big(\rho_1,\rho_2 : \mathcal{A}\big(\llbracket \vec{v}[\vec{h}], [b[\vec{h}]]\mathsf{H}(m,\mathsf{k})\rrbracket^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}}\big) = 1\big) \approx \mathbf{Pr}\big(\rho_1,\rho_2 : \mathcal{A}\big(\llbracket \vec{v}[\vec{r}\,], [b[\vec{r}\,]]g(m)\rrbracket^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}}\big) = 1\big)$$

Moreover, using the fact that the subterm $g(m)$ is guarded by $b[\vec{r}\,]$, we know that, except for a negligible number of samplings, m is never queried to the random function g, except once, in $[b[\vec{r}\,]]g(m)$. It follows that we can safely replace the last call to $g(m)$ by a call to $g'(m)$, which yields:

$$\mathbf{Pr}\big(\rho_1,\rho_2 : \mathcal{A}\big(\llbracket \vec{v}[\vec{r}\,], [b[\vec{r}\,]]g(m)\rrbracket^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}}\big) = 1\big) \approx \mathbf{Pr}\big(\rho_1,\rho_2 : \mathcal{A}\big(\llbracket \vec{v}[\vec{r}\,], [b[\vec{r}\,]]g'(m)\rrbracket^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}}\big) = 1\big)$$

Now, using again the PRF property of H, we know that:

$$\mathbf{Pr}\big(\rho_1,\rho_2 : \mathcal{A}\big(\llbracket \vec{v}[\vec{r}\,], [b[\vec{r}\,]]g'(m)\rrbracket^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}}\big) = 1\big) \approx \mathbf{Pr}\big(\rho_1,\rho_2 : \mathcal{A}\big(\llbracket \vec{v}[\vec{h}], [b[\vec{h}]]g'(m)\rrbracket^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}}\big) = 1\big)$$

Finally, since g' appears only once in $\vec{v}[\vec{h}], [b[\vec{h}]]g'(m)$, we can replace $g'(m)$ by a fresh nonce. Hence:

$$\mathbf{Pr}\big(\rho_1,\rho_2 : \mathcal{A}\big(\llbracket \vec{v}[\vec{h}], [b[\vec{h}]]g'(m)\rrbracket^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}}\big) = 1\big) \approx \mathbf{Pr}\big(\rho_1,\rho_2 : \mathcal{A}\big(\llbracket \vec{v}[\vec{h}], [b[\vec{h}]]\mathsf{n}\rrbracket^{\eta,\rho_1,\rho_2}_{\mathcal{M}_{\mathsf{c}}}\big) = 1\big)$$

Which concludes the proof of (2.17). ■

Remark 2.4. If we have a valid instance of PRF:

$$\frac{}{\begin{array}{l}\vec{u}, \mathsf{if}\ \dot{\bigvee}_{i\in I} m \doteq m_i\ \mathsf{then}\ \mathbf{0}\ \mathsf{else}\ \mathsf{H}(m,\mathsf{k}) \\ \sim\ \vec{u}, \mathsf{if}\ \dot{\bigvee}_{i\in I} m \doteq m_i\ \mathsf{then}\ \mathbf{0}\ \mathsf{else}\ \mathsf{n}\end{array}}\ \text{PRF}$$

then, using transitivity, we know that:

$$\frac{\dfrac{}{\begin{array}{l}\vec{u}, \mathsf{if}\ \dot{\bigvee}_{i\in I} m \doteq m_i\ \mathsf{then}\ \mathbf{0}\ \mathsf{else}\ \mathsf{H}(m,\mathsf{k}) \\ \sim\ \vec{u}, \mathsf{if}\ \dot{\bigvee}_{i\in I} m \doteq m_i\ \mathsf{then}\ \mathbf{0}\ \mathsf{else}\ \mathsf{n}\end{array}}\ \text{PRF} \qquad \vec{u}, \mathsf{if}\ \dot{\bigvee}_{i\in I} m \doteq m_i\ \mathsf{then}\ \mathbf{0}\ \mathsf{else}\ \mathsf{n} \sim \vec{v}}{\vec{u}, \mathsf{if}\ \dot{\bigvee}_{i\in I} m \doteq m_i\ \mathsf{then}\ \mathbf{0}\ \mathsf{else}\ \mathsf{H}(m,\mathsf{k}) \sim \vec{v}}\ \text{Trans}$$

Therefore the following axiom schema is admissible using PRF and the transitivity axiom Trans:

$$\frac{\vec{u}, \mathsf{if}\ \dot{\bigvee}_{i\in I} m \doteq m_i\ \mathsf{then}\ \mathbf{0}\ \mathsf{else}\ \mathsf{n} \sim \vec{v}}{\vec{u}, \mathsf{if}\ \dot{\bigvee}_{i\in I} m \doteq m_i\ \mathsf{then}\ \mathbf{0}\ \mathsf{else}\ \mathsf{H}(m,\mathsf{k}) \sim \vec{v}}\ \text{PRF} \quad \text{when} \begin{cases} \mathsf{fresh}(\mathsf{n}; \vec{u}, m) \\ \mathsf{k} \sqsubseteq_{\mathsf{H}(\_,\cdot)} \vec{u}, m \\ \{m_i \mid i \in I\} = \{u \mid \mathsf{H}(u,\mathsf{k}) \in \mathsf{st}(\vec{u}, m)\} \end{cases}$$

We will prefer the axiom schema above over the axiom schema given in Definition 2.31. By notation abuse, we also refer to the above axioms as PRF. □

2.7 Conclusion

We presented the syntax and semantics of the Bana-Comon logic for indistinguishability. We also defined computational models as a special case of sorted first-order models, where terms are interpreted as probabilistic polynomial-time Turing machines and $\sim$ is interpreted as computational indistinguishability.

Secondly, we defined protocols as infinite but finitely branching labelled transition systems. We gave two semantics for protocols: a computational semantics where the adversary adaptively chooses the next action to execute, and a symbolic semantics where the action sequence is fixed in advance. Because we require protocols to be finitely branching, we showed that these two notion are related: indistinguishability in the symbolic semantics, in some computational model $\mathcal{M}_{\mathsf{c}}$, implies indistinguishability in the computational semantics in $\mathcal{M}_{\mathsf{c}}$. Moreover, showing that two protocols are indistinguishable in $\mathcal{M}_{\mathsf{c}}$, for the symbolic semantics, immediately translates into the Bana-Comon logic: it amounts to proving validity of some (infinite) set of formulas in $\mathcal{M}_{\mathsf{c}}$.

This definition of protocols as labelled transition systems is generic. We believe it can capture any notion of security appearing in the literature. We support this claim in the next two chapters by constructing labelled transition systems for, in Chapter 3, Juels and Weis notion of Privacy [JW09], and in Chapter 4, a variant of Vaudenay's unlinkability [Vau07].

Finally, we explained how to restrict the models that have to be considered when proving actual protocols. We do this through axioms: on the one hand, structural axioms are valid in any computational model, and therefore can be safely added; on the other hand, implementation axioms forbid some computational models and reflect properties that must hold in any concrete implementation of the protocol studied. We designed several useful sets of implementation axioms for pairs, decryption, xor and boolean functions. Moreover, we translated four standard cryptographic assumptions into axioms: Indistinguishability against Chosen-Ciphertexts Attacks, Collision-Resistance under Hidden-Key attacks, Unforgeability against Chosen-Message Attacks and Pseudo Random Functions.

Chapter 3

Privacy Proofs of RFID Protocols

In this chapter, we illustrate the usefulness of the Bana-Comon approach and the axioms we designed in Chapter 2, by proving the security of two RFID protocols (more precisely their privacy). RFID protocols are usually simple protocols, due to the low computing capabilities of a RFID tag: the protocols mostly rely on hashing, xoring and concatenation. This is why they are a useful first application of the model: we do not need complex axioms, and security proofs remain tractable.

Contributions Our contributions are:

- First, to express computational privacy. There are various definitions of privacy for RFID protocols. We choose to formalize the notion of Privacy from [JW09]. As usual in computational security, this is a game-based definition, where an adversary tries to guess the challenger internal bit b. The game is designed in such a way that guessing the bit b amount to guessing some tag's identity.

  Of course, other definitions can be expressed in a similar way.[1] We follow the approach presented in Chapter 2:

  - given a protocol P, we define a labelled transition system priv-lts$_b$(P). Here, b is a boolean parameter which corresponds to the challenger internal bit in the Privacy game. We actually go one step further: we let priv-lts$_b^{n,m}$(P) be the restriction of priv-lts$_b$(P) to some given set of traces, called (n, m)-privacy traces. This restricted LTS captures exactly the notion of Privacy for n tags and m interactions between the adversary and the reader and tags.
  - using Theorem 2.1, we know that to show Privacy for n tags and m interactions in some computational model $\mathcal{M}_{\mathsf{c}}$, it is sufficient to prove that for every (n, m)-privacy trace τ:

$$\mathcal{M}_{\mathsf{c}} \models \phi_\tau^{\text{priv-lts}_{\mathbf{true}}(\mathrm{P})} \sim \phi_\tau^{\text{priv-lts}_{\mathbf{false}}(\mathrm{P})}$$

- We use this proof technique on two examples taken from [VDR08]: KCL [KCL07] and LAK [LAK06]. As far as we know, all published RFID protocols, that do not rely on encryption, are computationally insecure. This is also the case of these two protocols. We propose modified versions of the protocols, KCL$^+$ and LAK$^+$, which prevent the known attacks. Some of the modified versions are secure in the Dolev-Yao model. Depending on the assumptions on the primitives, they may however be insecure in the computational model. For instance, if we assume the hash function to be pre-image resistant and one-way, the corrected version of LAK, proved in the symbolic model in [HBD16], is not necessarily computationally secure: there might be attacks on both authentication and unlinkability. We actually need a family of keyed hash functions, which satisfies the PRF assumption. With the appropriate implementation assumptions, we formally prove the security of the two protocols. For LAK$^+$, we prove Privacy for two tags and six interactions, and for KCL$^+$, we prove Privacy for two tags and any number of interactions. The latter proof is by induction on the number of interactions. This is a proof technique that we will use again, on a larger scale, to prove unlinkability of a variant of the AKA protocol in Chapter 4.

[1] E.g., we will express a variant of Vaudenay's unlinkability [Vau07] in Chapter 4

Related Work RFID protocols have been studied, attacked, patched and automatically proved in the Dolev-Yao model (see for instance [HBD16]). On the computational side, [Vau07] investigates the computational definitions of unlinkability, together with examples of RFID protocols that satisfy (or not) the definitions. There are however very few proofs of security in the computational model and, to our knowledge, no *formal* security proof. For instance, an RFID protocol is proposed in [LBdM07], together with a (claimed) universally composable proof. The proof is however quite informal, and attacks were found on this protocol (see [OP08]). Admittedly, such attacks can be easily circumvented, but this shows that a formal approach is useful, if not necessary. Similarly, as reported in [JW09], other RFID protocols that were claimed secure turned out to be broken.

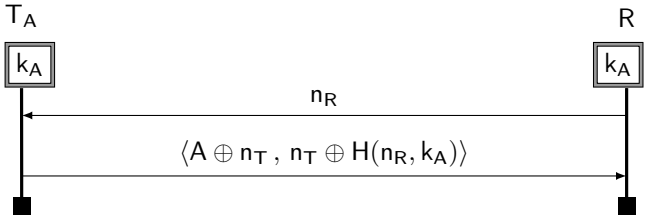
A large fraction of RFID protocols, the so-called *Ultralightweight* RFID protocols (e.g. [Chi07] and [PCER08]), aim at ensuring only weak security properties against passive attackers, because of the strong constraints on the number of gates in the RFID tags. We do not consider such protocols here.

Outline In Section 3.1 we recall the definition of privacy of a RFID protocol given by Juels and Weis in [JW09], and we show how this property can be translated as a labelled transition system. In Section 3.2 we describe the KCL and LAK protocols, we give known attacks on them and formally prove the security of fixed versions of the protocols. We also show that relaxing the assumptions yields some attacks. Finally, in Section 3.3, we show (as expected) that abstracting pseudo-random numbers with random numbers is sound, provided that the seed is not used for any other purpose.

3.1 Security Properties

Radio Frequency IDentification (RFID) systems allow to wirelessly identify objects. These systems are composed of *readers* and *tags*. Readers are radio-transmiters connected through a secure channel to a single server hosting a database with all the tracked objects information. Tags are wireless transponders attached to physical objects that have a limited memory and computational capacity (to reduce costs). To keep things simple, we assume a setting with a single reader, which represents both the database and the physical radio-transmitters.

Example 3.1. As an example, we use a simple version of the KCL protocol. The original protocol from [KCL07] is informally described below:



The key k_A is a shared secret key between the tag T_A and the reader R. Names n_T, n_R are randomly generated by, respectively, the tag and the reader, at the beginning of the protocol; this will be justified in Section 3.3. The protocol is expected to ensure both authentication and unlinkability. □

3.1.1 Privacy of RFID Protocols

We use the notion of Privacy for RFID protocols from Juels and Weis [JW09], which we informally recall. This is a game-based definition, in which the adversary is a probabilistic polynomial-time Turing machine interacting with a reader R and a finite set of tags $\{T_1, \ldots, T_n\}$ (also probabilistic Turing machines). The interactions between the adversary and the agents are through a fixed communication interface, which is described below and in Figure 3.1:

- A tag T_i stores a secret key k_i, an identity ID_i, a session identifier *sid* and the previous l challenge-response pairs of the current session. It has the following interface:
  - SetKey: Corrupts the tag by returning its old key k_i and identity ID_i, and allows the adversary to choose a new key k_i' and a new id ID_i'.

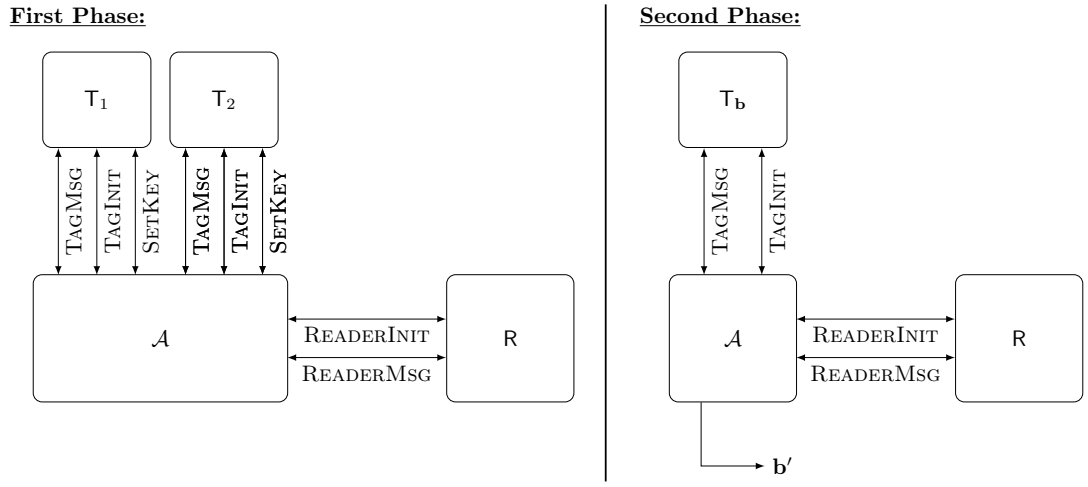


Figure 3.1: Privacy game with two tags $\mathsf{T}_1, \mathsf{T}_2$. The adversary $\mathcal{A}$ wins if $\mathbf{b} = \mathbf{b}'$.

  - TagInit: Initialize a tag with a session identifier sid'. The tag deletes the previous session identifier and the logged challenge-response pairs.
  - TagMsg: The tag receives a challenge c_i and returns a response r_i (that was computed using the key, the session identifier and the logged challenge-response pairs). Additionally, the tag logs the challenge-response pair (c_i, r_i).
- The reader R stores some private key material (for example a master secret key, the tags private keys ...) and entries of the form $(sid, status, c_0, r_0, \ldots, c_l)$ where *status* is either *open* or *closed* depending on whether the session is completed or on-going. It has the following interface:
  - ReaderInit: Returns a fresh session identifier sid[2] along with the first challenge c_o. The reader also stores a new entry of the form $(sid, open, c_o)$.
  - ReaderMsg: The reader receives a session identifier sid and a response r_i. It looks for a data entry of the form $(sid, open, c_0, r_0, \ldots, c_i)$, appends the message r_i to the data entry, and either closes the session (by changing the status from *open* to *closed*) or outputs a new challenge message c_{i+1} (possibly 0) and appends it to the data entry.

The adversary is allowed to corrupt (by a SetKey command) up to $n-2$ tags. At the end of a first phase of computations and interactions with the reader R and tags $\{\mathsf{T}_1, \ldots, \mathsf{T}_n\}$, the tags T_{n-1} and T_n are removed from the set of available tags. The adversary is not allowed to corrupt the tags T_{n-1} and T_n during the first phase. Then one of these tags is chosen uniformly at random by sampling a bit b and made accessible to the adversary as an oracle. The adversary performs a second phase of computations and interactions with the reader R, the tags $\{\mathsf{T}_1, \ldots, \mathsf{T}_{n-2}\}$, as well as the randomly selected tag T_{n-1+b} (obviously the adversary is not allowed to corrupt T_{n-1+b}). Finally the adversary outputs a bit b', and wins if it guessed the chosen tag (that is if $b = b'$). A protocol is said to verify m-Privacy if any adversary $\mathcal{A}$ using at most m calls to the interfaces, has a probability of winning the game bounded by $\frac{1}{2} + f_{\mathcal{A}}(\eta)$, where $f_{\mathcal{A}}$ is a negligible function in the security parameter. $f_{\mathcal{A}}(\eta)$ is the advantage of $\mathcal{A}$ against the m-Privacy game.

Remark 3.1. Our definition of privacy is slightly different from the one in [JW09]:

- We do not assume that the reader answers "reject" or "accept" when a session is completed. We can easily encode this feature by adding an answer from the reader at the end of the protocol with the corresponding message. Not taking this as the default behavior allows to model adversaries that are less powerful and do not have access to the result of the protocol.
- We use m-Privacy, whereas [JW09] uses (r, s, t)-Privacy where r and s are a bound on the number of calls to ReaderInit and TagInit respectively, and t is a bound on the running time. We dropped

[2] We use as session identifier the number of interactions with the agents since the game started.

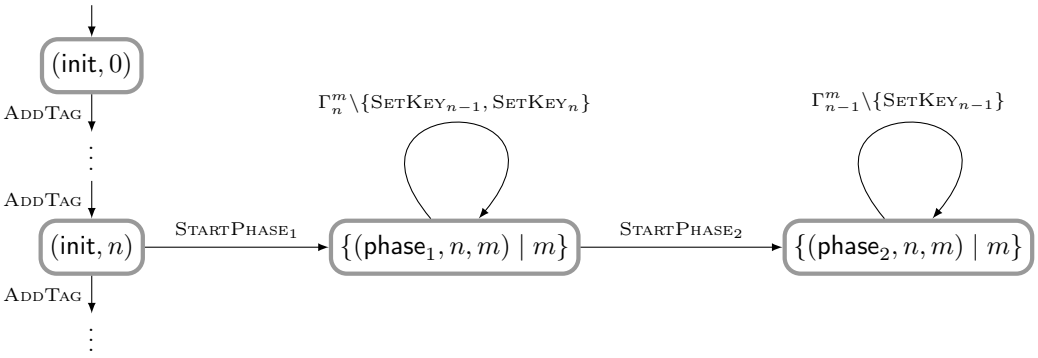


Figure 3.2: The Labelled Transition System priv-lts$_b$(P) for Bouded Session Privacy.

the explicit mention of t as we are only interested in proving privacy against any polynomial time adversary. Moreover using m or r, s is equivalent, as, for a given protocol, the number of calls to the interfaces is bounded by the number of calls to ReaderInit and to TagInit, and conversely.

- In [JW09], at the end of the first phase, the adversary chooses two uncorrupted tags T_{i_0} and T_{i_1}, which are removed from the set of available tags. Then one of these tags is made accessible through an oracle depending on the internal bit b. We use a simpler definition, and always remove the tags T_{n-1} and T_n. When considering attacks with a finite number of interactions between the adversary and the reader and tags, both definitions coincide: this is just a renaming of the tags. □

3.1.2 Privacy Labelled Transition System

We now construct a labelled transition system priv-lts(P) corresponding to the Privacy game for a protocol P. Actually, we define simultaneously two LTS using an internal bit b, which corresponds to the Privacy game internal bit. The LTS priv-lts$_b$(P) is depicted in Figure 3.2, and defined below.

- In a zeroth phase, we let the adversary choose the number of tags. We let $\mathcal{Q}_0$ be the set of nodes $\{(\mathsf{init}, i) \mid i \in \mathbb{N}\}$. Intuitively, we are in node (init, i) if there are currently i tags. For every $(\mathsf{init}, i) \in \mathcal{Q}_0$, we have a transition adding a tag:

$$(\textsc{AddTag}, \mathsf{void}, \epsilon, (\mathsf{init}, i+1)) \in \delta(\mathsf{init}, i)$$

  where void is a constant function symbol. Initially, we are in the state $q_\epsilon = (\mathsf{init}, 0)$. The protocol specification must contain the initial internal memory σ_ϵ. Moreover, the set of state variables Vars_σ must contain, for every i, at least the variables:
  - $x_{\mathsf{k}_i}, x_{\mathrm{ID}_i}$ storing, respectively, the key and the identity of tag T_i.
  - $(x_{i,j})_{j \le L}$ storing the j-th challenge-response of tag T_i. Remark that the protocol must have a finite number of challenge-response phase L.
- At any time, we can stop adding tags and start the first phase of the protocol. We let $\mathcal{Q}_1 = \{(\mathsf{phase}_1, i, m) \mid i, m \in \mathbb{N}\}$ be the set of nodes of phase one. The integer i is the number of tags that were added in the zeroth phase, and m is a counter which is incremented at every transitions. We use m to ensure freshness of names (by indexing them with m), and to upper-bound the number of sessions of the reader. For every $i, m \in \mathbb{N}$, we have the transition:

$$(\textsc{StartPhase}_1, \mathsf{void}, \epsilon, (\mathsf{phase}_1, i, 0)) \in \delta(\mathsf{init}, i)$$

- In the first phase we let the adversary interacts with the reader and the tags. We let Γ_n^m be the set of possible actions of an adversary interacting with n tags and m_R reader sessions:

$$\Gamma_n^{m_R} = \{\textsc{SetKey}_i, \textsc{TagInit}_i, \textsc{TagMsg}_i, \textsc{ReaderInit}, \textsc{ReaderMsg}_j \mid 1 \le i \le n, 1 \le j \le m_R\}$$

  The protocol specification must comprise, for every action $\alpha \in \Gamma$, a term t_α representing the answer of the reader or tag to the action α. We assume that t_α contains only fresh names[3]. If

[3] We use the integer counter m in the node to index names in t_α.

we want to re-use a name, we need to store it in a state variable. We also have a state update $\sigma_\alpha^{\mathsf{up}}$ representing the modifications to the reader and tags internal memory when executing α. In phase one, we can execute any action of $\Gamma_n^{m_R}$, except corrupting the tags T_{n-1} or T_n. For every $\alpha \in \Gamma_n^{m_R} \backslash \{\textsc{SetKey}_{n-1}, \textsc{SetKey}_n\}$, we have the transition:

$$(\alpha, t_\alpha, \sigma_\alpha^{\mathsf{up}}, (\mathsf{phase}_1, n, m+1)) \in \delta(\mathsf{phase}_1, n, m)$$

- We start phase two of the privacy game whenever we want. Let $\mathcal{Q}_2 = \{(\mathsf{phase}_2, i, m) \mid i, m \in \mathbb{N}\}$, and for every $i, m \in \mathbb{N}$ we have the transition:

$$\big(\textsc{StartPhase}_2, \mathsf{void}, \sigma_{\mathsf{prep}}^n, (\mathsf{phase}_2, i, m)\big) \in \delta(\mathsf{phase}_1, i, m)$$

  Where σ_{prep}^n sets all logged challenge-response pairs of tags T_{n-1} and T_n to unset, and keep the tag T_{n-1} or T_n according to the internal bit b:

$$\sigma_{\mathsf{prep}}^n(\mathsf{x}) = \begin{cases} \mathsf{unset} & \text{if } \mathsf{x} = x_{n-1,j} \text{ or } x_{n,j}, \text{ where } j \leq L \\ x_{\mathsf{k}_{n-1+b}} & \text{if } \mathsf{x} = x_{\mathsf{k}_{n-1}} \\ x_{\mathrm{ID}_{n-1+b}} & \text{if } \mathsf{x} = x_{\mathrm{ID}_{n-1}} \end{cases}$$

- Phase two works like phase one, except that we have one less tag and are not allowed to corrupt the tag T_{n-1}. Therefore, for every $\alpha \in \Gamma_{n-1}^{m_R} \backslash \{\textsc{SetKey}_{n-1}\}$, we have the transition:

$$(\alpha, t_\alpha, \sigma_\alpha^{\mathsf{up}}, (\mathsf{phase}_2, n, m+1)) \in \delta(\mathsf{phase}_2, n, m)$$

Example 3.2. Let us return to Example 3.1. Each tag T_{A_i} has an identifier A_i and a key k_{A_i}. In the KCL protocol the $\textsc{TagInit}_i$ call is useless because the tag has only one message to send in a round of the protocol ($\textsc{TagInit}_i$ is used to tell a tag to stop the current round of the protocol and to start a new one). We describe the terms t_α and state updates $\sigma_\alpha^{\mathsf{up}}$ when in state (phase_1, i, m) or (phase_2, i, m):

- $t_{\textsc{SetKey}_i} = \langle x_{\mathsf{k}_i}, x_{\mathrm{ID}_i} \rangle$: the data of the tag i are disclosed.
- $\sigma_{\textsc{SetKey}_i}^{\mathsf{up}} = \{x_{\mathsf{k}_i} \mapsto g_{\mathrm{KEY}_i}(\mathsf{x_{in}}), x_{\mathrm{ID}_i} \mapsto g_{\mathrm{ID}_i}(\mathsf{x_{in}})\}$: the key and id of the tag i are set to values chosen by the attacker ($g_{\mathrm{KEY}_i}, g_{\mathrm{ID}_i} \in \mathcal{G}$).
- $t_{\textsc{TagMsg}_i} = \langle x_{\mathrm{ID}_i} \oplus \mathsf{n_T}, \mathsf{n_T} \oplus \mathsf{H}(\mathsf{x_{in}}, x_{\mathsf{k}_i}) \rangle$: the reply of the tag i follows the protocol, according to its local store.
- $\sigma_{\textsc{TagMsg}_i}^{\mathsf{up}} = \epsilon$: there is no update in this case (nothing is stored for further verifications in this particular protocol)
- $t_{\textsc{ReaderInit}} = \langle m, \mathsf{n}_\mathsf{R}^m \rangle$: when starting a new session, the reader sends the session identifier m and a new challenge n_R^m.
- $\sigma_{\textsc{ReaderInit}}^{\mathsf{up}}$ updates the local memory of the reader:

$$\sigma_{\textsc{ReaderInit}}^{\mathsf{up}} = c^m \mapsto \mathsf{n}_\mathsf{R}^m$$

□

Privacy Traces A trace of actions τ of priv-lts$_b$(P) is uniquely characterized by:

- The number of tags n, which is the number of actions AddTag in τ.
- The number of interactions in the first phase p, which is the number of actions between $\textsc{StartPhase}_1$ and $\textsc{StartPhase}_2$.
- The number of interactions in the second phase q, which is the number of actions after $\textsc{StartPhase}_2$.
- The sequence of actions $(\alpha_i)_{1 \leq i \leq p+q}$ in:

$$\big(\Gamma_n^l \backslash \{\textsc{SetKey}_{n-1}, \textsc{SetKey}_n\}\big)_{1 \leq l \leq p} \times \Big(\Gamma_{n-1}^{p+l} \backslash \{\textsc{SetKey}_{n-1}\}\Big)_{1 \leq l \leq q}$$

We call such a trace a (n, p, q)-privacy trace. We also use the name of (n, m)-privacy trace (where $m = p + q$), when we do not care about the precise splitting of actions between phase one and phase two.

Using this, we can define privacy of a RFID protocol for a given number of tags n and interactions with the adversary.

Definition 3.1 (m-Fixed Trace Privacy)**.** Given an Ax-interpretation $\mathcal{I}_c$ of the function symbols in $\mathcal{F}_p$, a protocol P satisfies m-Fixed Trace Privacy for n tags if for every (n,m)-privacy trace τ and computational models $\mathcal{M}_c$ extending $\mathcal{I}_c$, we have:

$$\mathcal{M}_c \models \phi_\tau^{\text{priv-lts}_{\mathsf{true}}(\mathrm{P})} \sim \phi_\tau^{\text{priv-lts}_{\mathsf{false}}(\mathrm{P})}$$

We can now state the soundness theorem linking Fixed Trace Privacy to Juels and Weis's Privacy.

Theorem 3.1. *Let $\mathcal{I}_c$ be an Ax-interpretation of the function symbols in $\mathcal{F}_p$. If a protocol P satisfies m-Fixed Trace Privacy for n tags in $\mathcal{I}_c$ then it satisfies m-Privacy for n tags in $\mathcal{I}_c$.*

Proof. Let $\mathcal{I}_c$ be an Ax-interpretation of the function symbols in $\mathcal{F}_p$. Then a protocol P verifies m-Privacy with n tags in $\mathcal{I}_c$ if and only if for every adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ against the m-Privacy game is negligible. The conjunction of an interpretation $\mathcal{I}_c$ of the function symbols in $\mathcal{F}_p$ and an adversary $\mathcal{A}$ yields a computational model $\mathcal{M}_c^{\mathcal{A}}$ extending $\mathcal{I}_c$. Let priv-lts$_b^{n,m}$(P) be the restriction of priv-lts$_b$(P) to (n,m)-privacy traces. Then a protocol P verifies m-Privacy with n tags if and only if for every adversary $\mathcal{A}$:

$$\text{priv-lts}_{\mathsf{true}}^{n,m}(\mathrm{P}) \approx_{\mathcal{M}_c^{\mathcal{A}}} \text{priv-lts}_{\mathsf{false}}^{n,m}(\mathrm{P}) \tag{3.1}$$

Then, using Theorem 2.1, we know that to have (3.1) it is sufficient to show that for every τ:

$$\mathcal{M}_c^{\mathcal{A}} \models \phi_\tau^{\text{priv-lts}_{\mathsf{true}}^{n,m}(\mathrm{P})} \sim \phi_\tau^{\text{priv-lts}_{\mathsf{false}}^{n,m}(\mathrm{P})}$$

Or equivalently, for every m, (n,m)-privacy trace τ, we have to show that:

$$\mathcal{M}_c^{\mathcal{A}} \models \phi_\tau^{\text{priv-lts}_{\mathsf{true}}(\mathrm{P})} \sim \phi_\tau^{\text{priv-lts}_{\mathsf{false}}(\mathrm{P})}$$

■

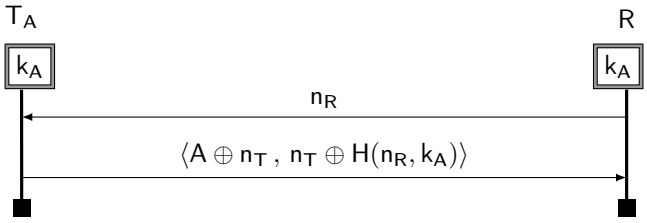
3.2 Two RFID Protocols

We are now going to describe two RFID protocols, LAK and KCL, as well as attacks, patches and security proofs of the fixed versions.

We first consider that names are randomly generated numbers, even though, because of the limited computing capabilities of the tags, they have to be implemented using a Cryptographic Pseudo-Random Number Generator (PRNG). This issue will be discussed in the Section 3.3: we will show that we can always safely abstract the pseudo random numbers as random numbers, provided that a PRNG is used and the random seed is never used for other purposes.

3.2.1 A Known Attack on KCL

Let us return to the example of the KCL protocol:



As reported in [VDR08], there is an attack that we depict in Figure 3.3. In this attack the tag is challenged twice with the same name: observing the exchanges between the tag and the reader, the adversary can replay the name. Finally the adversary checks if he is talking with the same tag by xoring the two components of the message sent by the second tag, and verifies whether the result is the same as what he obtained with the same operation in the first session.

In the left execution, the xor of the two part of the tag answers is the same:

$$\begin{aligned}\mathsf{T_A} \oplus \mathsf{n_T} \oplus \mathsf{n_T} \oplus \mathsf{H}(\mathsf{n_R}, \mathsf{k_A}) &= \mathsf{T_A} \oplus \mathsf{n'_T} \oplus \mathsf{n'_T} \oplus \mathsf{H}(\mathsf{n_R}, \mathsf{k_A}) \\ &= \mathsf{T_A} \oplus \mathsf{H}(\mathsf{n_R}, \mathsf{k_A})\end{aligned}$$

Whereas, in the right execution, we obtain two values $\mathsf{T_A} \oplus \mathsf{H}(\mathsf{n_R}, \mathsf{k_A})$ and $\mathsf{T_B} \oplus \mathsf{H}(\mathsf{n_R}, \mathsf{k_B})$ which will be different with high probability.

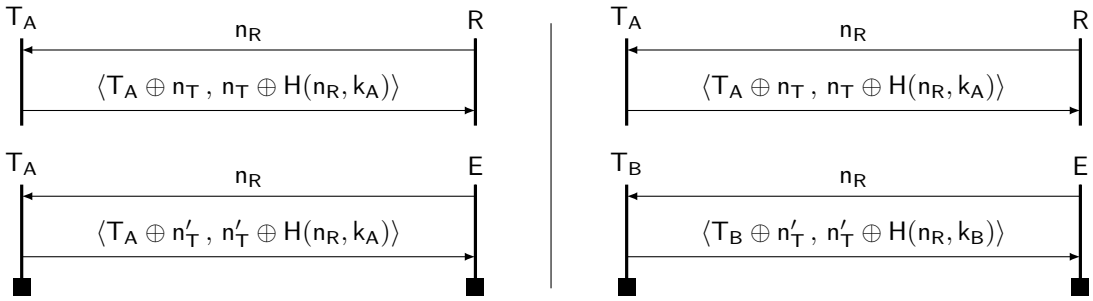
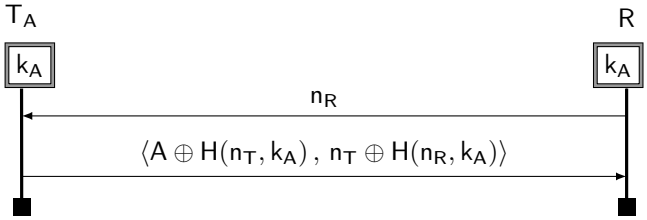


Figure 3.3: Attack against the original KCL protocol

3.2.2 KCL$^+$, a Revised Version of KCL

We propose a simple correction to the KCL protocol: we replace the first occurrence of the name $\mathsf{n_T}$ with its hash, breaking the algebraic property that was used in the attack. This protocol is depicted below. To our knowledge, there exists no formal study of this revised version.



We now illustrate our method by showing that the KCL$^+$ protocol verifies m-Privacy with two tags A and B. Assuming collision resistance only, there is actually an attack on the protocol KCL$^+$ (exactly the attack described later in Section 3.2.4). We therefore assume the PRF property.

To prove privacy of the KCL$^+$ protocol, we need some assumptions on the protocol primitives. We require that the pair, xor and boolean functions satisfy the axioms $\mathsf{Ax}_{\langle,\rangle}, \mathsf{Ax}_{\oplus}$ and $\mathsf{Ax}_{\mathsf{bool}}$ we gave in Subsection 2.5.2. Moreover, we need some assumption on the length of agent names and hashes: we require that agent names and hashes are of length η (the security parameter). Since names in $\mathcal{N}$ are always of length η in a computational model, we state that $\mathsf{len}(\mathsf{X}) = \mathsf{len}(\mathsf{n})$ and $\mathsf{len}(\mathsf{H}(x, y)) = \mathsf{len}(\mathsf{n})$ (for any agent X and $\mathsf{n} \in \mathcal{N}$).

Definition 3.2. We let $\mathsf{Ax}_{\mathsf{rfid}}$ be the union of $\mathsf{Ax}_{\langle,\rangle}, \mathsf{Ax}_{\oplus}, \mathsf{Ax}_{\mathsf{bool}}$ and, for any $\mathsf{n} \in \mathcal{N}$, the length axioms:

$$\frac{}{\mathsf{len}(\mathsf{X}) = \mathsf{len}(\mathsf{n})}\ \text{ID-len} \quad \text{where } \mathsf{X} \in \{\mathsf{A}, \mathsf{B}\} \qquad \frac{}{\mathsf{len}(\mathsf{H}(x, y)) = \mathsf{len}(\mathsf{n})}\ \text{H-len}$$

Theorem 3.2 (Unlinkability of KCL$^+$)**.** *For every m, the KCL$^+$ protocol verifies m-Fixed Trace Privacy for two tags for every $\mathsf{Ax}_{\mathsf{rfid}}$-interpretation $\mathcal{I}_c$ of $\mathcal{F}_p$ where H is interpreted as an PRF function.*

Proof. Using Theorem 3.1, it is sufficient to show that P satisfies m-Fixed Trace Privacy for two tags.

In this proof, we write $g(\phi)$ instead of $g_i(\phi)$ with $i = |\phi|$, where $g_i \in \mathcal{G}$. Moreover, the primed version of a term t is the term t, in which the names $\mathsf{n}_1, \ldots, \mathsf{n}_l$ appearing in t have been replaced by the primed names $\mathsf{n}'_1, \ldots, \mathsf{n}'_l$. We will use t_ϕ^{Id} (where $\mathsf{Id} = \mathsf{A}$ or B) to denote the response of the tag $\mathsf{T_{Id}}$ to a challenge:

$$t_\phi^{\mathsf{Id}} = \langle \mathsf{Id} \oplus \mathsf{H}(\mathsf{n_T}, \mathsf{k_{Id}}) , \mathsf{n_T} \oplus \mathsf{H}(g(\phi), \mathsf{k_{Id}}) \rangle$$

We prove this by induction on m. Let $\phi, \tilde{\phi}$ be two sequences of terms from the m-Fixed Trace Privacy definition, i.e.

$$\phi \equiv \phi_\tau^{\text{priv-lts}_{\mathbf{true}}(\mathrm{P})} \qquad \tilde{\phi} \equiv \phi_\tau^{\text{priv-lts}_{\mathbf{false}}(\mathrm{P})}$$

for some $(2, p, q)$-privacy trace τ (with $p + q = m$). By induction hypothesis, we assume that we have a derivation of $\phi \sim \tilde{\phi}$ (in the base case, this is the reflexivity of $\sim$). We have two cases.

If the adversary decides to start a new session with the reader, we need to show that $\phi, \mathsf{n_R} \sim \tilde{\phi}, \mathsf{n_R}$ where $\mathsf{n_R}$ is fresh in $\phi, \tilde{\phi}$. In that case, we apply the Fresh axiom and the induction hypothesis:

$$\frac{\phi \sim \tilde{\phi}}{\phi, \mathsf{n_R} \sim \tilde{\phi}, \mathsf{n_R}}\ \mathsf{Fresh}$$

Otherwise, the adversary decides to interact with the tags, e.g. A on the left and B on the right (the other cases are identical). In that case, we have to show that $\phi, t_\phi^{\mathsf{A}} \sim \tilde{\phi}, \tilde{t}_{\tilde{\phi}}^{\mathsf{B}}$ where:

$$t_\phi^{\mathsf{A}} \equiv \langle \mathsf{A} \oplus \mathsf{H}(\mathsf{n_T}, \mathsf{k_A}),\ \mathsf{n_T} \oplus \mathsf{H}(g(\phi), \mathsf{k_A}) \rangle \qquad \tilde{t}_{\tilde{\phi}}^{\mathsf{B}} \equiv \langle \mathsf{B} \oplus \mathsf{H}(\mathsf{n_T}, \mathsf{k_B}),\ \mathsf{n_T} \oplus \mathsf{H}(g(\tilde{\phi}), \mathsf{k_B}) \rangle$$

We let n be a fresh name and $\psi, \tilde{\psi}$ be the sequences of terms:

$$\psi \equiv \phi, \mathsf{n_T} \oplus \mathsf{H}(g(\phi), \mathsf{k_A}) \qquad \tilde{\psi} \equiv \tilde{\phi}, \mathsf{n_T} \oplus \mathsf{H}(g(\tilde{\phi}), \mathsf{k_B})$$

We start (from the root) our proof by applying the FA axiom (breaking the pair) and then to introduce an intermediate term $\mathsf{A} \oplus \mathsf{n}$ since, intuitively, $\mathsf{H}(\mathsf{n_T}, \mathsf{k_A})$ (resp. $\mathsf{H}(\mathsf{n_T}, \mathsf{k_B})$) should be indistinguishable from a random number.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\psi, \mathsf{H}(\mathsf{n_T}, \mathsf{k_A}) \sim \psi, \mathsf{n}}{\psi, \mathsf{A}, \mathsf{H}(\mathsf{n_T}, \mathsf{k_A}) \sim \psi, \mathsf{A}, \mathsf{n}}\ \mathsf{FA}}{\psi, \mathsf{A} \oplus \mathsf{H}(\mathsf{n_T}, \mathsf{k_A}) \sim \psi, \mathsf{A} \oplus \mathsf{n}}\ \mathsf{FA} \qquad P_1}{\psi, \mathsf{A} \oplus \mathsf{H}(\mathsf{n_T}, \mathsf{k_A}) \sim \tilde{\psi}, \mathsf{B} \oplus \mathsf{H}(\mathsf{n_T}, \mathsf{k_B})}\ \mathsf{Trans}}{\phi, t_\phi^A \sim \tilde{\phi}, \tilde{t}_{\tilde{\phi}}^B}\ \mathsf{FA}$$

where P_1 is a derivation of $\psi, \mathsf{A} \oplus \mathsf{n} \sim \tilde{\psi}, \mathsf{B} \oplus \mathsf{H}(\mathsf{n_T}, \mathsf{k_B})$.

Left Derivation We have to find first a a derivation of $\psi, \mathsf{H}(\mathsf{n_T}, \mathsf{k_A}) \sim \psi, \mathsf{n}$. The ultimate goal is to apply the PRF axioms. For that, we need to introduce, on both sides of the $\sim$ predicate, equality tests between the last message hashed under key $\mathsf{k_A}$ (i.e. $\mathsf{n_T}$), and all the previous hashed messages under key $\mathsf{k_A}$. We let $m_1, \ldots, m_l$ be the set of messages hashed with $\mathsf{k_A}$ in ϕ. We know that these messages are either names $\mathsf{n'_T}$, or of the form $g(\phi')$ where ϕ' is a strict prefix of ϕ.

Let $\alpha = \mathsf{H}(\mathsf{n_T}, \mathsf{k_A})$, $\beta = \mathsf{n}$. For all $1 \leq i \leq l$ we let $e_i \equiv \mathsf{eq}(\mathsf{n_T}, m_i)$, and s^x be the term:

$$\mathsf{if}\ e_1\ \mathsf{then}\ x\ \mathsf{else}\ \ldots\ \mathsf{if}\ e_l\ \mathsf{then}\ x\ \mathsf{else}\ x$$

We observe that, for every term u, $u = s^u$ is derivable from the equality axioms. We are now going to use the CS axiom to split the proof. To do so we introduce for every $1 \leq i \leq l$ the term u_i^x:

$$\mathsf{if}\ e_1\ \mathsf{then}\ 0\ \mathsf{else}\ \ldots \mathsf{if}\ e_{i-1}\ \mathsf{then}\ 0\ \mathsf{else}\ \mathsf{if}\ e_i\ \mathsf{then}\ x\ \mathsf{else}\ 0$$

And the term u_{l+1}^x:

$$\mathsf{if}\ e_1\ \mathsf{then}\ 0\ \mathsf{else}\ \ldots \mathsf{if}\ e_l\ \mathsf{then}\ 0\ \mathsf{else}\ x$$

By repeatedly applying the CS axiom we obtain:

$$\dfrac{\dfrac{\forall i \in \{1, \ldots, l+1\},\ \psi, e_1, \ldots, e_l, u_i^\alpha \sim \tilde{\psi}, e_1, \ldots, e_l, u_i^\beta}{\psi, s^{\mathsf{H}(\mathsf{n_T}, \mathsf{k_A})} \sim \psi, s^{\mathsf{n}}}\ \mathsf{CS}}{\psi, \mathsf{H}(\mathsf{n_T}, \mathsf{k_A}) \sim \psi, \mathsf{n}}\ \mathsf{Equ}$$

First note that, using the =-ind axiom, we derive, for every $1 \leq i \leq l$, $e_i = \mathsf{false}$. This allows us to deal with cases 1 to l, since this implies that $u_i^\alpha = u_i^\alpha = 0$ is derivable. Therefore we have for all $i \in \{1, \ldots, l\}$:

$$\dfrac{\dfrac{}{\psi, \mathsf{false}, \ldots, \mathsf{false}, 0 \sim \psi, \mathsf{false}, \ldots, \mathsf{false}, 0}\ \mathsf{Refl}}{\psi, e_1, \ldots, e_l, u_i^\alpha \sim \psi, e_1, \ldots, e_l, u_i^\beta}\ \mathsf{Equ}$$

Consider now the case $i = l + 1$. The conditions on the occurrences of H and $\mathsf{k_A}$ are satisfied, thanks to the choice of $e_1, \ldots, e_l$. Hence we can use the PRF axiom:

$$\dfrac{\dfrac{\dfrac{}{\psi, u_1^\alpha \sim \psi, u_1^\beta}\,\text{PRF}}{\psi, \mathsf{false}, \ldots, \mathsf{false}, u_{l+1}^\alpha \sim \psi, \mathsf{false}, \ldots, \mathsf{false}, u_{l+1}^\beta}\,\mathsf{FA}^l}{\psi, e_1, \ldots, e_l, u_1^\alpha \sim \psi, e_1, \ldots, e_l, u_1^\beta}\,\mathsf{Equ}$$

Right Derivation (P_1) Now, we have to derive $\psi, \mathsf{A} \oplus \mathsf{n} \sim \tilde{\psi}, \mathsf{B} \oplus \mathsf{H}(\mathsf{n_T}, \mathsf{k_B})$. We start by replacing A with B, splitting again the proof in two subgoals:

$$\dfrac{\psi, \mathsf{A} \oplus \mathsf{n} \sim \tilde{\psi}, \mathsf{B} \oplus \mathsf{n} \qquad \tilde{\psi}, \mathsf{B} \oplus \mathsf{n} \sim \tilde{\psi}, \mathsf{B} \oplus \mathsf{H}(\mathsf{n_T}, \mathsf{k_B})}{\psi, \mathsf{A} \oplus \mathsf{n} \sim \tilde{\psi}, \mathsf{B} \oplus \mathsf{H}(\mathsf{n_T}, \mathsf{k_B})}\,\mathsf{Trans}$$

For the right part, we first decompose the goal:

$$\dfrac{\dfrac{\dfrac{\tilde{\psi}, \mathsf{H}(\mathsf{n_T}, \mathsf{k_B}) \sim \tilde{\psi}, \mathsf{n}}{\tilde{\psi}, \mathsf{n} \sim \tilde{\psi}, \mathsf{H}(\mathsf{n_T}, \mathsf{k_B})}\,\mathsf{Sym}}{\tilde{\psi}, \mathsf{B}, \mathsf{n} \sim \tilde{\psi}, \mathsf{B}, \mathsf{H}(\mathsf{n_T}, \mathsf{k_B})}\,\mathsf{FA}}{\tilde{\psi}, \mathsf{B} \oplus \mathsf{n} \sim \tilde{\psi}, \mathsf{B} \oplus \mathsf{H}(\mathsf{n_T}, \mathsf{k_B})}\,\mathsf{FA}$$

Then, the derivation of $\tilde{\psi}, \mathsf{H}(\mathsf{n_T}, \mathsf{k_B}) \sim \tilde{\psi}, \mathsf{n}$ is similar to the derivation of $\psi, \mathsf{H}(\mathsf{n_T}, \mathsf{k_A}) \sim \psi, \mathsf{n}$.

For the left part, since n is fresh in ψ and $\tilde{\psi}$, we use the $\oplus$-ind axioms twice and the Fresh axiom:

$$\dfrac{\dfrac{\psi \sim \tilde{\psi}}{\psi, \mathsf{n} \sim \tilde{\psi}, \mathsf{n}}\,\mathsf{Fresh} \qquad \dfrac{}{\mathsf{len(A)} = \mathsf{len(n)}}\,\text{ID-len} \qquad \dfrac{}{\mathsf{len(B)} = \mathsf{len(n)}}\,\text{ID-len}}{\psi, \mathsf{A} \oplus \mathsf{n} \sim \tilde{\psi}, \mathsf{B} \oplus \mathsf{n}}\,\oplus\text{-ind}^2$$

It only remains to show that $\psi \sim \tilde{\psi}$. First, we split the proof in three sub-proofs using transitivity:

$$\overbrace{\phi, \mathsf{n_T} \oplus \mathsf{H}(g(\phi), \mathsf{k_A}) \sim \underbrace{\phi, \mathsf{n_T} \sim \tilde{\phi}, \mathsf{n_T}}_{\mathsf{MSim}}}^{\mathsf{LSim}} \overbrace{\sim \tilde{\phi}, \mathsf{n_T} \sim \tilde{\phi}, \mathsf{n_T} \oplus \mathsf{H}(g(\phi), \mathsf{k_B})}^{\mathsf{RSim}}$$

And we conclude using $\oplus$-ind and Fresh:

$$\dfrac{\mathsf{LSim} \qquad \dfrac{\phi \sim \tilde{\phi}}{\mathsf{MSim}}\,\mathsf{Fresh} \qquad \mathsf{RSim}}{\phi, \mathsf{n_T} \oplus \mathsf{H}(g(\phi), \mathsf{k_A}) \sim \tilde{\phi}, \mathsf{n_T} \oplus \mathsf{H}(g(\phi), \mathsf{k_B})}\,\mathsf{Trans}^2$$

$$\dfrac{\dfrac{\dfrac{}{\phi \sim \phi}\,\mathsf{Refl}}{\phi, \mathsf{n_T} \sim \phi, \mathsf{n_T}}\,\mathsf{Fresh} \qquad \dfrac{}{\mathsf{len}(\mathsf{H}(g(\phi), \mathsf{k_A})) = \mathsf{len}(\mathsf{n_T})}\,\mathsf{H\text{-}len}}{\mathsf{LSim}}\,\oplus\text{-ind} \qquad \dfrac{\dfrac{\dfrac{}{\tilde{\phi} \sim \tilde{\phi}}\,\mathsf{Refl}}{\tilde{\phi}, \mathsf{n_T} \sim \tilde{\phi}, \mathsf{n_T}}\,\mathsf{Fresh} \qquad \dfrac{}{\mathsf{len}(\mathsf{H}(g(\phi), \mathsf{k_B})) = \mathsf{len}(\mathsf{n_T})}\,\mathsf{H\text{-}len}}{\mathsf{RSim}}\,\oplus\text{-ind}$$

■

To keep the proof tractable, we considered only two tags. This means, in particular, that these tags cannot be corrupted tags. Nonetheless, our method is expressive enough for multiple tags, including corrupted ones, though we did not complete the proof in that case.

3.2.3 The LAK Protocol

The left part of Figure 3.4 describes the original protocol from [LAK06]. As mentioned before, this is a simplified version of the LAK protocol, without the key server. In the LAK protocol, the reader shares a private key $\mathsf{k_A}$ with each of its tags $\mathsf{T_A}$, and h is an hash function. This is a stateful protocol: the key is updated after each successful completion of the protocol, and the reader keeps in $\mathsf{k_A^0}$ the previous value of the key. This value is used as a backup in case $\mathsf{T_A}$ has not completed the protocol (for example because the last message was lost) and therefore not updated its version of the key. The protocol allows to recover from such a desynchronization: the reader R can use the previous version of $\mathsf{k_A}$ at the next

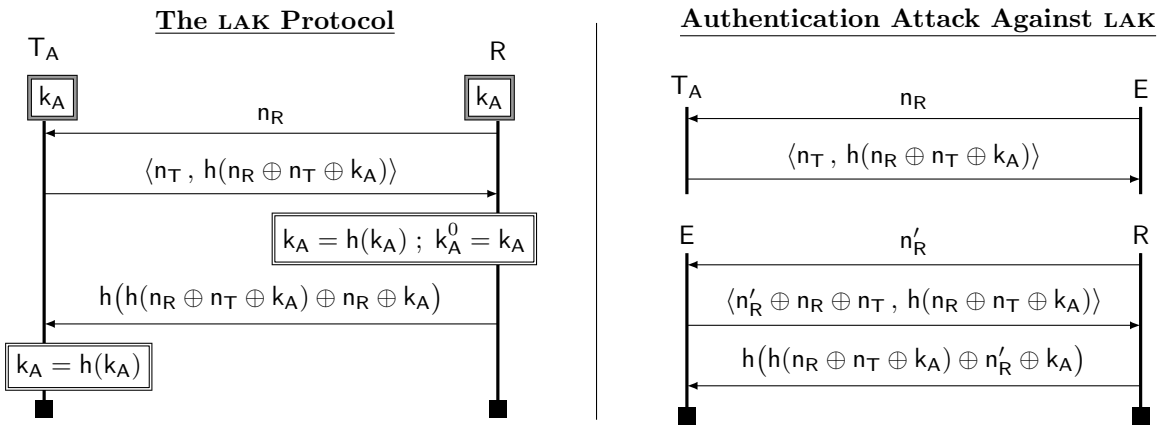


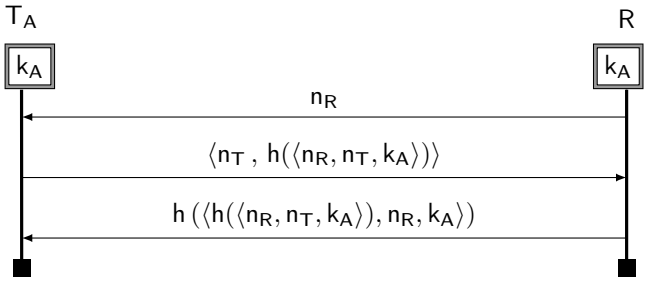
Figure 3.4: The LAK Protocol (Left) and a Known Authentication Attack Against LAK (Right).

round (which is the version used by $\mathsf{T_A}$) and finish the protocol. The protocol is supposed to achieve mutual authentication and unlinkability. Even though such properties can be defined in various ways, we recall below a known attack against the LAK protocol, which will force us to modify it.

An Attack on LAK An attack on authentication is described in [VDR08] and is depicted in the right part of Figure 3.4. In this attack, the adversary simply observes the beginning of an honest execution of the protocol (without completing the protocol, so that the reader and the tag do not update the key) between a tag A and the reader. The adversary obtains $\mathsf{h}(\mathsf{n_R} \oplus \mathsf{n_T} \oplus \mathsf{k_A})$ and the names $\mathsf{n_R}, \mathsf{n_T}$. He then interacts with the reader to get a new name $\mathsf{n'_R}$ and impersonates the tag A by choosing the returned tag $\mathsf{n'_T}$ such that $\mathsf{n'_R} \oplus \mathsf{n'_T} = \mathsf{n_R} \oplus \mathsf{n_T}$.

3.2.4 A Stateless Revised Version of LAK

In [HBD16], the authors consider a corrected (and stateless) version of the protocol, which they proved secure. This version of the protocol is described below:



This new version avoids the previous attack, which relied on the algebraic properties of exclusive-or. Formally, the protocol is described in the applied pi-calculus in [HBD16], in which they prove the strong unlinkability property of [ACRR10] in the Dolev-Yao model for an unbounded number of sessions.

Attack Against Stateless LAK Since the stateless version of LAK was proved in the symbolic model, no computational security assumptions were made on h. We show in Figure 3.5 that choosing h to be a one-way cryptographic hash function (OW-CPA and Strongly Collision Resistant for example) is not enough to guarantee unlinkability.

The attack is quite simple: it suffices that the hash function h leaks a few bits of the hashed message (which is possible for an one-way hash function). This means that, when hashing a message of the form $\langle \mathsf{n_R}, \mathsf{n_T}, \mathsf{k} \rangle$, the hash function h will leak some bits of the agent key k. Since the keys are drawn uniformly at random, there is a non negligible probability for the leaked bits to be different when hashing messages

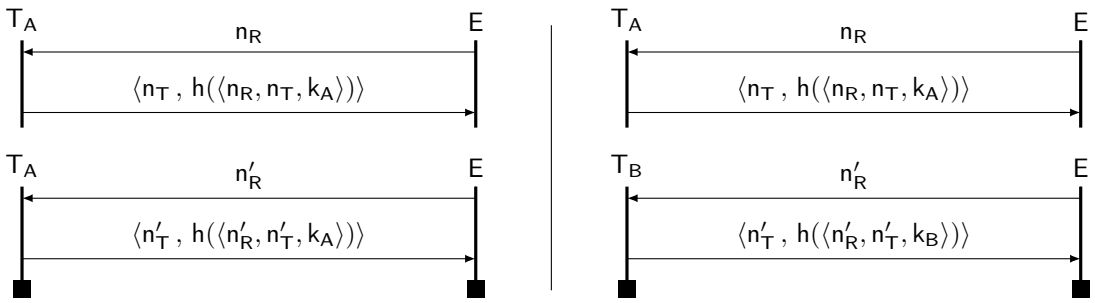


Figure 3.5: Unlinkability Attack in Two Rounds Against the Stateless LAK Protocol

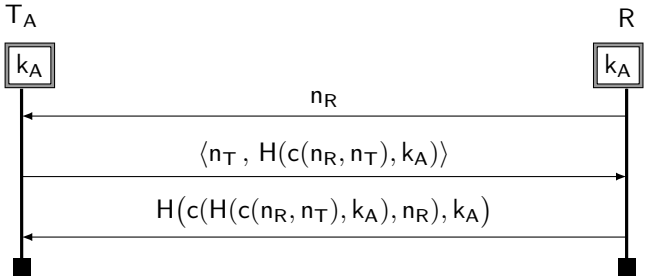


Figure 3.6: The LAK$^+$ Protocol

with different keys. In particular an adversary will be be able to distinguish $h(\langle n_R, n_T, k_A\rangle), h(\langle n'_R, n'_T, k_A\rangle)$ from $h(\langle n_R, n_T, k_A\rangle), h(\langle n'_R, n'_T, k_B\rangle)$ with high probability.

Observe that this attack would still work if we modified the protocol to update the keys after a successful execution of the protocol (in other word, if we consider the original LAK protocol with concatenation instead of xor), because the attacker could start executions of the protocol without finishing them, preventing the keys from being updated.

Remark 3.2. In the original paper introducing LAK [LAK06], the hash function is described as a one-way cryptographic hash function, which a priori does not prevent the attack described above. However, in the security analysis section, the authors assume the function to be indistinguishable from a random oracle, which prevents the attack. It is actually sufficient to assume PRF, for which there are effective constructions (subject to hardness assumptions). □

3.2.5 The LAK$^+$ Protocol

We describe here a stateless version of the LAK protocol, that we call LAK$^+$. The protocol is depicted in Figure 3.6. As in the LAK protocol, the reader shares with each tag a secret key k. We use a keyed-hash function that is assumed to be PRF to prevent the attack depicted in Section 3.2.4. This protocol uses a function c that combines the names. It could be a priori a xor, as in the original protocol, or a pairing, as in the revised version of [HBD16] or something else. We look for sufficient conditions on this function c, such that the protocol is secure.

We start by describing two different attacks that rely on some properties of the function c. In each case, we give a sufficient condition on c that prevents the attack. Next, we show that these two conditions are sufficient to prove that the LAK$^+$ protocol verifies the Bounded Session Privacy property.

First Attack: The attack depicted below is a generalization of the attack from [VDR08]. It works when there exists a function s (computable in probabilistic polynomial time) such that the quantity below is not negligible:

$$\mathbf{Pr}\big(n_R, n_T, n'_R : c(n_R, n_T) = c(n'_R, s(n_R, n_T, n'_R))\big) \tag{3.2}$$

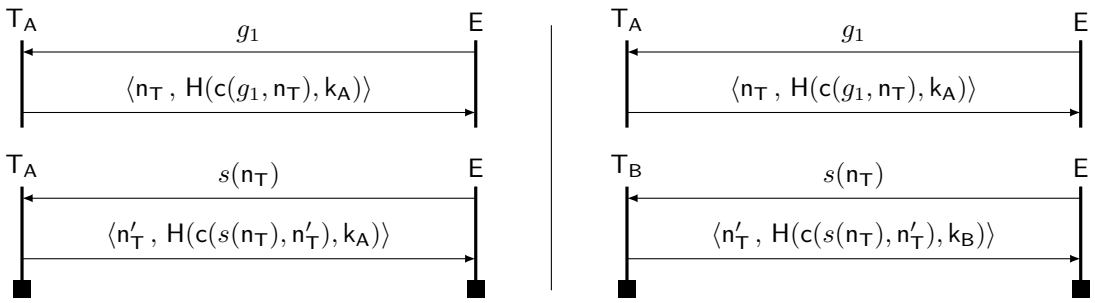
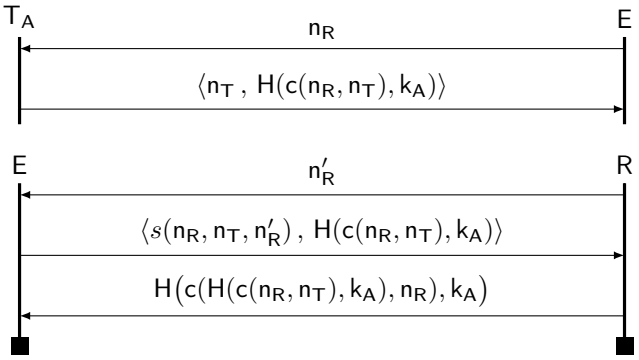


Figure 3.7: Unlinkability Attack Against LAK$^+$

This condition is satisfied if c is the xor operation (e.g. by taking $s(\mathsf{n_R}, \mathsf{n_T}, \mathsf{n'_R}) = \mathsf{n'_R} \oplus \mathsf{n_T} \oplus \mathsf{n'_R}$).



The attacker starts by sending a name $\mathsf{n_R}$ to the tag, and gets the name $\mathsf{n_T}$ chosen by the tag as well as the hash $\mathsf{H}(\mathsf{c}(\mathsf{n_R}, \mathsf{n_T}), \mathsf{k_A})$. Then the attacker initiates a second round of the protocol with the reader. The reader sends first a name $\mathsf{n'_R}$. The attacker is then able to answer, re-using the hash $\mathsf{H}(\mathsf{c}(\mathsf{n_R}, \mathsf{n_T}), \mathsf{k_A})$ sent by the tag in the first round, choosing $s(\mathsf{n_R}, \mathsf{n_T}, \mathsf{n'_R})$ as a replacement of the name $\mathsf{n'_T}$. Using Equation (3.2), there is a non negligible probability for the reader to accept the forged message as genuine.

This attack can be prevented by requiring c to be injective on its first argument:

$$\forall a, b, x, y.\ \mathsf{eq}(\mathsf{c}(a, b), \mathsf{c}(x, y)) \Rightarrow \mathsf{eq}(a, x)$$

Second Attack: We have an unlinkability attack if we can distinguish between the answers of the tags, even though the hash function is assumed to be a PRF. This is possible if there exists a constant g_1 and a function s such that:

$$\mathbf{Pr}\big(x, y : \mathsf{c}(g_1, x) = \mathsf{c}(s(x), y)\big) \text{ is not negligible} \tag{3.3}$$

If this is the case, then the unlinkability attack described in Figure 3.7 has a non negligible probability of success in distinguishing two consecutive rounds with the same tag A from one round with the tag A and one round with the tag B.

The attack works as follows: it starts by impersonating the reader, sends g_1 to the tag and gets the response $\langle \mathsf{n_T}, \mathsf{H}(\mathsf{c}(g_1, \mathsf{n_T}), \mathsf{k_A})\rangle$. Then the attacker initiates a new round of the protocol by sending $s(\mathsf{n_T})$ to the second tag. Using Equation 3.3, there is a non negligible probability that the hash in the response from the tag A in the second round of the protocol is the same as in the first round, whereas this will not be the case if the second round is initiated with B.

This attack can be prevented by asking c to be injective on its second argument:

$$\forall a, b, x, y.\ \mathsf{eq}(\mathsf{c}(a, b), \mathsf{c}(x, y)) \Rightarrow \mathsf{eq}(b, y)$$

$$\mathsf{if}\ \mathsf{eq}(u,u')\ \mathsf{then}\ \mathsf{false}\ \mathsf{else}\ \mathsf{eq}(\mathsf{c}(u,v),\mathsf{c}(u',v')) = \mathsf{false}$$
$$\mathsf{if}\ \mathsf{eq}(v,v')\ \mathsf{then}\ \mathsf{false}\ \mathsf{else}\ \mathsf{eq}(\mathsf{c}(u,v),\mathsf{c}(u',v')) = \mathsf{false}$$

Figure 3.8: Injectivity Axioms on the Combination Function c

$$\alpha \equiv \langle \mathsf{n}'_\mathsf{T}\,,\ \mathsf{H}(\mathsf{c}(g(\phi_2),\mathsf{n}'_\mathsf{T}),\mathsf{k}_\mathsf{A})\rangle$$
$$\beta \equiv \mathsf{H}\big(\mathsf{c}(\mathsf{n}'_\mathsf{R},\pi_1(g(\phi_3))),\mathsf{k}_\mathsf{A}\big)$$
$$\gamma \equiv \mathsf{H}\big(\mathsf{c}(\pi_2(g(\phi_3)),\mathsf{n}'_\mathsf{R}),\mathsf{k}_\mathsf{A}\big)$$

$\epsilon_1 \equiv \mathsf{n}'_\mathsf{R} \doteq g(\phi_0)$	$e_1 \equiv \mathsf{c}(\mathsf{n}'_\mathsf{R},\pi_1(g(\phi_3))) \doteq \mathsf{c}(g(\phi_0),\mathsf{n}_\mathsf{T})$	(in term $s^A_{\phi_0}$)
$\epsilon_2 \equiv \mathsf{n}'_\mathsf{R} \doteq \mathsf{n}_\mathsf{R}$	$e_2 \equiv \mathsf{c}(\mathsf{n}'_\mathsf{R},\pi_1(g(\phi_3))) \doteq \mathsf{c}(\mathsf{n}_\mathsf{R},\pi_1(g(\phi_1)))$	(in term $t^A_{\phi_1}$)
$\epsilon_3 \equiv \mathsf{n}'_\mathsf{R} \doteq \pi_2(g(\phi_1))$	$e_3 \equiv \mathsf{c}(\mathsf{n}'_\mathsf{R},\pi_1(g(\phi_3))) \doteq \mathsf{c}(\pi_2(g(\phi_1)),\mathsf{n}_\mathsf{R})$	(in term $t^A_{\phi_1}$)
$\epsilon_4 \equiv \mathsf{n}'_\mathsf{R} \doteq g(\phi_2)$ $\epsilon'_4 \equiv \pi_1(g(\phi_3)) \doteq \mathsf{n}'_\mathsf{T}$	$e_4 \equiv \mathsf{c}(\mathsf{n}'_\mathsf{R},\pi_1(g(\phi_3))) \doteq \mathsf{c}(g(\phi_2),\mathsf{n}'_\mathsf{T})$	(in term $s^A_{\phi_2}$)
$\epsilon_5 \equiv \mathsf{n}'_\mathsf{R} \doteq \pi_1(g(\phi_3))$ $\epsilon'_5 \equiv \mathsf{n}'_\mathsf{R} \doteq \pi_2(g(\phi_3))$	$e_5 \equiv \mathsf{c}(\mathsf{n}'_\mathsf{R},\pi_1(g(\phi_3))) \doteq \mathsf{c}(\pi_2(g(\phi_3)),\mathsf{n}'_\mathsf{R})$	(in term $t^A_{\phi_3}$)

Figure 3.9: Term Definitions for the LAK$^+$ Unlinkability Proof

Unlinkability of the LAK$^+$ Protocol To prevent all the attacks against LAK$^+$ described above, we are going to require c to be right and left injective. This can easily be expressed in the logic using the two axioms in Figure 3.8, which are satisfied, for instance, when c is a the pair function.

Three messages are sent in a complete session of the LAK$^+$ protocol: two by the reader and one by the tag. Therefore, if we want to show interesting properties of the LAK$^+$ protocol, we need to consider at least 6 terms in the trace (two full sessions, e.g. twice with the same tag $\mathsf{T_A}$ or with the tag $\mathsf{T_A}$ and the tag $\mathsf{T_B}$). This leads us to consider the 6-Fixed Trace Privacy of the LAK$^+$ protocol.

Theorem 3.3. *The* LAK$^+$ *protocol verifies* 6*-Privacy with two tags for every* $\mathsf{Ax_{rfid}}$*-interpretation* $\mathcal{I}_\mathsf{c}$ *of* $\mathcal{F}_p$ *where* H *is interpreted as a* PRF *function and where the injectivity axioms of Figure 3.8 are valid.*

In particular, the following formula is derivable:

$$n_R, s^A_{\phi_0}, t^A_{\phi_1}, n'_R, s'^A_{\phi_2}, t'^A_{\phi_3} \sim n_R, s^A_{\phi_0}, t^A_{\phi_1}, n'_R, s'^B_{\phi_2}, t'^B_{\tilde{\phi}_3}$$

where:

$$s^{Id}_\phi = \langle n_T\,,\ H(\mathsf{c}(g(\phi), n_T), k_{Id})\rangle$$
$$t^{Id}_\phi = \big[H\big(\mathsf{c}(n_R,\pi_1(g(\phi))),k_{Id}\big) \doteq \pi_2(g(\phi))\big]\ H\big(\mathsf{c}(\pi_2(g(\phi)),n_R),k_{Id}\big)$$
$$\phi_0 = n_R \qquad \phi_1 = n_R, s^A_{\phi_0} \qquad \phi_2 = n_R, s^A_{\phi_0}, t^A_{\phi_1}$$
$$\phi_3 = n_R, s^A_{\phi_0}, t^A_{\phi_1}, n'_R, s'^A_{\phi_2} \qquad \tilde{\phi}_3 = n_R, s^A_{\phi_0}, t^A_{\phi_1}, n'_R, s'^B_{\phi_2}$$

As with the KCL$^+$ protocol, by induction on m, it should be possible to generalize the result to an arbitrary m-Fixed Trace Privacy, although we did not do the proof.

Proof. Unsurprisingly, it turns out that left and right injectivity of c implies the injectivity of c. That is, the following formula is derivable using $\mathsf{Ax_{struct}}$, $\mathsf{Ax_{rfid}}$ and the structural axioms $\mathsf{Ax_{struct}}$:

$$\mathsf{eq}(\mathsf{c}(u,v),\mathsf{c}(u',v')) \leftrightarrow \mathsf{if}\ \mathsf{eq}(u,u')\ \mathsf{then}\ v \doteq v'\ \mathsf{else}\ \mathsf{false} \tag{3.4}$$

The proof is straightforward using left and right injectivity and the $\mathsf{if\_then\_else\_}$ axioms.

Most of the formulas are easy to prove, so we are going to focus on the formula explicitly given in the theorem statement, which is in our opinion the hardest case. Before starting, we define several new terms in Figure 3.9. We have similar definition for the tilded versions $\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}, \ldots$. We start by applying the FA axiom several times:

Proof Tree P_1:

$$\dfrac{\phi, u_1^\alpha, u_1^\beta, u_1^\gamma \sim \phi, u_1^\alpha, u_1^{\mathsf{n}}, u_1^\gamma \qquad \dfrac{\dfrac{\dfrac{\phi, u_1^\alpha, u_1^\gamma \sim \tilde{\phi}, \tilde{u}_1^{\tilde{\alpha}}, \tilde{u}_1^{\tilde{\gamma}}}{\phi, u_1^\alpha, \mathsf{n}, u_1^\gamma \sim \tilde{\phi}, \tilde{u}_1^{\tilde{\alpha}}, \mathsf{n}, \tilde{u}_1^{\tilde{\gamma}}}\,\mathsf{Fresh}}{\phi, u_1^\alpha, u_1^{\mathsf{n}}, u_1^\gamma \sim \tilde{\phi}, \tilde{u}_1^{\tilde{\alpha}}, \tilde{u}_1^{\mathsf{n}}, \tilde{u}_1^{\tilde{\gamma}}}\,\mathsf{FA}^*}}{\phi, u_1^\alpha, u_1^\beta, u_1^\gamma \sim \tilde{\phi}, \tilde{u}_1^{\tilde{\alpha}}, \tilde{u}_1^{\tilde{\beta}}, \tilde{u}_1^{\tilde{\gamma}}} \qquad \tilde{\phi}, \tilde{u}_1^{\tilde{\alpha}}, \tilde{u}_1^{\mathsf{n}}, \tilde{u}_1^{\tilde{\gamma}} \sim \tilde{\phi}, \tilde{u}_1^{\tilde{\alpha}}, \tilde{u}_1^{\tilde{\beta}}, \tilde{u}_1^{\tilde{\gamma}}}\,\mathsf{Trans}$$

Proof Tree P_2:

$$\dfrac{\phi, u_1^\alpha, u_1^\gamma \sim \phi, u_1^\alpha, u_1^{\mathsf{n}} \qquad \dfrac{\dfrac{\phi, u_1^\alpha \sim \tilde{\phi}, \tilde{u}_1^{\tilde{\alpha}}}{\phi, u_1^\alpha, \mathsf{n} \sim \tilde{\phi}, \tilde{u}_1^{\tilde{\alpha}}, \mathsf{n}}\,\mathsf{Fresh}}{\phi, u_1^\alpha, u_1^{\mathsf{n}} \sim \tilde{\phi}, \tilde{u}_1^{\tilde{\alpha}}, \tilde{u}_1^{\mathsf{n}}}\,\mathsf{FA}^* \qquad \tilde{\phi}, \tilde{u}_1^{\tilde{\alpha}}, \tilde{u}_1^{\mathsf{n}} \sim \tilde{\phi}, \tilde{u}_1^{\tilde{\alpha}}, \tilde{u}_1^{\tilde{\gamma}}}{\phi, u_1^\alpha, u_1^\gamma \sim \tilde{\phi}, \tilde{u}_1^{\tilde{\alpha}}, \tilde{u}_1^{\tilde{\gamma}}}\,\mathsf{Trans}$$

Figure 3.10: Derivations P_1 and P_2

$$\dfrac{\phi_2, \alpha, \beta, \gamma \sim \phi_2, \tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}}{\phi_3, t'^{A}_{\phi_3} \sim \tilde{\phi}_3, t'^{B}_{\tilde{\phi}_3}}\,\mathsf{FA}^*$$

We are now going to use the CS axiom on the conditional e_4, e_5 to split the proof. To do so we introduce the term:

$$u^x \equiv \mathsf{if}\ e_4\ \mathsf{then}\ \big(\mathsf{if}\ e_5\ \mathsf{then}\ x\ \mathsf{else}\ x\big)\ \mathsf{else}\ \big(\mathsf{if}\ e_5\ \mathsf{then}\ x\ \mathsf{else}\ x\big)$$

and the terms:

$$\begin{aligned}
u_1^x &\equiv \mathsf{if}\ e_4\ \mathsf{then}\ 0\ \mathsf{else}\ \big(\mathsf{if}\ e_5\ \mathsf{then}\ 0\ \mathsf{else}\ x\big)\\
u_2^x &\equiv \mathsf{if}\ e_4\ \mathsf{then}\ 0\ \mathsf{else}\ \big(\mathsf{if}\ e_5\ \mathsf{then}\ x\ \mathsf{else}\ 0\big)\\
u_3^x &\equiv \mathsf{if}\ e_4\ \mathsf{then}\ \big(\mathsf{if}\ e_5\ \mathsf{then}\ 0\ \mathsf{else}\ x\big)\ \mathsf{else}\ 0\\
u_4^x &\equiv \mathsf{if}\ e_4\ \mathsf{then}\ \big(\mathsf{if}\ e_5\ \mathsf{then}\ x\ \mathsf{else}\ 0\big)\ \mathsf{else}\ 0
\end{aligned}$$

Similarly we introduced the tilded versions of these terms. We observe that for all term s we have $s = u^s$ and $s = \tilde{u}^s$. Therefore we can apply the CS axiom, which gives us:

$$\dfrac{\forall i \in \{1, \ldots, 4\},\ \ \phi_2, e_4, e_5, u_i^\alpha, u_i^\beta, u_i^\gamma \sim \phi_2, \tilde{e}_4, \tilde{e}_5, \tilde{u}_i^{\tilde{\alpha}}, \tilde{u}_i^{\tilde{\beta}}, \tilde{u}_i^{\tilde{\gamma}}}{\phi_2, \alpha, \beta, \gamma \sim \phi_2, \tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}}\,\mathsf{CS}^*$$

We let $\phi = \phi_2, e_4, e_5$ and $\tilde{\phi} = \phi_2, \tilde{e}_4, \tilde{e}_5$.

Case $i = 1$ Let n be a fresh name, we start by the derivation P_1 displayed in Figure 3.10. Using =-ind we know that $\epsilon_1 = \epsilon_2 = \epsilon_3 = \mathsf{false}$, and using the left injectivity of c this shows that $e_1 = e_2 = e_3 = \mathsf{false}$. Therefore we know that:

$$u_1^\beta = v^\beta \equiv \mathsf{if}\ e_1\ \mathsf{then}\ 0\ \mathsf{else}\ \mathsf{if}\ e_2\ \mathsf{then}\ 0\ \mathsf{else}\ \big(\mathsf{if}\ e_3\ \mathsf{then}\ 0\ \mathsf{else}\ \big(u_1^\beta\big)\big)$$

$$u_1^{\mathsf{n}} = v^{\mathsf{n}} \equiv \mathsf{if}\ e_1\ \mathsf{then}\ 0\ \mathsf{else}\ \mathsf{if}\ e_2\ \mathsf{then}\ 0\ \mathsf{else}\ \big(\mathsf{if}\ e_3\ \mathsf{then}\ 0\ \mathsf{else}\ \big(u_1^{\mathsf{n}}\big)\big)$$

Hence we can apply the PRF axiom, which shows that:

$$\dfrac{\dfrac{}{\phi, u_1^\alpha, v^\beta, u_1^\gamma \sim \phi, u_1^\alpha, v^{\mathsf{n}}, u_1^\gamma}\,\mathsf{PRF}}{\phi, u_1^\alpha, u_1^\beta, u_1^\gamma \sim \phi, u_1^\alpha, u_1^{\mathsf{n}}, u_1^\gamma}\,\mathsf{Equ}$$

Similarly we show that:

$$\dfrac{\dfrac{}{\tilde{\phi}, \tilde{u}_1^{\tilde{\alpha}}, \tilde{v}^{\mathsf{n}}, \tilde{u}_1^{\tilde{\gamma}} \sim \tilde{\phi}, \tilde{u}_1^{\tilde{\alpha}}, \tilde{v}^{\tilde{\beta}}, \tilde{u}_1^{\tilde{\gamma}}}\,\mathsf{PRF}}{\tilde{\phi}, \tilde{u}_1^{\tilde{\alpha}}, \tilde{u}_1^{\mathsf{n}}, \tilde{u}_1^{\tilde{\gamma}} \sim \tilde{\phi}, \tilde{u}_1^{\tilde{\alpha}}, \tilde{u}_1^{\tilde{\beta}}, \tilde{u}_1^{\tilde{\gamma}}}\,\mathsf{Equ}$$

It remains to show that $\phi, u_1^\alpha, u_1^\gamma \sim \tilde{\phi}, \tilde{u}_1^{\tilde{\alpha}}, \tilde{u}_1^{\tilde{\gamma}}$. We do this exactly like we did to get rid of the u_1^β and $\tilde{u}_1^{\tilde{\beta}}$. First we use FA, Trans and Fresh to get the derivation P_2 displayed in Figure 3.10.

The formulas $\phi, u_1^\alpha, u_1^\gamma \sim \phi, u_1^\alpha, u_1^{\mathsf{n}}$ and $\tilde{\phi}, \tilde{u}_1^{\tilde{\alpha}}, \tilde{u}_1^{\mathsf{n}} \sim \tilde{\phi}, \tilde{u}_1^{\tilde{\alpha}}, \tilde{u}_1^{\tilde{\gamma}}$ are dealt with exactly like we did for $\phi, u_1^\alpha, u_1^\beta, u_1^\gamma \sim \phi, u_1^\alpha, u_1^{\mathsf{n}}, u_1^\gamma$, introducing the corresponding conditional tests. We do not detail these two cases, but notice that the right injectivity of c is needed for them.

We now need to show that $\phi, u_1^\alpha \sim \tilde{\phi}, \tilde{u}_1^{\tilde{\alpha}}$, which is done by applying the FA axiom several time:

$$\dfrac{\dfrac{\phi_2, \mathsf{n}'_\mathsf{R}, \mathsf{n}'_\mathsf{T}, \mathsf{H}(\mathsf{c}(g(\phi_2), \mathsf{n}'_\mathsf{T}), \mathsf{k}_\mathsf{A}) \sim \phi_2, \mathsf{n}'_\mathsf{R}, \mathsf{n}'_\mathsf{T}, \mathsf{H}(\mathsf{c}(g(\phi_2), \mathsf{n}'_\mathsf{T}), \mathsf{k}_\mathsf{B})}{\phi_3 \sim \tilde{\phi}_3}\,\mathsf{FA}^*}{\phi, u_1^\alpha \sim \tilde{\phi}, \tilde{u}_1^{\tilde{\alpha}}}\,\mathsf{FA}^*$$

Let $\psi \equiv \phi_2, \mathsf{n}'_\mathsf{R}, \mathsf{n}'_\mathsf{T}$, it is then easy to show that $\psi, \mathsf{H}(\mathsf{c}(g(\phi_2), \mathsf{n}'_\mathsf{T}), \mathsf{k}_\mathsf{A}) \sim \psi, \mathsf{H}(\mathsf{c}(g(\phi_2), \mathsf{n}'_\mathsf{T}), \mathsf{k}_\mathsf{A})$ is derivable using the fact that n'_T is fresh in ψ, the right injectivity of c and the PRF axiom.

Case $i = 2$ and 3 These case are very similar to the case $i = 1$, except that we need to use the Dup axiom at some point to get rid of the double occurrence of γ (in case $i = 2$) or α (in case $i = 3$).

Case $i = 4$ Using (3.4) we know that

$$e_4 = \mathsf{if}\ \epsilon_4\ \mathsf{then}\ \epsilon'_4\ \mathsf{else}\ \mathsf{false} \qquad\qquad e_5 = \mathsf{if}\ \epsilon_5\ \mathsf{then}\ \epsilon'_5\ \mathsf{else}\ \mathsf{false}$$

Since booleans $\epsilon'_4 \equiv \pi_1(g(\phi_3)) \doteq \mathsf{n}'_\mathsf{T}$ and $\epsilon_5 \equiv \mathsf{n}'_\mathsf{R} \doteq \pi_1(g(\phi_3))$ we have:

$$\begin{aligned}\mathsf{if}\ \epsilon'_4\ \mathsf{then}\ \epsilon_5\ \mathsf{else}\ \mathsf{false} &= \mathsf{if}\ \epsilon'_4\ \mathsf{then}\ \mathsf{n}'_\mathsf{R} \doteq \mathsf{n}'_\mathsf{T}\ \mathsf{else}\ \mathsf{false}\\ &= \mathsf{if}\ \epsilon'_4\ \mathsf{then}\ \mathsf{false}\ \mathsf{else}\ \mathsf{false}\\ &= \mathsf{false}\end{aligned}$$

And therefore, for all term v we have $u_4^v = 0$. Similarly we have $\tilde{u}_4^v = 0$ This means that we have:

$$\dfrac{\dfrac{\phi_3 \sim \tilde{\phi}_3}{\phi, 0, 0, 0 \sim \tilde{\phi}, 0, 0, 0}\,\mathsf{FA}}{\phi, u_4^\alpha, u_4^\beta, u_4^\gamma \sim \tilde{\phi}, u_4^{\tilde{\alpha}}, u_4^{\tilde{\beta}}, u_4^{\tilde{\gamma}}}\,\mathsf{Equ}$$

We already showed in the case $i = 1$ that $\phi_3 \sim \tilde{\phi}_3$ is derivable. ■

3.3 Pseudo-Random Number Generator

A PRNG uses an internal state, which is updated at each call, and outputs a pseudo random number. This can be modeled by a function G taking the internal state as input, and outputing a pair with the new internal state and the generated pseudo random number (retrieved using the projections π_S and π_o). Besides, a function init_S is used to initialized the internal state with a random seed (which can be hard-coded in the tag).

Definition 3.3. A PRNG is a tuple of polynomial functions $(G, \mathsf{init}_S, \pi_S, \pi_o)$ such that for every PPTM $\mathcal{A}$ and for every n, the following quantity is negligible in η:

$$|\mathbf{Pr}\left(r \in \{0,1\}^\eta\ :\ \mathcal{A}(\pi_o(s_0), \ldots, \pi_o(s_n)) = 1\right) - \mathbf{Pr}\left(r_0, \ldots, r_n \in \{0,1\}^\eta :\ \mathcal{A}(r_0, \ldots, r_n) = 1\right)|$$

where $s_0 = G(\mathsf{init}_S(r, 1^\eta))$ and for all $0 \leq i < n,\ s_{i+1} = G(\pi_S(s_i))$.

This can be translated in the logic using the PRNG axioms.

Definition 3.4. We let PRNG be the set of axioms:

$$\dfrac{}{\pi_o(s_0), \ldots, \pi_o(s_n) \sim \mathsf{n}_0, \ldots, \mathsf{n}_n}\,\mathsf{PRNG} \qquad \text{when} \begin{cases} s_0 \equiv G(\mathsf{init}_S(\mathsf{n})) \\ \forall 0 \leq i < n,\ s_{i+1} \equiv G(\pi_S(s_i)) \end{cases}$$

The soundness of these axioms is an immediate consequence of Definition 3.3.

Proposition 3.1. *The* PRNG *axioms are valid in any computational model* $\mathcal{M}_c$ *where* (G, init_S) *is interpreted as a* PRNG.

For each protocol where a strict separation exists between the cryptographic material used for random number generation and the other primitives (e.g. encryption keys), pseudo random numbers generated using a PRNG can be abstracted as random numbers.

Proposition 3.2. *For every names* $n, (n_i)_{i\leq n}$ *and contexts* $U_0, \ldots, U_n$ *that do not contain these names, the following formula is derivable using the structural axioms* $\mathsf{Ax}_{\mathsf{struct}}$ *and the* PRNG *axioms:*

$$U_0[\pi_o(s_0)], \ldots, U_n[\pi_o(s_n)] \sim U_0[n_0], \ldots, U_n[n_n]$$

where $s_0 \equiv G(\mathit{init}_S(n))$ *and* $\forall 0 \leq i < n,\ s_{i+1} \equiv G(\pi_S(s_i))$.

Proof. Let $\mathsf{n}, (\mathsf{n}_i)_{i\leq n}$ and $U_0, \ldots, U_n$ be such that $U_0, \ldots, U_n$ do not contain $\mathsf{n}, (\mathsf{n}_i)_{i\leq n}$. Let $s_0 \equiv G(\mathsf{init}_S(\mathsf{n}))$ and $\forall 0 \leq i < n,\ s_{i+1} \equiv G(\pi_S(s_i))$. We want to give a derivation of:

$$U_0[\pi_o(s_0)], \ldots, U_n[\pi_o(s_n)] \sim U_0[\mathsf{n}_0], \ldots, U_n[\mathsf{n}_n]$$

the structural axioms $\mathsf{Ax}_{\mathsf{struct}}$ and the PRNG axioms.

For all i, we let the context C_i and the names $(\mathsf{n}^p_{i,j})_j$ be such that $U_i \equiv C_i[(\mathsf{n}^p_{i,j})_j]$ and C_i does not contain any name (only function applications and holes). Then using the FA axiom we have:

$$\dfrac{\dfrac{\left((\mathsf{n}^p_{i,j})_j\right)_{i\leq n}, (\pi_o(s_i))_{i\leq n} \sim \left((\mathsf{n}^p_{i,j})_j\right)_{i\leq n}, (\mathsf{n}_i)_{i\leq n}}{\left((\mathsf{n}^p_{i,j})_j, \pi_o(s_i)\right)_{i\leq n} \sim \left((\mathsf{n}^p_{i,j})_j, \mathsf{n}_i\right)_{i\leq n}}\,\mathsf{Perm}}{\left(C_i[(\mathsf{n}^p_{i,j})_j][\pi_o(s_i)]\right)_{i\leq n} \sim \left(C_i[(\mathsf{n}^p_{i,j})_j][\mathsf{n}_i]\right)_{i\leq n}}\,\mathsf{FA}^*$$

Now, we can use the Dup axiom to get rid of multiple occurrences of the same name: indeed if there exists a name m such that $\mathsf{m} \equiv \mathsf{n}^p_{i,j}$ and $\mathsf{m} \equiv \mathsf{n}^p_{i',j'}$ then we can keep only one occurrence of m. Let $\mathsf{m}_1, \ldots, \mathsf{m}_l$ be such that for all $i \neq j, \mathsf{m}_i \neq \mathsf{m}_j$ and $\{\mathsf{m}_i \mid i \leq l\} = \{\mathsf{n}^p_{i,j} \mid i \leq n, j\}$, then:

$$\dfrac{(\mathsf{m}_i)_{i\leq n}, (\pi_o(s_i))_{i\leq n} \sim (\mathsf{m}_i)_{i\leq n}, (\mathsf{n}_i)_{i\leq n}}{\left((\mathsf{n}^p_{i,j})_j\right)_{i\leq n}, (\pi_o(s_i))_{i\leq n} \sim \left((\mathsf{n}^p_{i,j})_j\right)_{i\leq n}, (\mathsf{n}_i)_{i\leq n}}\,\mathsf{Dup}^*$$

Now by assumptions we know that $\{\mathsf{n}, (\mathsf{n}_i)_{i\leq n}\} \cap \{\mathsf{m}_i \mid i \leq l\} = \emptyset$, therefore we can apply the Fresh axiom for all $i \leq l$ to get rid of m_i. Finally we conclude with the PRNG axiom:

$$\dfrac{\dfrac{}{(\pi_o(s_i))_{i\leq n} \sim (\mathsf{n}_i)_{i\leq n}}\,\text{PRNG}}{(\mathsf{m}_i)_{i\leq n}, (\pi_o(s_i))_{i\leq n} \sim (\mathsf{m}_i)_{i\leq n}, (\mathsf{n}_i)_{i\leq n}}\,\mathsf{Fresh}^*$$

■

Remark 3.3 (Forward Secrecy)**.** We did not study forward secrecy of RFID protocols, but this could easily be done. The standard forward secrecy assumption on a PRNG states that leaking the internal state $\pi_S(s_n)$ of the PRNG (e.g. with a physical attack on the RFID chip) does not allow the adversary to gain any information about the previously generated names $(\pi_o(s_n))_{i\leq n}$. This could be expressed in the logic using, for example, the following axioms:

$$\pi_o(s_0), \ldots, \pi_o(s_n), \pi_S(s_n) \sim \mathsf{n}_0, \ldots, \mathsf{n}_n, \pi_S(s_n) \quad \text{when} \begin{cases} s_0 \equiv G(\mathsf{init}_S(\mathsf{n})) \\ \forall 0 \leq i < n,\ s_{i+1} \equiv G(\pi_S(s_i)) \end{cases}$$

□

3.4 Conclusion

We gave a framework for formally proving the security of RFID protocols in the computational model, by expressing Juels and Weis notion of Privacy in the Bana-Comon model, using our labelled transition system approach. We then illustrated this method on two examples, providing formal security proofs. We also showed that the security assumptions used in the proofs of these two protocols cannot be weakened (at least not in an obvious way).

Chapter 4

The 5G-AKA Authentication Protocol Privacy

The protocols and the privacy property studied in Chapter 3 are simple. Although the simplicity of this case study makes it a good first application of the Bana-Comon approach, it leaves us wondering how the method would fare on a more involved example. In this chapter, we remedy this problem, by studying a complex protocol and property. More precisely, we investigate the security of the 5G-AKA authentication protocol described in the 5G mobile communication standards.

5G-AKA is a new version of the AKA protocol, which tries to achieve a better privacy than the 3G and 4G versions, through the use of asymmetric randomized encryption. Nonetheless, we show that except for the IMSI-catcher attack, all known attacks against 5G-AKA privacy still apply. Therefore, we modify the 5G-AKA protocol to prevent these attacks, while satisfying 5G-AKA efficiency constraints as much as possible. Then, using the Bana-Comon indistinguishability logic, we formally prove that our protocol is σ-unlinkable. This is a new security notion, which allows for a fine-grained quantification of a protocol privacy. We also prove mutual authentication as a secondary result.

4.1 Introduction

Mobile communication technologies are widely used for voice, text and Internet access. These technologies allow a subscriber's device, typically a mobile phone, to connect wirelessly to an antenna, and from there to its service provider. The two most recent generations of mobile communication standards, the 3G and 4G standards, have been designed by the 3GPP consortium. The *fifth generation* (5G) of mobile communication standards is being finalized, and drafts are now available [TS318]. These standards describe protocols that aim at providing security guarantees to the subscribers and service providers. One of the most important such protocol is the *Authentication and Key Agreement* (AKA) protocol, which allows a subscriber and its service provider to establish a shared secret key in an authenticated fashion. There are different variants of the AKA protocol, one for each generation.

In the 3G and 4G-AKA protocols, the subscriber and its service provider share a long term secret key. The subscriber stores this key in a cryptographic chip, the *Universal Subscriber Identity Module* (*USIM*), which also performs all the cryptographic computations. Because of the *USIM* limited computational power, the protocols only use symmetric key cryptography without any pseudo-random number generation on the subscriber side. Therefore the subscriber does not use a random challenge to prevent replay attacks, but instead relies on a sequence number SQN. Since the sequence number has to be tracked by the subscriber and its service provider, the AKA protocols are stateful.

Because a user could be easily tracked through its mobile phone, it is important that the AKA protocols provide privacy guarantees. The 3G and 4G-AKA protocols try to do that using temporary identities. While this provides some privacy against a *passive adversary*, this is not enough against an *active adversary*. Indeed, these protocols allow an antenna to ask for a user permanent identity when it does not know its temporary identity (this naturally happens in roaming situations). This mechanism is abused by IMSI-catchers [Str07] to collect the permanent identities of all mobile devices in range.

The IMSI-catcher attack is not the only known attack against the privacy of the AKA protocols. In [BHP+17], the authors show how an attacker can obtain the least significant bits of a subscriber's

sequence number, which allows the attacker to monitor the user's activity. The authors of [AMR+12] describe a linkability attack against the 3G-AKA protocol. This attack is similar to the attack on the French e-passport [ACRR10], and relies on the fact that 3G-AKA protocol uses different error messages if the authentication failed because of a bad Mac or because a de-synchronization occurred.

The 5G standards include changes to the AKA protocol to improve its privacy guarantees. In 5G-AKA, a user never sends its permanent identity in plain-text. Instead, it encrypts it using a *randomized asymmetric encryption* with its service provider public key. While this prevents the IMSI-catcher attack, this is not sufficient to get unlinkability. Indeed, the attacks from [AMR+12, BHP+17] against the 3G and 4G-AKA protocols still apply. Moreover, the authors of [FOR16] proposed an attack against a variant of the AKA protocol introduced in [AMR+12], which uses the fact that an encrypted identity can be replayed. It turns out that their attack also applies to 5G-AKA.

Objectives Our goal is to improve the privacy of 5G-AKA while satisfying its design and efficiency constraints. In particular, our protocol should be as efficient as the 5G-AKA protocol, have a similar communication complexity and rely on the same cryptographic primitives. Moreover, we want formal guarantees on the privacy provided by our protocol.

Related Work There are several formal analysis of AKA protocols in the symbolic models. In [CKR18], the authors use the DEEPSEC tool to prove unlinkability of the protocol for three sessions. In [AMR+12] and [vdBVdR15], the authors use PROVERIF to prove unlinkability of AKA variants for, respectively, three sessions and an unbounded number of sessions. In these three works, the authors abstracted away several key features of the protocol. Because DEEPSEC and PROVERIF do not support the xor operator, they replaced it with a symmetric encryption. Moreover, sequence numbers are modeled by nonces in [AMR+12] and [CKR18]. While [vdBVdR15] models the sequence number update, they assume it is always incremented by one, which is incorrect. Finally, none of these works modeled the re-synchronization or the temporary identity mechanisms. Because of these inaccuracies in their models, they all miss attacks.

In [BDH+18], the authors use the TAMARIN prover to analyse multiple properties of 5G-AKA. For each property, they either find a proof, or exhibit an attack. To our knowledge, this is the most precise symbolic analysis of an AKA protocol. For example, they correctly model the xor and the re-synchronization mechanisms, and they represent sequence numbers as integers (which makes their model stateful). Still, they decided not to include the temporary identity mechanism. Using this model, they successfully rediscover the linkability attack from [AMR+12].

We are aware of two analysis of AKA protocols in the computational model. In [FOR16], the authors present a significantly modified version of AKA, called PRIV-AKA, and claim it is unlinkable. However, we discovered a linkability attack against the protocol, which falsifies the authors claim. In [LSWW14], the authors study the 4G-AKA protocol *without its first message*. They show that this reduced protocol satisfies a form of anonymity (which is weaker than unlinkability). Because they consider a weak privacy property for a reduced protocol, they fail to capture the linkability attacks from the literature.

Contributions This chapter contributions are:

- We study the privacy of the 5G-AKA protocol described in the 3GPP draft [TS318]. Thanks to the introduction of asymmetric encryption, the 5G version of AKA is not vulnerable to the IMSI-catcher attack. However, we show that the linkability attacks from [FOR16, AMR+12, BHP+17] against older versions of AKA still apply to 5G-AKA.
- We present a new linkability attack against PRIV-AKA, a significantly modified version of the AKA protocol introduced and claimed unlinkable in [FOR16]. This attack exploits the fact that, in PRIV-AKA, a message can be delayed to yield a state update later in the execution of the protocol, where it can be detected.
- We propose the AKA$^+$ protocol, which is a modified version of 5G-AKA with better privacy guarantees and satisfying the same design and efficiency constraints.
- We introduce a new privacy property, called σ-unlinkability, inspired from [HPVP11] and Vaudenay's Strong Privacy [Vau07]. Our property is parametric and allows us to have a fine-grained quantification of a protocol privacy.

- We formally prove that AKA$^+$ satisfies the σ-unlinkability property in the Bana-Comon model. Our proof is for any number of agents and sessions that are not related to the security parameter. We also show that AKA$^+$ provides mutual authentication.

Outline In Section 4.2 and 4.3 we describe the 5G-AKA protocol and the known linkability attacks against it. We present the AKA$^+$ protocol in Section 4.4, and we define the σ-unlinkability property in Section 4.5. We show how we model the AKA$^+$ protocol using the Bana-Comon logic in Section 4.6, and we describe the set of axioms we use in this chapter in Section 4.7. In Section 4.8, we state and sketch the proofs of the mutual authentication and σ-unlinkability of AKA$^+$. We prove mutual authentication in Section 4.9. In Section 4.10, we give some acceptance characterization conditions, which we use in Section 4.11 to prove that the AKA$^+$ protocol is σ-unlinkable. Finally, we conclude in Section 4.14.

Starred Proofs and Sections Several proofs and sections of this chapter are annotated by a star $\star$, followed by a page number. This indicates that they are technical, and that the rest of the chapter should be understandable without reading them. The page number corresponds to the page the technical proof or section ends.

4.2 The 5G-AKA Protocol

We present the 5G-AKA protocol described in the 3GPP standards [TS318]. This is a three-party authentication protocol between:

- The *User Equipment* (*UE*). This is the subscriber's physical device using the mobile communication network (e.g. a mobile phone). Each *UE* contains a cryptographic chip, the *Universal Subscriber Identity Module* (*USIM*), which stores the user confidential material (such as secret keys).
- The *Home Network* (*HN*), which is the subscriber's service provider. It maintains a database with the necessary data to authenticate its subscribers.
- The *Serving Network* (*SN*). It controls the base station (the antenna) the *UE* is communicating with through a wireless channel.

If the *HN* has a base station nearby the *UE*, then the *HN* and the *SN* are the same entity. But this is not always the case (e.g. in roaming situations). When no base station from the user's *HN* are in range, the *UE* uses another network's base station.

The *UE* and its corresponding *HN* share some confidential key material and the *Subscription Permanent Identifier* (SUPI), which uniquely identifies the *UE*. The *SN* does not have access to the secret key material. It follows that all cryptographic computations are performed by the *HN*, and sent to the *SN* through a secure channel. The *SN* also forwards all the information it gets from the *UE* to the *HN*. But the *UE* permanent identity is not kept hidden from the *SN*: after a successful authentication, the *HN* sends the SUPI to the *SN*. This is not technically needed, but is done for legal reasons. Indeed, the *SN* needs to know whom it is serving to be able to answer to *Lawful Interception* requests.

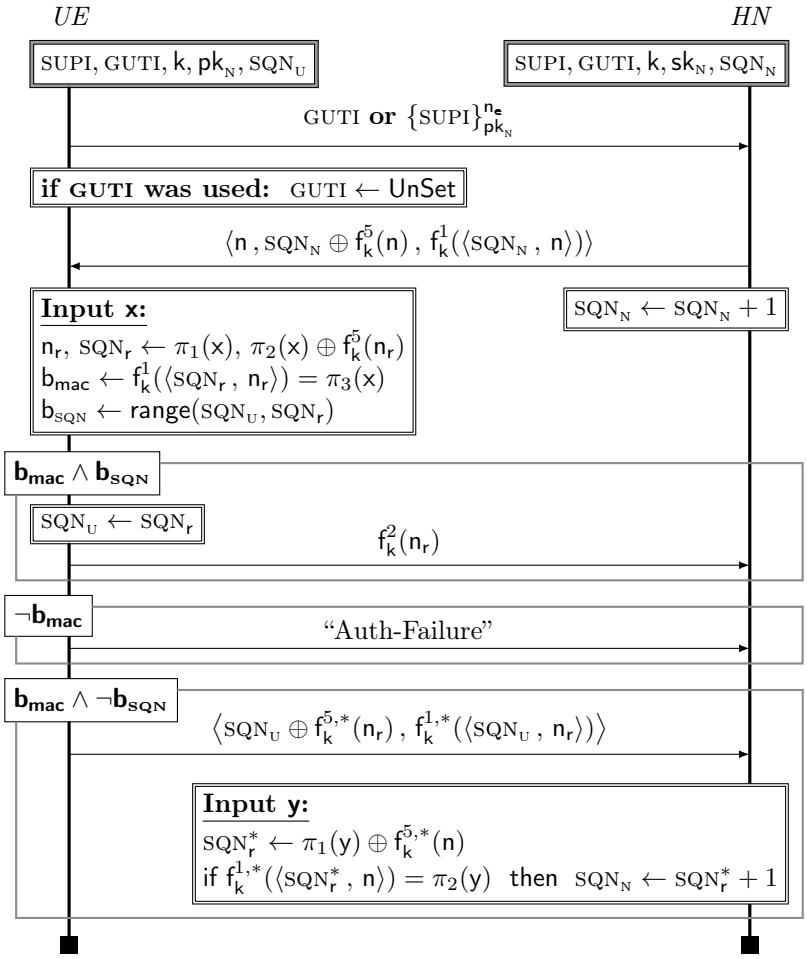
Therefore, privacy requires to trust both the *HN* and the *SN*. Since, in addition, they communicate through a secure channel, we decided to model them as a single entity and we include the *SN* inside the *HN*. A description of the protocol with three distinct parties can be found in [BDH$^+$18].

4.2.1 Description of the Protocol

The 5G standard proposes two authentication protocols, EAP-AKA$'$ and 5G-AKA. Since their differences are not relevant for privacy, we only describe the 5G-AKA protocol.

Cryptographic Primitives As in the 3G and 4G variants, the 5G-AKA protocol uses several keyed cryptographic one-way functions: f^1–f^5, $\mathsf{f}^{1,*}$ and $\mathsf{f}^{5,*}$. These functions are used both for integrity and confidentiality, and take as input a long term secret key k (which is different for each subscriber).

A major novelty in the 5G version of AKA is the introduction of an asymmetric randomized encryption $\{\cdot\}^{\mathsf{n_e}}_{\mathsf{pk}}$. Here pk is the public key, and $\mathsf{n_e}$ is the encryption randomness. Previous versions of AKA did not use asymmetric encryption because the *USIM*, which is a cryptographic micro-processor, had no randomness generation capabilities. The asymmetric encryption is used to conceal the identity of the *UE*, by sending $\{\text{SUPI}\}^{\mathsf{n_e}}_{\mathsf{pk}}$ instead of transmitting the SUPI in clear (as in 3G and 4G-AKA).



Conventions: ← denotes assignments, and has a lower priority than the equality comparison operator =.

Figure 4.1: The 5G-AKA Protocol

Temporary Identities After a successful run of the protocol, the *HN* may issue a temporary identity, a *Globally Unique Temporary Identifier* (GUTI), to the *UE*. Each GUTI can be used in *at most one session* to replace the encrypted identity $\{\text{SUPI}\}^{n_e}_{pk}$. It is renewed after each use. Using a GUTI allows to avoid computing the asymmetric encryption. This saves a pseudo-random number generation and the expensive computation of an asymmetric encryption.

Sequence Numbers The 5G-AKA protocol prevents replay attacks using a sequence number SQN instead of a random challenge. This sequence number is included in the messages, incremented after each successful run of the protocol, and must be tracked and updated by the *UE* and the *HN*. As it may get de-synchronized (e.g. because a message is lost), there are two versions of it: the *UE* sequence number SQN_U, and the *HN* sequence number SQN_N.

State The *UE* and *HN* share the *UE* identity SUPI, a long-term symmetric secret key k, a sequence number SQN_U and the *HN* public key pk_N. The *UE* also stores in GUTI the value of the last temporary identity assigned to it (if there is one). Finally, the *HN* stores the secret key sk_N corresponding to pk_N, its version SQN_N of every *UE*'s sequence number and a mapping between the GUTIs and the SUPIs.

Authentication Protocol The 5G-AKA protocol is represented in Figure 4.1. We now describe an honest execution of the protocol. First, the *UE* initiates the protocol by identifying itself to the *HN*, which it can do in two different ways:

- It can send a temporary identity GUTI, if one was assigned to it. After sending the GUTI, the *UE* sets it to UnSet to ensure that it will not be used more than once. Otherwise, it would allow an adversary to link sessions together.
- It can send its concealed permanent identity $\{\text{SUPI}\}^{\mathsf{n_e}}_{\mathsf{pk_N}}$, using the *HN* public key $\mathsf{pk_N}$ and a fresh randomness $\mathsf{n_e}$.

Upon reception of an identifying message, the *HN* retrieves the permanent identity SUPI: if it received a temporary identity GUTI, this is done through a database look-up; and if a concealed permanent identity was used, it uses $\mathsf{sk_N}$ to decrypt it. It can then recover SQN_N and the key k associated to the identity SUPI from its memory. The *HN* then generates a fresh nonce n. It masks the sequence number SQN_N by xoring it with $\mathsf{f}^5_\mathsf{k}(\mathsf{n})$, and mac the message by computing $\mathsf{f}^1_\mathsf{k}(\langle \text{SQN}_\text{N}, \mathsf{n}\rangle)$. It then sends the message $\langle \mathsf{n}, \text{SQN}_\text{N} \oplus \mathsf{f}^5_\mathsf{k}(\mathsf{n}), \mathsf{f}^1_\mathsf{k}(\langle \text{SQN}_\text{N}, \mathsf{n}\rangle)\rangle$.

When receiving this message, the *UE* computes $\mathsf{f}^5_\mathsf{k}(\mathsf{n})$. With it, it unmasks SQN_N and checks the authenticity of the message by re-computing $\mathsf{f}^1_\mathsf{k}(\langle \text{SQN}_\text{N}, \mathsf{n}\rangle)$ and verifying that it is equal to the third component of the message. It also checks whether SQN_N and SQN_U are in range[1]. If both checks succeed, the *UE* sets SQN_U to SQN_N, which prevents this message from being accepted again. It then sends $\mathsf{f}^2_\mathsf{k}(\mathsf{n})$ to prove to *HN* the knowledge of k. If the authenticity check fails, an "Auth-Failure" message is sent. Finally, if the authenticity check succeeds but the range check fails, *UE* starts the re-synchronization sub-protocol, which we describe below.

Re-synchronization The re-synchronization protocol allows the *HN* to obtain the current value of SQN_U. First, the *UE* masks SQN_U by xoring it with $\mathsf{f}^{5,*}_\mathsf{k}(\mathsf{n})$, mac the message using $\mathsf{f}^{1,*}_\mathsf{k}(\langle \text{SQN}_\text{U}, \mathsf{n}\rangle)$ and sends the pair $\langle \text{SQN}_\text{U} \oplus \mathsf{f}^{5,*}_\mathsf{k}(\mathsf{n}), \mathsf{f}^{1,*}_\mathsf{k}(\langle \text{SQN}_\text{U}, \mathsf{n}\rangle)\rangle$. When receiving this message, the *HN* unmasks SQN_U and checks the mac. If the authentication test is successful, *HN* sets the value of SQN_N to $\text{SQN}_\text{U} + 1$. This ensures that *HN* first message in the next session of the protocol is in the correct range.

GUTI Assignment There is a final component of the protocol which is not described in Figure 4.1 (as it is not used in the privacy attacks we present later). After a successful run of the protocol, the *HN* generates a new temporary identity GUTI and links it to the *UE*'s permanent identity in its database. Then, it sends the concealed fresh GUTI to the *UE*. The sub-protocol used to send a fresh GUTI is not used in the privacy attacks we present in the next session. Therefore, we omit its description.

4.3 Unlinkability Attacks Against 5G-AKA

We present in this section several attacks against AKA that appeared in the literature. All these attacks but one (the IMSI-catcher attack) carry over to 5G-AKA. Moreover, several fixes of the 3G and 4G versions of AKA have been proposed. We discuss the two most relevant fixes, the first by Arapinis et al. [AMR+12], and the second by Fouque et al. [FOR16].

None of these fixes are satisfactory. The modified AKA protocol given in [AMR+12] has been shown flawed in [FOR16]. The authors of [FOR16] then propose their own protocol, called PRIV-AKA, and claim it is unlinkable (they only provide a proof sketch). While analyzing the PRIV-AKA protocol, we discovered an attack allowing to permanently de-synchronize the *UE* and the *HN*. Since a de-synchronized *UE* can be easily tracked (after being de-synchronized, the *UE* rejects all further messages), our attack is also an unlinkability attack. This is in direct contradiction with the security property claimed in [FOR16]. This is a novel attack that never appeared in the literature.

4.3.1 IMSI-Catcher Attack

All the older versions of AKA (4G and earlier) are vulnerable to the IMSI-catcher attack [Str07]. This attack simply relies on the fact that, in these versions of AKA, the permanent identity (called the *International Mobile Subscriber Identity* or IMSI in the 4G specifications) is not encrypted but sent in plain-text. Moreover, even if a temporary identity is used (a *Temporary Mobile Subscriber Identity* or TMSI), an attacker can simply send a Permanent-ID-Request message to obtain the *UE*'s permanent identity. The attack is depicted in Figure 4.2.

[1] The specification is loose: it only requires that $\text{SQN}_\text{U} < \text{SQN}_\text{N} \leq \text{SQN}_\text{U} + C$, where C is some constant chosen by the *HN*.

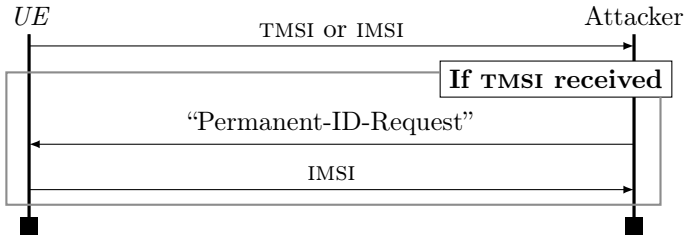


Figure 4.2: An IMSI-Catcher Attack

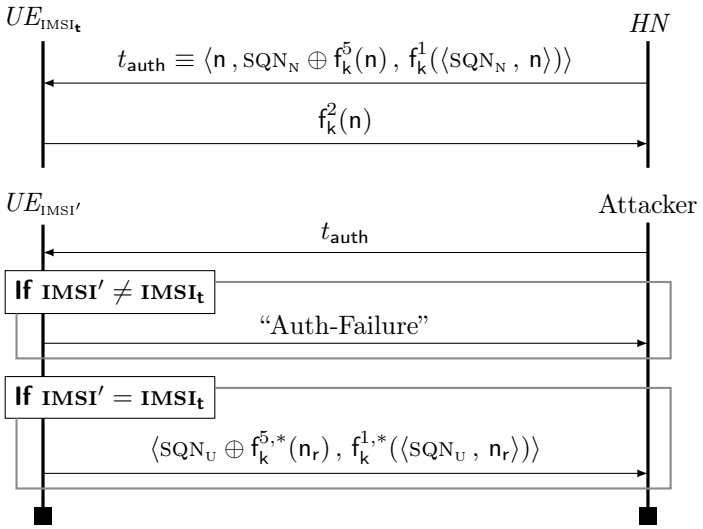


Figure 4.3: The Failure Message Attack by [AMR+12].

This necessitates an active attacker with its own base station. At the time, this required specialized hardware, and was believed to be too expensive. This is no longer the case, and can be done for a few hundreds dollars (see [SSB+16]).

4.3.2 The Failure Message Attack

In [AMR+12], Arapinis et al. propose to use an asymmetric encryption to protect against the IMSI-catcher attack: each *UE* carries the public-key of its corresponding *HN*, and uses it to encrypt its permanent identity. This is basically the solution that was adopted by 3GPP for the 5G version of AKA. Interestingly, they show that this is not enough to ensure privacy, and give a linkability attack that does not rely on the identification message sent by *UE*. While their attack is against the 3G-AKA protocol, it is applicable to the 5G-AKA protocol.

The Attack The attack is depicted in Figure 4.3, and works in two phases. First, the adversary eavesdrops a successful run of the protocol between the *HN* and the target *UE* with identity $IMSI_t$, and stores the authentication message t_{auth} sent by *HN*. In a second phase, the attacker $\mathcal{A}$ tries to determine whether a *UE* with identity $IMSI'$ is the initial *UE* (i.e. whether $IMSI' = IMSI_t$). To do this, $\mathcal{A}$ initiates a new session of the protocol and replays the message t_{auth}. If $IMSI' \neq IMSI_t$, then the mac test fails, and $UE_{IMSI'}$ answers "Auth-Failure". If $IMSI' = IMSI_t$, then the mac test succeeds but the range test fails, and $UE_{IMSI'}$ sends a re-synchronization message.

The adversary can distinguish between the two messages, and therefore knows if it is interacting with the original or a different *UE*. Moreover, the second phase of the attack can be repeated every time the adversary wants to check for the presence of the tracked user $IMSI_t$ in its vicinity.

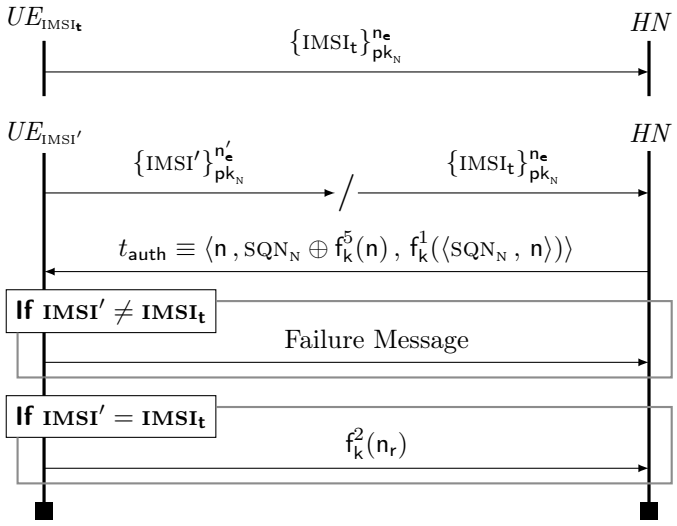


Figure 4.4: The Encrypted IMSI Replay Attack by [FOR16].

Proposed Fix To protect against the failure message attack, the authors of [AMR+12] propose that the *UE* encrypts both error message using the public key pk_N of the *HN*, making them indistinguishable. To the adversary, there is no distinctions between an authentication and a de-synchronization failure. The fixed AKA protocol, *without the identifying message* $\{\text{IMSI}\}^{n_e}_{pk_N}$, was formally checked in the symbolic model using the PROVERIF tool. Because this message was omitted in the model, an attack was missed. We present this attack in the next section.

4.3.3 The Encrypted IMSI Replay Attack

In [FOR16], Fouque et al. give an attack against the fixed AKA proposed by Arapinis et al. in [AMR+12]. Their attack, described in Figure 4.4, uses the fact the identifying message $\{\text{IMSI}_t\}^{n_e}_{pk_N}$ in the proposed AKA protocol by Arapinis et al. can be replayed.

In a first phase, the attacker $\mathcal{A}$ eavesdrops and stores the identifying message $\{\text{IMSI}_t\}^{n_e}_{pk_N}$ of an honest session between the user UE_{IMSI_t} it wants to track and the *HN*. Then, every time $\mathcal{A}$ wants to determine whether some user $UE_{\text{IMSI}'}$ is the tracked user UE_{IMSI_t}, it intercepts the identifying message $\{\text{IMSI}'\}^{n'_e}_{pk_N}$ sent by $UE_{\text{IMSI}'}$, and replaces it with the stored message $\{\text{IMSI}_t\}^{n_e}_{pk_N}$. Finally, $\mathcal{A}$ lets the protocol continue without further tampering. We have two possible outcomes:

- If $\text{IMSI}' \neq \text{IMSI}_t$ then the message t_{auth} sent by *HN* is mac-ed using the wrong key, and the *UE* rejects the message. Hence the attacker observes a failure message.
- If $\text{IMSI}' = \text{IMSI}_t$ then t_{auth} is accepted by $UE_{\text{IMSI}'}$, and the attacker observes a success message.

Therefore the attacker knows if it is interacting with $UE(\text{IMSI}_t)$ or not, which breaks unlinkability.

4.3.4 Attack Against The PRIV-AKA Protocol

The authors of [FOR16] then propose the PRIV-AKA protocol, which is a significantly modified version of AKA. The authors claim that their protocol achieves authentication and client unlinkability. But we discovered a de-synchronization attack: it is possible to permanently de-synchronize the *UE* and the *HN*. Our attack uses the fact that in PRIV-AKA, the *HN* sequence number is incremented only upon reception of the confirmation message from the *UE*. Therefore, by intercepting the last message from the *UE*, we can prevent the *HN* from incrementing its sequence number. We now describe the attack.

We run a session of the protocol, but we intercept the last message and store it for latter use. Note that the *HN*'s session is not closed. At that point, the *UE* and the *HN* are de-synchronized by one. We re-synchronize them by running a full session of the protocol. We then re-iterate the steps described above: we run a session of the protocol, prevent the last message from arriving at the *HN*, and then

run a full session of the protocol to re-synchronize the *HN* and the *UE*. Now the *UE* and the *HN* are synchronized, and we have two stored messages, one for each uncompleted session. We then send the two messages to the corresponding *HN* sessions, which accept them and increment the sequence number. In the end, it is incremented by *two*.

The problem is that the *UE* and the *HN* cannot recover from a de-synchronization by two. We believe that this was missed by the authors of [FOR16].[2] Remark that this attack is also an unlinkability attack. To attack some user UE_{IMSI}'s privacy, we permanently de-synchronize it. Then each time UE_{IMSI} tries to run the PRIV-AKA protocol, it will abort, which allows the adversary to track it.

Remark 4.1. Our attack requires that the *HN* does not close the first session when we execute the second session. At the end of the attack, before sending the two stored messages, there are two *HN* sessions simultaneously opened for the same *UE*. If the *HN* closes any un-finished sessions when starting a new session with the same *UE*, our attack does not work.

But this makes another unlinkability attack possible. Indeed, closing a session because of some later session between the *HN* and the same *UE* reveals a link between the two sessions. We describe the attack. First, we start a session i between a user UE_{A} and the *HN*, but we intercept and store the last message t_{A} from the user. Then, we let the *HN* run a full session with some user UE_{X}. Finally, we complete the initial session i by sending the stored message t_{A} to the *HN*. Here, we have two cases. If $\mathsf{X} = \mathsf{A}$, then the *HN* closed the first session when it completed the second. Hence it rejects t_{A}. If $\mathsf{X} \neq \mathsf{A}$, then the first session is still opened, and it accepts t_{A}.

Closing a session may leak information to the adversary. Protocols which aim at providing unlinkability must explicit when sessions can safely be closed. By default, we assume a session stays open. In a real implementation, a timeout *tied to the session* (and not the user identity) could be used to avoid keeping sessions opened forever. □

4.3.5 Sequence Numbers and Unlinkability

We conjecture that it is not possible to achieve functionality (i.e. honest sessions eventually succeed), authentication and unlinkability at the same time when using a sequence number based protocol with no random number generation capabilities in the *UE* side. We briefly explain our intuition.

In any sequence number based protocol, the agents may become de-synchronized because they cannot know if their last message has been received.[3] Furthermore, the attacker can cause de-synchronization by blocking messages. The problem is that we have contradictory requirements. On the one hand, to ensure authentication, an agent must reject a replayed message. On the other hand, in order to guarantee unlinkability, an honest agent has to behave the same way when receiving a message from a synchronized agent or from a de-synchronized agent. Since functionality requires that a message from a synchronized agent is accepted, it follows that a message from a de-synchronized agent must be accepted. Intuitively, it seems to us that an honest agent cannot distinguish between a protocol message which is being replayed and an honest protocol message from a de-synchronized agent. It follows that a replayed message should be both rejected and accepted, which is a contradiction.

This is only a conjecture. We do not have a formal statement, or a proof. Actually, it is unclear how to formally define the set of protocols that rely on sequence numbers to achieve authentication. Note however that all requirements can be satisfied simultaneously if we allow *both* parties to generate random challenges in each session (in AKA, only *HN* uses a random challenge). Examples of challenge based unlinkable authentication protocols can be found in [HBD16].

4.4 The AKA$^+$ Protocol

We now describe our principal contribution, which is the design of the AKA$^+$ protocol. This is a fixed version of the 5G-AKA protocol offering some form of privacy against an *active* attacker. First, we explicit the efficiency and design constraints. We then describe the AKA$^+$ protocol, and explain how we designed this protocol from 5G-AKA by fixing all the previously described attacks. As we mentioned before, we think unlinkability cannot be achieved under these constraints. Nonetheless, our protocol satisfies

[2] "the two sequence numbers may become desynchronized by one step [...]. Further desynchronization is prevented [...]" (p. 266 [FOR16])

[3] Indeed, in an asynchronous communication system one never knows if the last message has been received.

some weaker notion of unlinkability that we call σ-unlinkability. This is a new security property that we introduce. Finally, we will show a subtle attack, and explain how we fine-tuned AKA$^+$ to prevent it.

4.4.1 Efficiency and Design Constraints

We now explicit the protocol design constraints. These constraints are necessary for an efficient, inexpensive to implement and backward compatible protocol. Observe that, in a mobile setting, it is very important to avoid expensive computations as they quickly drain the *UE*'s battery.

Communication Complexity In 5G-AKA, authentication is achieved using only three messages: two messages are sent by the *UE*, and one by the *HN*. We want our protocol to have a similar communication complexity. While we did not manage to use only three messages in all scenarios, our protocol achieves authentication in less than four messages.

Cryptographic primitives We recall that all cryptographic primitives are computed in the *USIM*, where they are implemented in hardware. It follows that using more primitives in the *UE* would make the *USIM* more voluminous and expensive. Hence we restrict AKA$^+$ to the cryptographic primitives used in 5G-AKA: we use only symmetric keyed one-way functions and asymmetric encryption. Notice that the *USIM* cannot do asymmetric *decryption*. As in 5G-AKA, we use some in-expensive functions, e.g. xor, pairs, by-one increments and boolean tests. We believe that relying on the same cryptographic primitives helps ensuring backward compatibility, and would simplify the protocol deployment.

Random Number Generation In 5G-AKA, the *UE* generates at most one nonce per session, which is used to randomize the asymmetric encryption. Moreover, if the *UE* was assigned a GUTI in the previous session then there is no random number generation. Remark that when the *UE* and the *HN* are de-synchronized, the authentication fails and the *UE* sends a re-synchronization message. Since the session fails, no fresh GUTI is assigned to the *UE*. Hence, the next session of the protocol has to conceal the SUPI using $\{\text{SUPI}\}^{\mathsf{n_e}}_{\mathsf{pk}_\text{N}}$, which requires a random number generation. Therefore, we constrain our protocol to use at most one random number generation by the *UE* per session, and only if no GUTI has been assigned or if the *UE* and the *HN* have been de-synchronized.

Summary We summarize the constraints for AKA$^+$:

- It must use at most four messages per sessions.
- The *UE* may use only keyed one-way functions and asymmetric *encryption*. The *HN* may use these functions, plus asymmetric *decryption*.
- The *UE* may generate at most one random number per session, and only if no GUTI is available, or if re-synchronization with the *HN* is necessary.

4.4.2 Key Ideas

In this section, we present the two key ideas used in the design of the AKA$^+$ protocol.

Postponed Re-Synchronization Message We recall that whenever the *UE* and the *HN* are de-synchronized, the authentication fails and the *UE* sends a re-synchronization message. The problem is that this message can be distinguished from a Mac failure message, which allows the attack presented in Section 4.3.2. Since the session fails, no GUTI is assigned to the *UE*, and the next session will use the asymmetric encryption to conceal the SUPI. The first key idea is to piggy-back on the randomized encryption of the *next session* to send a concealed re-synchronization message. More precisely, we replace the message $\{\text{SUPI}\}^{\mathsf{n_e}}_{\mathsf{pk}_\text{N}}$ by $\{\langle \text{SUPI}, \text{SQN}_\text{U}\rangle\}^{\mathsf{n_e}}_{\mathsf{pk}_\text{N}}$. This has several advantages:

- We can remove the re-synchronization message that lead to the unlinkability attack presented in Section 4.3.2. In AKA$^+$, whenever the mac check or the range check fails, the same failure message is sent.
- This does not require more random number generation by the *UE*, since a random number is already being generated to conceal the SUPI in the next session.

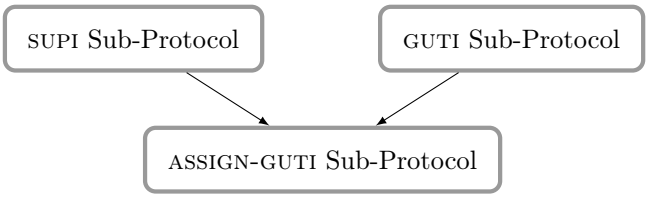


Figure 4.5: General Architecture of the AKA$^+$ Protocol

The 3GPP technical specification (see [TS318], Annex C) requires that the asymmetric encryption used in the 5G-AKA protocol is the ECIES encryption scheme, which is an hybrid encryption scheme. Hybrid encryption schemes use a randomized asymmetric encryption to conceal a temporary key. This key is then used to encrypt the message using a symmetric encryption, which is in-expensive. Hence encrypting the pair $\langle \text{SUPI}, \text{SQN}_U \rangle$ is almost as fast as encrypting only SUPI, and requires the *UE* to generate the same amount of randomness.

***HN* Challenge Before Identification** To prevent the Encrypted IMSI Replay Attack of Section 4.3.3, we add a random challenge n from the *HN*. The *UE* initiates the protocol by requesting a challenge without identifying itself. When requested, the *HN* generates and sends a fresh challenge n to the *UE*, which includes it in its response by mac-ing it with the SUPI using a symmetric one-way function Mac^1 with key $\mathsf{k}_{\mathsf{m}}^{\text{ID}}$. The *UE* response is now:

$$\left\langle \{\langle \text{SUPI}, \text{SQN}_U \rangle\}^{\mathsf{n_e}}_{\mathsf{pk}_N}, \mathsf{Mac}^1_{\mathsf{k}^{\text{ID}}_{\mathsf{m}}}(\langle \{\langle \text{SUPI}, \text{SQN}_U \rangle\}^{\mathsf{n_e}}_{\mathsf{pk}_N}, \mathsf{n} \rangle) \right\rangle$$

This challenge is only needed when the encrypted permanent identity is used. If the *UE* uses a temporary identity GUTI, then we do not need to use a random challenge. Indeed, temporary identities can only be used once before being discarded, and are therefore not subject to replay attacks. By consequence we split the protocol in two sub-protocols:

- The SUPI sub-protocol uses a random challenge from the *HN*, encrypts the permanent identity and allows to re-synchronize the *UE* and the *HN*.
- The GUTI sub-protocol is initiated by the *UE* using a temporary identity.

In the SUPI sub-protocol, the *UE*'s answer includes the challenge. We use this to save one message: the last confirmation step from the *UE* is not needed, and is removed. The resulting sub-protocol has four messages. Observe that the GUTI sub-protocol is faster, since it uses only three messages.

4.4.3 Architecture and States

Instead of a monolithic protocol, we have three sub-protocols: the SUPI and GUTI sub-protocols, which handle authentication; and the ASSIGN-GUTI sub-protocol, which is run after authentication has been achieved and assigns a fresh temporary identity to the *UE*. A full session of the AKA$^+$ protocol comprises a session of the SUPI or GUTI sub-protocols, followed by a session of the ASSIGN-GUTI sub-protocol. This is graphically depicted in Figure 4.5.

Since the GUTI sub-protocol uses only three messages and does not require the *UE* to generate a random number or compute an asymmetric encryption, it is faster than the SUPI sub-protocol. By consequence, the *UE* should always use the GUTI sub-protocol if it has a temporary identity available.

The *HN* runs concurrently an arbitrary number of sessions, but a subscriber cannot run more than one session at the same time. Of course, sessions from *different* subscribers may be concurrently running. We associate a unique integer, the session number, to every session, and we use $HN(j)$ and $UE_{\text{ID}}(j)$ to refer to the j-th session of, respectively, the *HN* and the *UE* with identity ID.

One-Way Functions We separate functions that are used only for confidentiality from functions that are also used for integrity. We have two confidentiality functions f and f^{r}, which use the key k, and five integrity functions Mac^1–Mac^5, which use the key $\mathsf{k_m}$. We require that f and f^{r} (resp. Mac^1–Mac^5) satisfy jointly the PRF assumption. This is a new assumption, which requires that these functions are *simultaneously* computationally indistinguishable from random functions.

Definition 4.1 (Jointly PRF Functions)**.** Let $H_1(\cdot,\cdot),\ldots,H_n(\cdot,\cdot)$ be a finite family of keyed hash functions from $\{0,1\}^* \times \{0,1\}^\eta$ to $\{0,1\}^\eta$. The functions $H_1,\ldots,H_n$ are *Jointly Pseudo Random Functions* if, for any PPTM adversary $\mathcal{A}$ with access to oracles $\mathcal{O}_{f_1},\ldots,\mathcal{O}_{f_n}$:

$$|\mathbf{Pr}(k:\ \mathcal{A}^{\mathcal{O}_{H_1(\cdot,k)},\ldots,\mathcal{O}_{H_n(\cdot,k)}}(1^\eta)=1) - \mathbf{Pr}(g_1,\ldots,g_n:\ \mathcal{A}^{\mathcal{O}_{g_1(\cdot)},\ldots,\mathcal{O}_{g_n(\cdot)}}(1^\eta)=1)|$$

is negligible, where:

- k is drawn uniformly in $\{0,1\}^\eta$.
- $g_1,\ldots,g_n$ are drawn uniformly in the set of all functions from $\{0,1\}^*$ to $\{0,1\}^\eta$.

Observe that if $H_1,\ldots,H_n$ are jointly PRF then, in particular, every individual H_i is a PRF.

Remark 4.2. While this is a non-usual assumption, it is simple to build a set of functions $H_1,\ldots,H_n$ which are jointly PRF from a single PRF H. First, let $(\mathsf{tag}_i(\cdot))_{1\leq i\leq n}$ be a set of tagging functions. We require that these functions are unambiguous, i.e. for all bit-strings u,v and $i\neq j$ we must have $\mathsf{tag}_i(u)\neq\mathsf{tag}_j(v)$. Then for every $1\leq i\leq n$, we let $H_i(x,y)=H(\mathsf{tag}_i(x),y)$. It is straightforward to show that if H is a PRF then $H_1,\ldots,H_n$ are jointly PRF. □

***UE* Persistent State** Each UE_{ID} with identity ID has a state $\mathsf{state}_{\text{U}}^{\text{ID}}$ persistent across sessions. It contains the following immutable values: the permanent identity SUPI = ID, the confidentiality key k^{ID}, the integrity key $\mathsf{k}_{\mathsf{m}}^{\text{ID}}$ and the *HN*'s public key pk_{N}. The states also contain mutable values: the sequence number SQN_{U}, the temporary identity GUTI_{U} and the boolean $\mathsf{valid\text{-}guti}_{\text{U}}$. We have $\mathsf{valid\text{-}guti}_{\text{U}}=\mathsf{false}$ whenever no valid temporary identity is assigned to the *UE*. Finally, there are mutable values that are not persistent across sessions. E.g. $\mathsf{b\text{-}auth}_{\text{U}}$ stores *HN*'s random challenge, and $\mathsf{e\text{-}auth}_{\text{U}}$ stores *HN*'s random challenge *when the authentication is successful.*

***HN* Persistent State** The *HN* state $\mathsf{state}_{\text{N}}$ contains the secret key sk_{N} corresponding to the public key pk_{N}. Also, for every subscriber with identity ID, it stores the keys k^{ID} and $\mathsf{k}_{\mathsf{m}}^{\text{ID}}$, the permanent identity SUPI = ID, the *HN* version of the sequence number $\text{SQN}_{\text{N}}^{\text{ID}}$ and the temporary identity $\text{GUTI}_{\text{N}}^{\text{ID}}$. It stores in $\mathsf{session}_{\text{N}}^{\text{ID}}$ the random challenge of the last session that was either a successful SUPI session which modified the sequence number, or a GUTI session which authenticated ID. This is used to detect and prevent some subtle attacks, which we present later. Finally, every session *HN*(j) stores in $\mathsf{b\text{-}auth}_{\text{N}}^{j}$ the identity claimed by the *UE*, and in $\mathsf{e\text{-}auth}_{\text{N}}^{j}$ the identity of the *UE* it authenticated.

4.4.4 The SUPI, GUTI and ASSIGN-GUTI Sub-Protocols

We describe honest executions of the three sub-protocols of the AKA$^+$ protocol. An honest execution is an execution where the adversary dutifully forwards the messages without tampering. Each execution is between a *UE* and *HN*(j).

The SUPI Sub-Protocol This protocol uses the *UE*'s permanent identity, re-synchronizes the *UE* and the *HN* and is expensive to run. The protocol is sketched in Figure 4.6.

The *UE* initiates the protocol by requesting a challenge from the network. When asked, *HN*(j) sends a fresh challenge n^j. After receiving n^j, the *UE* stores it in $\mathsf{b\text{-}auth}_{\text{U}}$, and answers with the encryption of its permanent identity together with the current value of its sequence number, using the *HN* public key pk_{N}. It also includes the mac of this encryption and of the challenge, which yields the message:

$$\left\langle \{\langle \text{SUPI}\,,\,\text{SQN}_{\text{U}}\rangle\}^{\mathsf{n_e}}_{\mathsf{pk}_{\text{N}}}\,,\,\mathsf{Mac}^1_{\mathsf{k}^{\text{ID}}_{\mathsf{m}}}(\langle \{\langle \text{SUPI}\,,\,\text{SQN}_{\text{U}}\rangle\}^{\mathsf{n_e}}_{\mathsf{pk}_{\text{N}}}\,,\,\mathsf{n}^j\rangle)\right\rangle$$

Then the *UE* increments its sequence number by one. When it gets this message, the *HN* retrieves the pair $\langle \text{SUPI}\,,\,\text{SQN}_{\text{U}}\rangle$ by decrypting the encryption using its secret key sk_{N}. For every identity ID, it checks if SUPI = ID and if the mac is correct. If this is the case, *HN* authenticated ID, and it stores ID in $\mathsf{b\text{-}auth}_{\text{N}}^{j}$ and $\mathsf{e\text{-}auth}_{\text{N}}^{j}$. After having authenticated ID, *HN* checks whether the sequence number SQN_{U} it received is greater than or equal to $\text{SQN}_{\text{N}}^{\text{ID}}$. If this holds, it sets $\text{SQN}_{\text{N}}^{\text{ID}}$ to $\text{SQN}_{\text{U}}+1$, stores n^j in $\mathsf{session}_{\text{N}}^{\text{ID}}$, generates a fresh temporary identity GUTI^j and stores it into $\text{GUTI}_{\text{N}}^{\text{ID}}$. This additional check ensures that the *HN* sequence number is always increasing, which is a crucial property of the protocol.

If the *HN* authenticated ID, it sends a confirmation message $\mathsf{Mac}^2_{\mathsf{k}^{\text{ID}}_{\mathsf{m}}}(\langle \mathsf{n}^j\,,\,\text{SQN}_{\text{U}}+1\rangle)$ to the *UE*. This message is sent even if the received sequence number SQN_{U} is smaller than $\text{SQN}_{\text{N}}^{\text{ID}}$. When receiving the

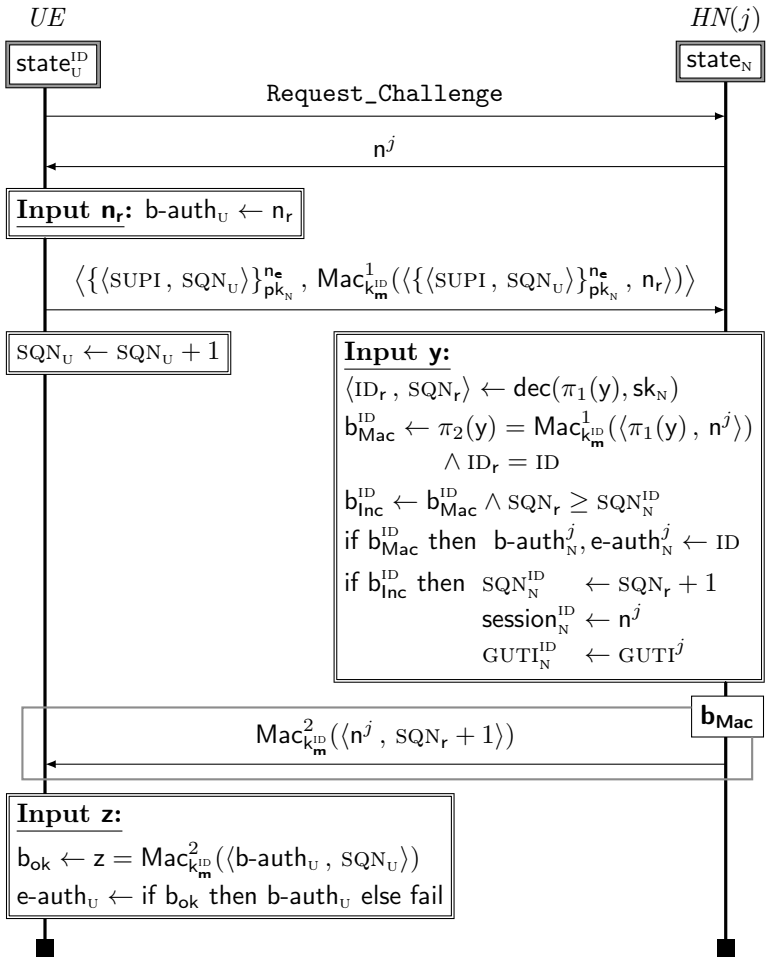


Figure 4.6: The SUPI Sub-Protocol of the AKA$^+$ Protocol

confirmation message, if the mac is valid then the *UE* authenticated the *HN*, and it stores in e-auth$_U$ the initial random challenge (which it keeps in b-auth$_U$). If the mac test fails, it stores in e-auth$_U$ the special value fail.

The GUTI Sub-Protocol This protocol uses the *UE*'s temporary identity, requires synchronization to succeed and is inexpensive. The protocol is sketched in Figure 4.7.

When valid-guti$_U$ is true, the *UE* can initiate the protocol by sending its temporary identity $GUTI_U$. The *UE* then sets valid-guti$_U$ to false to guarantee that this temporary identity is not used again. When receiving a temporary identity x, *HN* looks if there is an ID such that $GUTI^{ID}_N$ is equal to x and is not UnSet. If the temporary identity belongs to ID, it sets $GUTI^{ID}_N$ to UnSet and stores ID in b-auth^j_N. Then it generates a random challenge n^j, stores it in session$^{ID}_N$, and sends it to the *UE*, together with the xor of the sequence number SQN^{ID}_N with $f_{k^{ID}}(n^j)$, and a mac:

$$\langle n^j, SQN^{ID}_N \oplus f_{k^{ID}}(n^j), Mac^3_{k^{ID}_m}(\langle n^j, SQN^{ID}_N, GUTI^{ID}_N\rangle)\rangle$$

When it receives this message, the *UE* retrieves the challenge n^j at the beginning of the message, computes $f_{k^{ID}}(n^j)$ and uses this value to unconceal the sequence number SQN^{ID}_N. It then computes $Mac^3_{k^{ID}_m}(\langle n^j, SQN^{ID}_N, GUTI_U\rangle)$ and compares it to the mac received from the network. If the macs are not equal, or if the range check range(SQN_U, SQN^{ID}_N) fails, it puts fail into b-auth$_U$ and e-auth$_U$ to record that the authentication was not successful. If both tests succeed, it stores in b-auth$_U$ and e-auth$_U$ the random challenge, increments SQN_U by one and sends the confirmation message $Mac^4_{k^{ID}_m}(n^j)$. When receiving this message, the *HN* verifies that the mac is correct. If this is the case then the *HN* authenticated

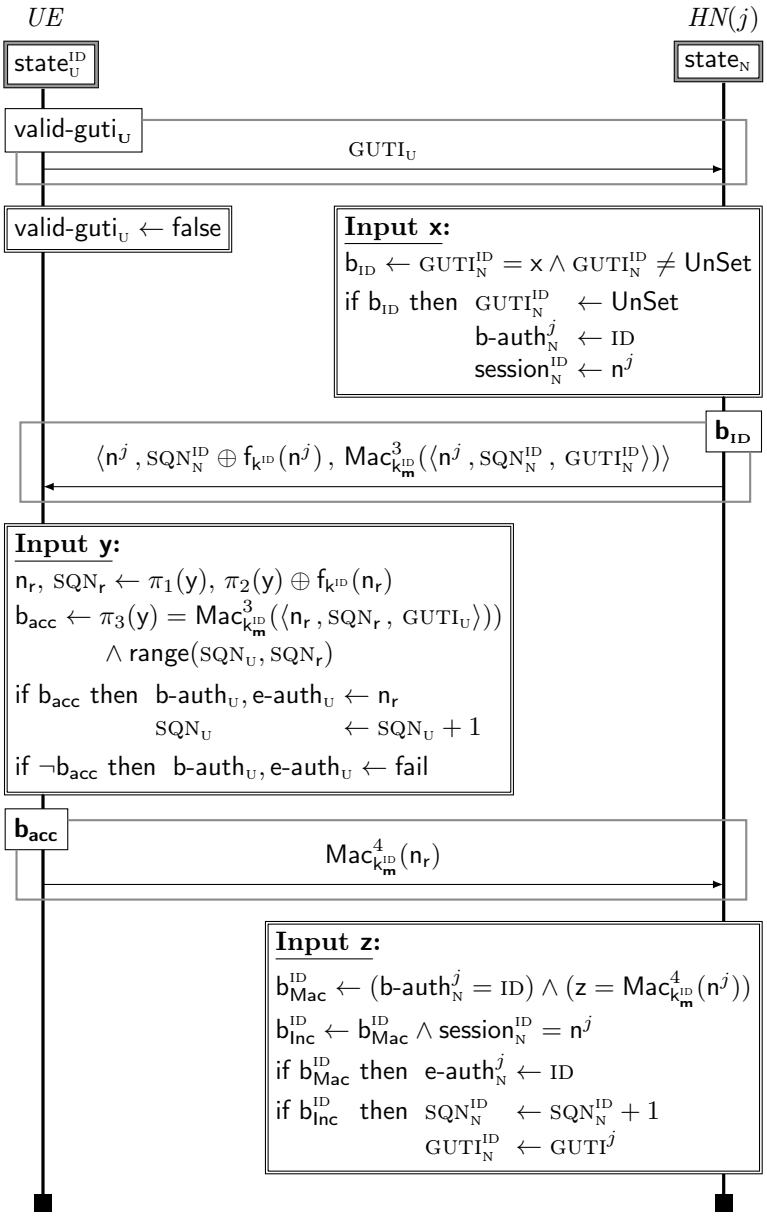


Figure 4.7: The GUTI Sub-Protocol of the AKA$^+$ Protocol

the *UE*, and stores ID into e-auth$_N^{ID}$. Then, *HN* checks whether session$_N^{ID}$ is still equal to the challenge n^j stored in it at the beginning of the session. If this is true, the *HN* increments SQN$_N^{ID}$ by one, generates a fresh temporary identity GUTI^j and stores it into GUTI$_N^{ID}$.

The ASSIGN-GUTI Sub-Protocol The ASSIGN-GUTI sub-protocol is run after a successful authentication, regardless of the authentication sub-protocol used. It assigns a fresh temporary identity to the *UE* to allow the next AKA$^+$ session to run the faster GUTI sub-protocol. It is depicted in Figure 4.8.

The *HN* conceals the temporary identity GUTI^j generated by the authentication sub-protocol by xoring it with f$^r_{k^{ID}}$(n^j), and macs it. When receiving this message, *UE* unconceals the temporary identity GUTI$_N^{ID}$ by xoring its first component with f$^r_{k^{ID}_m}$(e-auth$_U$) (since e-auth$_U$ contains the *HN*'s challenge after authentication). Then *UE* checks that the mac is correct and that the authentication was successful. If it is the case, it stores GUTI$_N^{ID}$ in GUTI$_U$ and sets valid-guti$_U$ to true.

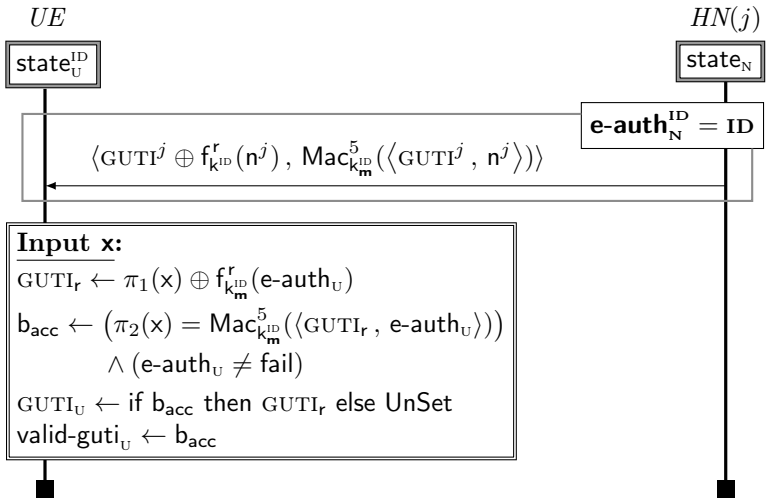


Figure 4.8: The ASSIGN-GUTI Sub-Protocol of the AKA$^+$ Protocol

4.5 Unlinkability

We now define the unlinkability property we use, which is inspired from [HPVP11] and Vaudenay's privacy [Vau07].

Definition The property is defined by a game in which an adversary tries to link together some subscriber's sessions. The adversary is a PPTM which interacts, through oracles, with N different subscribers with identities $\text{ID}_1, \ldots, \text{ID}_N$, and with the *HN*. The adversary cannot use a subscriber's permanent identity to refer to it, as it may not know it. Instead, we associate a virtual handler vh to any subscriber currently running a session of the protocol. We maintain a list l_{free} of all subscribers that are ready to start a session. We now describe the oracles $\mathcal{O}_b$:

- `StartSession`(): starts a new *HN* session and returns its session number j.
- `SendHN`(m, j) (resp. `SendUE`(m, vh)): sends the message m to $HN(j)$ (resp. the *UE* associated with vh), and returns $HN(j)$ (resp. vh) answer.
- `ResultHN`(j) (resp. `ResultUE`(vh)): returns true if $HN(j)$ (resp. the *UE* associated with vh) has made a successful authentication.
- `DrawUE`$(\text{ID}_{i_0}, \text{ID}_{i_1})$: checks that ID_{i_0} and ID_{i_1} are both in l_{free}. If that is the case, returns a new virtual handler pointing to ID_{i_b}, depending on an internal secret bit b. Then, it removes ID_{i_0} and ID_{i_1} from l_{free}.
- `FreeUE`(vh): makes the virtual handler vh no longer valid, and adds back to l_{free} the two identities that were removed when the virtual handler was created.

We recall that a function is negligible if and only if it is asymptotically smaller than the inverse of any polynomial. An adversary $\mathcal{A}$ interacting with $\mathcal{O}_b$ is winning the q-unlinkability game if: $\mathcal{A}$ makes less than q calls to the oracles; and it can guess the value of the internal bit b with a probability better than $1/2$ by a non-negligible margin, i.e. if the following quantity is non negligible in η:

$$\left|2 \times \mathbf{Pr}\left(b : \mathcal{A}^{\mathcal{O}_b}(1^\eta) = b\right) - 1\right|$$

Finally, a protocol is q-unlinkable if there are no winning adversaries against the q-unlinkability game.

Corruption In [HPVP11, Vau07], the adversary is allowed to corrupt some tags using a `Corrupt` oracle. Several classes of adversary are defined by restricting its access to the corruption oracle. A *strong* adversary has unrestricted access, a *destructive* adversary can no longer use a tag after corrupting it (it is destroyed), a *forward* adversary can only follow a `Corrupt` call by further `Corrupt` calls, and finally a

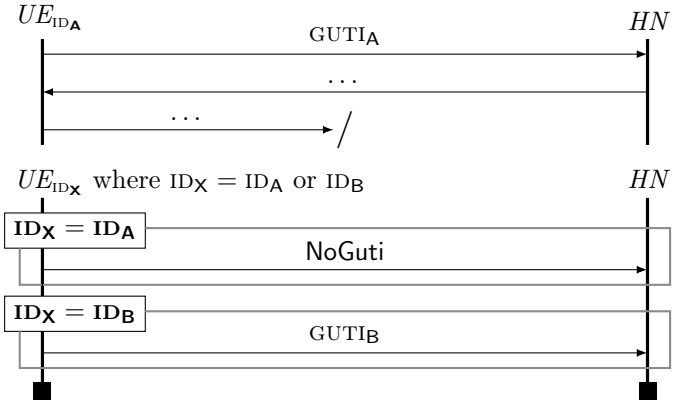


Figure 4.9: Consecutive GUTI Sessions of AKA$^+$ Are Not Unlinkable.

weak adversary cannot use `Corrupt` at all. A protocol is $\mathcal{C}$ unlinkable if no adversary in $\mathcal{C}$ can win the unlinkability game. Clearly, we have the following relations:

$$\textit{strong} \Rightarrow \textit{destructive} \Rightarrow \textit{forward} \Rightarrow \textit{weak}$$

The 5G-AKA protocol does not provide forward secrecy: indeed, obtaining the long-term secret of a *UE* allows to decrypt all its past messages. By consequence, the best we can hope for is *weak* unlinkability. Since such adversaries cannot call `Corrupt`, we removed the oracle from our definition.

Wide Adversary Remark that the adversary knows if the protocol was successful or not using the `ResultUE` and `ResultHN` oracles (such an adversary is called *wide* in Vaudenay's terminology [Vau07]). Indeed, in an authenticated key agreement protocol, this information is always available to the adversary: if the key exchange succeeds then it is followed by another protocol using the newly established key; while if it fails then either a new key-exchange session is initiated, or no message is sent. Hence the adversary knows if the key exchange was successful by passive monitoring.

4.5.1 σ-Unlinkability

In accord with our conjecture in Section 4.3.5, the AKA$^+$ protocol is not unlinkable. Indeed, an adversary $\mathcal{A}$ can easily win the linkability game. First, $\mathcal{A}$ ensures that $\mathrm{ID_A}$ and $\mathrm{ID_B}$ have a valid temporary identity assigned: $\mathcal{A}$ calls `DrawUE`$(\mathrm{ID_A}, \mathrm{ID_A})$ to obtain a virtual handler for $\mathrm{ID_A}$, and runs a SUPI and ASSIGN-GUTI sessions between $\mathrm{ID_A}$ and the *HN* with no interruptions. This assigns a temporary identity to $\mathrm{ID_A}$. We use the same procedure for $\mathrm{ID_B}$.

Then, $\mathcal{A}$ executes the attack described in Figure 4.9. It starts a GUTI session with $\mathrm{ID_A}$, and intercepts the last message. At that point, $\mathrm{ID_A}$ no longer has a temporary identity, while $\mathrm{ID_B}$ still does. Then, it calls `DrawUE`$(\mathrm{ID_A}, \mathrm{ID_B})$, which returns a virtual handler vh to $\mathrm{ID_A}$ or $\mathrm{ID_B}$. The attacker then starts a new GUTI session with vh. If vh is a handler for $\mathrm{ID_A}$, the *UE* returns NoGuti.[4] If vh aliases $\mathrm{ID_B}$, the *UE* returns the temporary identity GUTI$_\mathsf{A}$. The adversary $\mathcal{A}$ can distinguish between these two cases, and therefore wins the game.

σ-Unlinkability To prevent this, we want to forbid `DrawUE` to be called on de-synchronized subscribers. We do this by modifying the state of the user chosen by `DrawUE`. We let σ be an update on the state of the subscribers. We then define the oracle `DrawUE`$_\sigma(\mathrm{ID}_{i_0}, \mathrm{ID}_{i_1})$: it checks that $\mathrm{ID_A}$ and $\mathrm{ID_B}$ are both free, then *applies the update* σ to ID_{i_b}'s state, and returns a new virtual handler pointing to ID_{i_b}. The (q, σ)-unlinkability game is the q-unlinkability game in which we replace `DrawUE` with `DrawUE`$_\sigma$. A protocol is (q, σ)-unlinkable if and only if there is no winning adversary against the (q, σ)-unlinkability game. Finally, a protocol is σ-unlinkable if it is (q, σ)-unlinkable for any q.

[4] This is the special constant message the user sends whenever a temporary identity is asked from him. This message was omitted in the description of the protocol in Figure 4.7. We refer the reader to the next section for the complete formal modeling of the AKA$^+$ protocol.

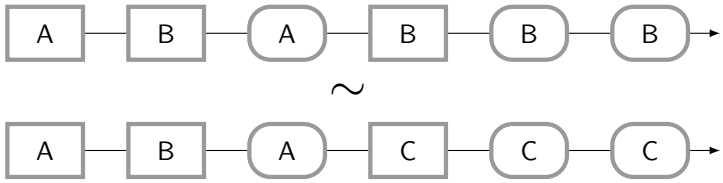


Figure 4.10: Two indistinguishable executions. Square (resp. round) nodes are executions of the SUPI (resp. GUTI) protocol. Each time the SUPI protocol is used, we can change the subscriber's identity.

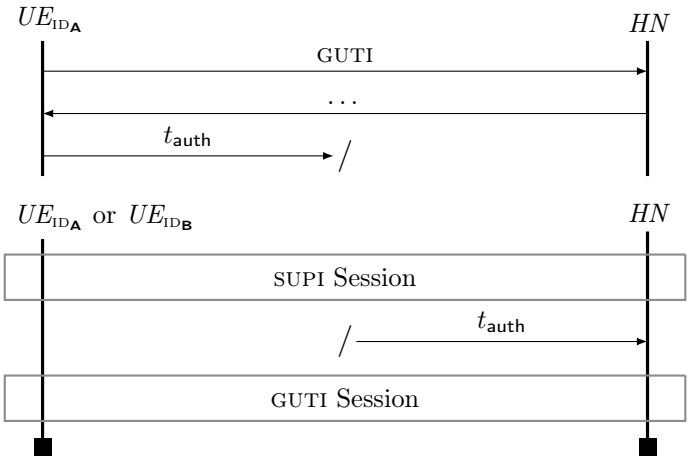


Figure 4.11: A Subtle Attack Against The $\mathrm{AKA}^{+}_{\mathsf{no\text{-}inc}}$ Protocol

Application to AKA$^+$ The privacy guarantees given by the σ-unlinkability property depend on the choice of σ. The idea is to choose a σ that allows to establish privacy in *some scenarios* of the standard unlinkability game.[5]

We illustrate this on the AKA$^+$ protocol. Let $\sigma_{\mathsf{ul}} = \mathsf{valid\text{-}guti}_{\mathrm{U}} \mapsto \mathsf{false}$ be the function that makes the *UE*'s temporary identity not valid. This simulates the fact that the GUTI has been used and is no longer available. If the *UE*'s temporary identity is not valid, then it can only run the SUPI sub-protocol. Hence, if the AKA$^+$ protocol is σ_{ul}-unlinkable, then no adversary can distinguish between a normal execution and an execution where we change the identity of a subscriber each time it runs the SUPI sub-protocol. We give in Figure 4.10 an example of such a scenario. We now state our main result:

Theorem 4.1. *The AKA$^+$ protocol is σ_{ul}-unlinkable for an arbitrary number of agents and sessions when the asymmetric encryption $\{\_\}_{\_}$ is* IND-CCA$_1$ *secure and f and f^r (resp. Mac^1–Mac^5) satisfy jointly the* PRF *assumption.*

This result is shown later in the chapter. The intuition is that no adversary can distinguish between two sessions of the SUPI protocol. Moreover, the SUPI protocol has two important properties. First, it re-synchronizes the user with the *HN*, which prevents the attacker from using any prior de-synchronization. Second, the AKA$^+$ protocol is designed in such a way that no message sent by the *UE* before a successful SUPI session can modify the *HN*'s state after the SUPI session. Therefore, any time the SUPI protocol is run, we get a "clean slate" and we can change the subscriber's identity. Note that we have a trade-off between efficiency and privacy: the SUPI protocol is more expensive to run, but provides more privacy.

4.5.2 A Subtle Attack

We now explain what is the role of $\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}$, and how it prevents a subtle attack against the σ_{ul}-unlinkability of AKA$^+$. We let $\mathrm{AKA}^{+}_{\mathsf{no\text{-}inc}}$ be the AKA$^+$ protocol where we modify the GUTI sub-protocol

[5] Remark that when σ is the empty state update, the σ-unlinkability and unlinkability properties coincide.

we described in Figure 4.7: in the state update of the *HN*'s last input, we remove the check $\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}} = \mathsf{n}^j$ (i.e. $\mathsf{b}_{\mathsf{Inc}}^{\mathrm{ID}} = \mathsf{b}_{\mathsf{Mac}}^{\mathrm{ID}}$). The attack is described in Figure 4.11.

First, we run a session of the GUTI sub-protocol between $UE_{\mathrm{ID}_\mathsf{A}}$ and the *HN*, but we do not forward the last message t_{auth} to the *HN*. We then call $\mathtt{DrawUE}_{\sigma_{\mathsf{ul}}}(\mathrm{ID}_\mathsf{A}, \mathrm{ID}_\mathsf{B})$, which returns a virtual handler vh to ID_A or ID_B. We run a full session using the SUPI sub-protocol with vh, and then send the message t_{auth} to the *HN*. We can check that, because we removed the condition $\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}} = \mathsf{n}^j$ from $\mathsf{b}_{\mathsf{Inc}}^{\mathrm{ID}}$, this message causes the *HN* to increment $\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}_\mathsf{A}}$ by one. At that point, $UE_{\mathrm{ID}_\mathsf{A}}$ is de-synchronized but $UE_{\mathrm{ID}_\mathsf{B}}$ is synchronized. Finally, we run a session of the GUTI sub-protocol. The session has two possible outcomes: if vh aliases to A then it fails, while if vh aliases to B, it succeeds. This leads to an attack.

When we removed the condition $\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}} = \mathsf{n}^j$, we broke the "clean slate" property of the SUPI sub-protocol: we can use a message from a session that started *before* the SUPI session to modify the state *after* the SUPI session. $\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}}$ allows to detect whether another session has been executed since the current session started, and to prevent the update of the sequence number when this is the case.

4.6 Modeling in The Bana-Comon Logic

We use the Bana-Comon logic to model the σ_{ul}-unlinkability of the AKA$^+$ protocol. To improve readability, protocol descriptions often omit some details: e.g., in Section 4.4, we sometimes omitted the description of the error messages. In other words, the AKA$^+$ protocol presented in Section 4.4 is *underspecified*. The failure message attack of [AMR+12] demonstrates that such details may be crucial for security. Therefore, before proving the AKA$^+$ protocol's security, we need to fully formalize it, and to make all assumptions explicit. We see two possible approaches:

- The first option, which we did *not* choose, consists in specifying the AKA$^+$ protocol in the *computational model*. In that case, the agents are interactive Turing machines, which need to be described, and the assumptions are properties of these machines. Since the σ_{ul}-unlinkability property is game-based, it directly applies to such a specification. Then, we translate the σ_{ul}-unlinkability of this protocol as the indistinguishability of two LTS. This require a tedious proof showing that the translation is sound. We also need to translate the assumptions into axioms of the logic. Finally, we have to prove that the two LTS are indistinguishable in the logic.
- The other option, which we opted for, consists in directly describing the protocol in the Bana-Comon logic, using a LTS. The assumptions on the protocol can be directly expressed in the logic using axioms. This is simpler than describing the protocol and the assumptions as interactive Turing machines and properties of these machines, and then translating them. Moreover, we do not need any soundness proof. The problem with this approach lies in the security property, as σ_{ul}-unlinkability is a game-based property. Even though it is straightforward to express directly the σ_{ul}-unlinkability of a protocol using LTSs,[6] we cannot establish a formal link between the game-based and LTS-based properties, since the game-based setting was never fully formalized. Instead, we informally argue that our formal LTS-based definition of σ_{ul}-unlinkability corresponds to the game-based definition of Section 4.5.1.

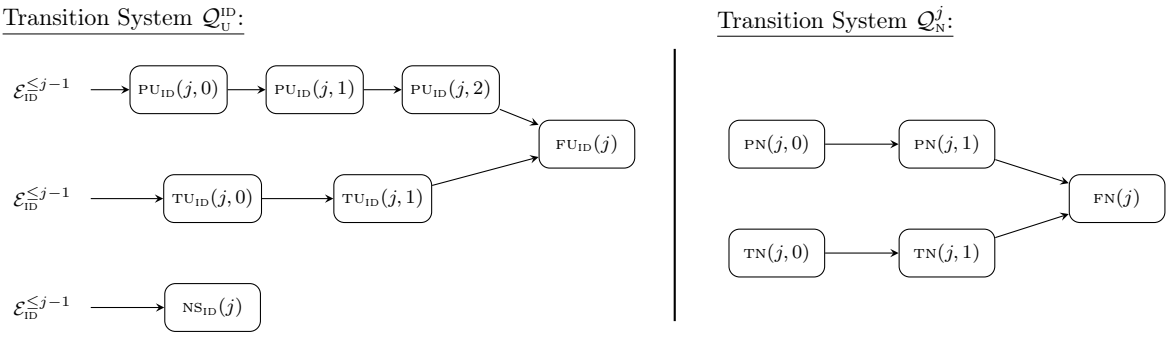
4.6.1 The AKA$^+$ Protocol Action Trace

We let $\mathcal{S}_{\mathsf{id}}^{\omega}$ be a countable set of zero-arity function symbols, which are used to represent identities. We are going to define the AKA^+_N protocol, which is the AKA$^+$ protocol on N identities $\mathcal{S}_{\mathsf{id}} = \{\mathrm{ID}_1, \ldots, \mathrm{ID}_N\}$. The full AKA$^+$ protocol can be obtained from the protocols $(\mathrm{AKA}^+_N)_{N\in\mathbb{N}}$ using a construction similar to the one in Section 3.1.2, with a initial phase selecting the number of agents.

Symbolic State For every identity $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$, we use several variables to represent UE_{ID}'s state. E.g. $\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}$ and $\mathrm{GUTI}_{\mathrm{U}}^{\mathrm{ID}}$ store, respectively, UE_{ID}'s sequence number and temporary identity. Similarly, we have variables for *HN*'s state, e.g. $\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}$. We let Vars_σ be the set of variables used in AKA^+_N:

$$\mathsf{Vars}_\sigma = \bigcup_{\substack{\mathrm{A}\in\{\mathrm{U},\mathrm{N}\}\\ j\in\mathbb{N},\mathrm{ID}\in\mathcal{S}_{\mathsf{id}}}} \left\{ \begin{array}{l} \mathrm{SQN}_{\mathrm{A}}^{\mathrm{ID}}, \mathrm{GUTI}_{\mathrm{A}}^{\mathrm{ID}}, \mathsf{e\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}, \mathsf{b\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}, \mathsf{e\text{-}auth}_{\mathrm{N}}^{j} \\ \mathsf{b\text{-}auth}_{\mathrm{N}}^{j}, \mathsf{s\text{-}valid\text{-}guti}_{\mathrm{U}}^{\mathrm{ID}}, \mathsf{valid\text{-}guti}_{\mathrm{U}}^{\mathrm{ID}}, \mathsf{session}_{\mathrm{N}}^{\mathrm{ID}} \end{array} \right\}$$

[6] As σ_{ul}-unlinkability requires that the executions of specific scenarios of the protocol are indistinguishable.



Convention: where $\mathcal{E}_{\text{ID}}^{\leq j} = \{\text{PU}_{\text{ID}}(j_0,i), \text{TU}_{\text{ID}}(j_0,i), \text{FU}_{\text{ID}}(j_0), \text{NS}_{\text{ID}}(j_0) \mid j_0 \leq j\}$, the initial states of $\mathcal{Q}_{\text{U}}^{\text{ID}}$ are $\text{PU}_{\text{ID}}(0,0)$ and $\text{TU}_{\text{ID}}(0,0)$, and the initial states of $\mathcal{Q}_{\text{N}}^{j}$ are $\text{PN}(j,0)$ and $\text{TN}(j,0)$. Every state of $\mathcal{Q}_{\text{U}}^{\text{ID}}$ and $\mathcal{Q}_{\text{N}}^{j}$ is final.

Figure 4.12: The Transition Systems Used to Define Valid Action Traces.

We recall that a symbolic state σ is a mapping from Vars_σ to terms. Intuitively, $\sigma(\mathsf{x})$ is a term representing (the distribution of) the value of x.

Example 4.1. We can express the fact that $\text{GUTI}_{\text{U}}^{\text{ID}}$ is unset in a symbolic state σ by having $\sigma(\text{GUTI}_{\text{U}}^{\text{ID}}) \equiv \mathsf{UnSet}$. Also, given a state σ, we can state that σ' is the state σ in which we incremented $\text{SQN}_{\text{U}}^{\text{ID}}$ by having $\sigma'(\mathsf{x})$ be the term $\sigma(\text{SQN}_{\text{U}}^{\text{ID}}) + 1$ if x is $\text{SQN}_{\text{U}}^{\text{ID}}$, and $\sigma(\mathsf{x})$ otherwise. □

Action Labels In the $(q, \sigma_{\mathsf{ul}})$-unlinkability game, the adversary chooses dynamically which oracle it wants to call. This is not convenient to use in proofs, as we do not know statically the i-th action of the adversary. We prefer an alternative point-of-view, in which the trace of oracle calls is fixed. Then, there are no winning adversaries against the σ_{ul}-unlinkability game with a fixed trace of oracle calls if the adversary's interactions with the oracles when $b = 0$ are indistinguishable from the interactions with the oracles when $b = 1$.

For every set of identities $\mathcal{S}_{\mathsf{id}}$, we use the following action labels $\mathcal{L}$ to represent symbolic calls to the $(q, \sigma_{\mathsf{ul}})$-unlinkability oracles:

- $\text{NS}_{\text{ID}}(j)$ represents a call to $\texttt{DrawUE}_{\sigma_{\mathsf{ul}}}(\text{ID}, \_)$ when $b = 0$ or $\texttt{DrawUE}_{\sigma_{\mathsf{ul}}}(\_, \text{ID})$ when $b = 1$.
- $\text{PU}_{\text{ID}}(j,i)$ (resp. $\text{TU}_{\text{ID}}(j,i)$) is the i-th user message in the session $UE_{\text{ID}}(j)$ of the SUPI (resp. GUTI) sub-protocol.
- $\text{FU}_{\text{ID}}(j)$ is the only user message in the session $UE_{\text{ID}}(j)$ of the ASSIGN-GUTI sub-protocol.
- $\text{PN}(j,i)$ (resp. $\text{TN}(j,i)$) is the i-th network message in the session $HN(j)$ of the SUPI (resp. GUTI) sub-protocol.
- $\text{FN}(j)$ is the only network message in the session $HN(j)$ of the ASSIGN-GUTI sub-protocol.

The remaining oracle calls either have no outputs and do not modify the state (e.g. `StartSession`), or can be simulated using the oracles above. E.g., since the *HN* sends an error message whenever the protocol is not successful, the output of `ResultHN` can be deduced from the protocol messages.

Valid Action Traces We recall that an *action trace* τ is a finite sequence of action labels. Remark that some sequences of actions do not correspond to a valid execution of the protocol. E.g. since the session $UE_{\text{ID}}(j)$ cannot execute both the SUPI and the GUTI protocols, a *valid action trace* cannot contain both $\text{PU}_{\text{ID}}(j, \_)$ and $\text{TU}_{\text{ID}}(j, \_)$. Similarly, the *HN*'s second message in the SUPI protocol cannot be sent before the first message, hence $\text{PN}(j,1)$ cannot appear before $\text{PN}(j,0)$ in τ. This motivate the definition of *valid action traces*.

Definition 4.2. Let $(\mathcal{Q}_{\text{U}}^{\text{ID}})_{\text{ID} \in \mathcal{S}_{\mathsf{id}}}$ and $(\mathcal{Q}_{\text{N}}^{j})_{j \in \mathbb{N}}$ be the transition systems in Figure 4.12. A trace $\tau = \mathsf{ai}_0, \ldots, \mathsf{ai}_n$ is a *valid* action trace of the protocol AKA_N^+ if and only if τ is an interleaving of the words $w_{\text{ID}_1}, \ldots, w_{\text{ID}_N}, w_{\text{N}}^0, \ldots, w_{\text{N}}^l, \ldots$ where:

- for every $1 \leq j \leq N$, w_{ID_j} is a run of $\mathcal{Q}_{\text{U}}^{\text{ID}_j}$.
- for every $j \in \mathbb{N}$, w_{N}^j is a run of $\mathcal{Q}_{\text{N}}^j$.

Example 4.2. We give valid action traces corresponding to the honest execution of AKA_N^+ between $UE_{\text{ID}}(i)$ and $HN(j)$. If the SUPI protocol is used, we have the trace $\tau_{\text{SUPI}}^{i,j}(\text{ID})$:

$$\text{PU}_{\text{ID}}(i,0),\ \text{PN}(j,0), \text{PU}_{\text{ID}}(i,1), \text{PN}(j,1), \text{PU}_{\text{ID}}(i,2), \text{FN}(j), \text{FU}_{\text{ID}}(i)$$

And if the GUTI sub-protocol is used, the trace $\tau_{\text{GUTI}}^{i,j}(\text{ID})$:

$$\text{TU}_{\text{ID}}(i,0), \text{TN}(j,0), \text{TU}_{\text{ID}}(i,1), \text{TN}(j,1), \text{FN}(j), \text{FU}_{\text{ID}}(i)$$

Which such notations, the left trace τ_l of the attack described in Figure 4.11, in which the adversary only interacts with A, is:

$$\text{TU}_{\mathsf{A}}(0,0), \text{TN}(0,0), \text{TU}_{\mathsf{A}}(0,1), \tau_{\text{SUPI}}^{1,1}(\mathsf{A}), \text{TN}(0,1), \tau_{\text{GUTI}}^{2,2}(\mathsf{A})$$

Similarly, we can give the right trace τ_r in which the adversary interacts with A and B:

$$\text{TU}_{\mathsf{A}}(0,0), \text{TN}(0,0), \text{TU}_{\mathsf{A}}(0,1), \tau_{\text{SUPI}}^{0,1}(\mathsf{B}), \text{TN}(0,1), \tau_{\text{GUTI}}^{1,2}(\mathsf{B})$$ □

4.6.2 The AKA$^+$ Protocol Symbolic Outputs and State Updates

In Section 2.4 of Chapter 2, we represented the symbolic output t_{ai} and symbolic state update $\sigma_{\mathsf{ai}}^{\mathsf{up}}$ of a protocol P when executing the action $\mathsf{ai} \in \mathcal{L}$ using terms with variables in $\{\mathsf{x}_{\mathsf{in}}\} \cup \mathsf{Vars}_\sigma$. Then, for any action trace τ, this defined the symbolic trace:

$$\mathsf{s\text{-}trace}_\tau^P = (\_, \phi_i, \sigma_i)_{0 \leq i \leq l}$$

where ϕ_l is a finite sequence of ground terms representing the sequence of messages observed by the adversary when executing the protocol P on the action trace τ, and σ_l is the symbolic state after τ. Recall that in Chapter 2, we let ϕ_τ^P be the last symbolic frame in $\mathsf{s\text{-}trace}_\tau^P$, i.e. ϕ_l. Similarly, we define the last symbolic state σ_τ^P, and the last message observe t_τ^P.

Definition 4.3. For every action trace τ and protocol P, if $\mathsf{s\text{-}trace}_\tau^P = (\_, \phi_i, \sigma_i)_{0 \leq i \leq l}$ and t is the last term in ϕ_l, i.e. $\phi_l \equiv \_, t$, then we let:

$$t_\tau^P \equiv t \qquad\qquad \sigma_\tau^P \equiv \sigma_l$$

Moreover, if τ_0 is the largest strict prefix of τ, i.e. $\tau = \tau_0, \mathsf{ai}$, then we let $\sigma_\tau^{\mathsf{in},\mathsf{P}} \equiv \sigma_{\tau_0}^P$ be the symbolic state before the execution of the last action; and $\phi_\tau^{\mathsf{in},\mathsf{P}} \equiv \phi_{\tau_0}^P$ be the sequence of all messages observed during the execution of τ, except for the last message.

Description of AKA$^+$ We describe the symbolic messages and state updates of AKA$^+$. In this chapter, we only consider the AKA$^+$ protocol. Therefore, when the number of identities N is irrelevant, we omit the protocol name and write ϕ_τ, σ_τ and t_τ instead of $\phi_\tau^{\text{AKA}_N^+}$, $\sigma_\tau^{\text{AKA}_N^+}$ and $t_\tau^{\text{AKA}_N^+}$. We start by definition the initial frame ϕ_ϵ and initial symbolic state.

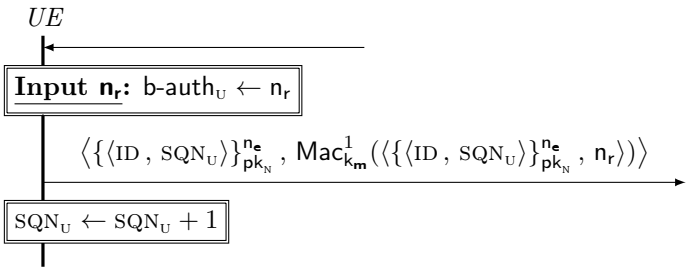
Definition 4.4. The initial frame of the AKA$^+$ protocol is $\phi_\epsilon \equiv \mathsf{pk}_{\text{N}}$, and its initial symbolic state σ_ϵ is the function from Vars_σ to terms defined by having, for every $\text{ID} \in \mathcal{S}_{\mathsf{id}}$ and $j \in \mathbb{N}$:

$$\sigma_\epsilon(\text{SQN}_{\text{U}}^{\text{ID}}) \equiv \mathsf{sqn\text{-}init}_{\text{U}}^{\text{ID}} \qquad \sigma_\epsilon(\text{SQN}_{\text{N}}^{\text{ID}}) \equiv \mathsf{sqn\text{-}init}_{\text{N}}^{\text{ID}} \qquad \sigma_\epsilon(\text{GUTI}_{\text{X}}^{\text{ID}}) \equiv \mathsf{UnSet} \qquad \sigma_\epsilon(\mathsf{e\text{-}auth}_{\text{U}}^{\text{ID}}) \equiv \mathsf{fail}$$

$$\sigma_\epsilon(\mathsf{b\text{-}auth}_{\text{U}}^{\text{ID}}) \equiv \mathsf{fail} \qquad \sigma_\epsilon(\mathsf{e\text{-}auth}_{\text{N}}^{j}) \equiv \mathsf{fail} \qquad \sigma_\epsilon(\mathsf{b\text{-}auth}_{\text{N}}^{j}) \equiv \mathsf{fail} \qquad \sigma_\epsilon(\mathsf{s\text{-}valid\text{-}guti}_{\text{U}}^{\text{ID}}) \equiv \mathsf{false}$$

$$\sigma_\epsilon(\mathsf{valid\text{-}guti}_{\text{U}}^{\text{ID}}) \equiv \mathsf{false} \qquad \sigma_\epsilon(\mathsf{session}_{\text{N}}^{\text{ID}}) \equiv \mathsf{UnSet}$$

Now, for every action label ai, we define t_{ai} and $\sigma^{\mathsf{up}}_{\mathsf{ai}}$ using the variables $\{\mathsf{x_{in}}\} \cup \mathsf{Vars}_\sigma$. As an example, we describe the second message and state update of the session $UE_{\text{ID}}(j)$ for the SUPI sub-protocol, which corresponds to the action $\mathsf{ai} = \text{PU}_{\text{ID}}(j, 1)$. We recall the relevant part of Figure 4.6:



First, we build a term representing the asymmetric encryption of the pair containing the *UE*'s permanent identity ID and its sequence number. The permanent identity ID is simply represented using a constant function symbol ID, and UE_{ID}'s sequence number is stored in the variable $\text{SQN}^{\text{ID}}_{\text{U}}$. Finally, we use the asymmetric encryption function symbol to build the term $t^{\mathsf{enc}}_{\mathsf{ai}} \equiv \{\langle \text{ID}, \text{SQN}^{\text{ID}}_{\text{U}} \rangle\}^{\mathsf{n}^j_{\mathsf{e}}}_{\mathsf{pk}_{\text{N}}}$. Notice that the encryption is randomized using a nonce $\mathsf{n}^j_{\mathsf{e}}$, and that the freshness of the randomness is guaranteed by indexing the nonce with the session number j. Finally, we can give t_{ai} and $\sigma^{\mathsf{up}}_{\mathsf{ai}}$:

$$t_{\mathsf{ai}} \equiv \langle t^{\mathsf{enc}}_{\mathsf{ai}}, \mathsf{Mac}^1_{\mathsf{k_m}}(\langle t^{\mathsf{enc}}_{\mathsf{ai}}, \mathsf{x_{in}} \rangle) \rangle \qquad \sigma^{\mathsf{up}}_{\mathsf{ai}} \equiv \begin{cases} \text{SQN}^{\text{ID}}_{\text{U}} \mapsto \mathsf{suc}(\text{SQN}^{\text{ID}}_{\text{U}}) & \mathsf{e\text{-}auth}^{\text{ID}}_{\text{U}} \mapsto \mathsf{fail} \\ \mathsf{b\text{-}auth}^{\text{ID}}_{\text{U}} \mapsto \mathsf{x_{in}} & \text{GUTI}^{\text{ID}}_{\text{U}} \mapsto \mathsf{UnSet} \\ \mathsf{valid\text{-}guti}^{\text{ID}}_{\text{U}} \mapsto \mathsf{false} \end{cases}$$

Remark that we omitted some state updates in the description of the protocol in Figure 4.6. For example, UE_{ID} temporary identity $\text{GUTI}^{\text{ID}}_{\text{U}}$ is reset when starting the SUPI sub-protocol. In the Bana-Comon model, these details are made explicit.

The description of t_{ai} and $\sigma^{\mathsf{up}}_{\mathsf{ai}}$ for the other actions can be found in Figure 4.13 and Figure 4.14. Observe that we describe one more message for the SUPI and GUTI protocols than in Section 4.4. This is because we added one message ($\text{PU}_{\text{ID}}(j, 2)$ for SUPI and $\text{TN}(j, 1)$ for GUTI) for proof purposes, to simulate the `ResultUE` and `ResultHN` oracles. Also, notice that in the GUTI protocol, when *HN* receives an unassigned GUTI, it sends a decoy message to a special dummy identity ID_{dum}.

Remark 4.3. For every action trace $\tau = \tau_0, \mathsf{ai}$, the symbolic term t_τ can be obtained from t_{ai} by replacing every occurrence of $\mathsf{x_{in}}$ by $g(\phi^{\mathsf{in}}_\tau)$, and every state variable $\mathsf{x} \in \mathsf{Vars}_\sigma$ by $\sigma^{\mathsf{in}}_\tau(\mathsf{x})$. For example, if $\mathsf{ai} = \text{PU}_{\text{ID}}(j, 1)$, then $t^{\mathsf{enc}}_\tau \equiv \{\langle \text{ID}, \sigma^{\mathsf{in}}_\tau(\text{SQN}^{\text{ID}}_{\text{U}}) \rangle\}^{\mathsf{n}^j_{\mathsf{e}}}_{\mathsf{pk}_{\text{N}}}$ and:

$$t_\tau \equiv \langle t^{\mathsf{enc}}_\tau, \mathsf{Mac}^1_{\mathsf{k_m}}(\langle t^{\mathsf{enc}}_\tau, g(\phi^{\mathsf{in}}_\tau) \rangle) \rangle \qquad \sigma^{\mathsf{up}}_\tau \equiv \begin{cases} \text{SQN}^{\text{ID}}_{\text{U}} \mapsto \mathsf{suc}(\sigma^{\mathsf{in}}_\tau(\text{SQN}^{\text{ID}}_{\text{U}})) & \mathsf{e\text{-}auth}^{\text{ID}}_{\text{U}} \mapsto \mathsf{fail} \\ \mathsf{b\text{-}auth}^{\text{ID}}_{\text{U}} \mapsto g(\phi^{\mathsf{in}}_\tau) & \text{GUTI}^{\text{ID}}_{\text{U}} \mapsto \mathsf{UnSet} \\ \mathsf{valid\text{-}guti}^{\text{ID}}_{\text{U}} \mapsto \mathsf{false} \end{cases}$$

We can get σ_τ similarly. In Figure 4.13 and Figure 4.14, several intermediate terms are defined for some action labels, e.g. $\mathsf{accept}^{\text{ID}}_{\mathsf{ai}}$, $\mathsf{msg}^{\text{ID}}_{\mathsf{ai}}$. Here also we lift these definitions to action traces. For example, for $\mathsf{ai} = \text{PU}_{\text{ID}}(j, 2)$ and $\tau = \_, \mathsf{ai}$, we let:

$$\mathsf{accept}^{\text{ID}}_\tau \equiv \mathsf{eq}(g(\phi^{\mathsf{in}}_\tau), \mathsf{Mac}^2_{\mathsf{k_m}}(\langle \sigma^{\mathsf{in}}_\tau(\mathsf{b\text{-}auth}^{\text{ID}}_{\text{U}}), \sigma^{\mathsf{in}}_\tau(\text{SQN}^{\text{ID}}_{\text{U}}) \rangle)) \qquad \square$$

4.6.3 Modeling σ-Unlinkability

We associate, to any execution of the $(q, \sigma_{\mathsf{ul}})$-unlinkability game with a fixed trace of oracle calls, a pair of action traces (τ_l, τ_r), which corresponds to the adversary's interactions with the oracles when b is, respectively, 0 and 1. We do this as follows:

- First, we consider a valid action trace τ on a set of identities $\mathcal{S}_{\mathsf{vh}}$, seen as virtual handlers. The trace τ is the sequence of oracle calls as seen by the adversary.
- We consider a mapping θ_l which associates, to every virtual handler in $\mathcal{S}_{\mathsf{vh}}$, an identity in $\mathcal{S}_{\mathsf{id}}$, where $\mathcal{S}_{\mathsf{id}} = \{\text{ID}_1, \ldots, \text{ID}_N\}$. This mapping must check that new virtual handlers are associated to

Case $\mathsf{ai} = \text{NS}_{\text{ID}}(j)$. $\sigma^{\mathsf{up}}_{\mathsf{ai}} \equiv \mathsf{valid\text{-}guti}^{\text{ID}}_{\text{U}} \mapsto \mathsf{false}$

Case $\mathsf{ai} = \text{PU}_{\text{ID}}(j,0)$. $t_{\mathsf{ai}} \equiv \texttt{Request\_Challenge}$

Case $\mathsf{ai} = \text{PN}(j,0)$. $t_{\mathsf{ai}} \equiv \mathsf{n}^j$

Case $\mathsf{ai} = \text{PU}_{\text{ID}}(j,1)$. Let $t^{\mathsf{enc}}_{\mathsf{ai}} \equiv \{\langle \text{ID}\,,\, \text{SQN}^{\text{ID}}_{\text{U}}\rangle\}^{\mathsf{n}^j_{\mathsf{e}}}_{\mathsf{pk}_{\text{N}}}$, then:

$$
\begin{aligned}
t_{\mathsf{ai}} &\equiv \left\langle t^{\mathsf{enc}}_{\mathsf{ai}}\,,\, \mathsf{Mac}^1_{\mathsf{k}^{\text{ID}}_{\mathsf{m}}}(\langle t^{\mathsf{enc}}_{\mathsf{ai}}\,,\, \mathsf{x}_{\mathsf{in}}\rangle)\right\rangle \\
\sigma^{\mathsf{up}}_{\mathsf{ai}} &\equiv \begin{cases} \text{SQN}^{\text{ID}}_{\text{U}} \mapsto \mathsf{suc}(\text{SQN}^{\text{ID}}_{\text{U}}) & \mathsf{e\text{-}auth}^{\text{ID}}_{\text{U}} \mapsto \mathsf{fail} \qquad \mathsf{valid\text{-}guti}^{\text{ID}}_{\text{U}} \mapsto \mathsf{false} \\ \mathsf{b\text{-}auth}^{\text{ID}}_{\text{U}} \mapsto \mathsf{x}_{\mathsf{in}} & \text{GUTI}^{\text{ID}}_{\text{U}} \mapsto \mathsf{UnSet} \end{cases}
\end{aligned}
$$

Case $\mathsf{ai} = \text{PN}(j,1)$. Let $t_{\mathsf{dec}} \equiv \mathsf{dec}(\pi_1(\mathsf{x}_{\mathsf{in}}), \mathsf{sk}_{\text{N}})$, and let:

$$
\begin{aligned}
\mathsf{accept}^{\text{ID}_i}_{\mathsf{ai}} &\equiv \mathsf{eq}(\pi_2(\mathsf{x}_{\mathsf{in}}), \mathsf{Mac}^1_{\mathsf{k}^{\text{ID}_i}_{\mathsf{m}}}(\langle \pi_1(\mathsf{x}_{\mathsf{in}})\,,\, \mathsf{n}^j\rangle)) \;\wedge\; \mathsf{eq}(\pi_1(t_{\mathsf{dec}}), \text{ID}_i) \\
\mathsf{inc\text{-}accept}^{\text{ID}_i}_{\mathsf{ai}} &\equiv \mathsf{accept}^{\text{ID}_i}_{\mathsf{ai}} \wedge \mathsf{geq}(\pi_2(t_{\mathsf{dec}}), \text{SQN}^{\text{ID}_i}_{\text{N}})
\end{aligned}
$$

$$
\begin{aligned}
t_{\mathsf{ai}} \equiv\; & \mathsf{if}\ \mathsf{accept}^{\text{ID}_1}_{\mathsf{ai}}\ \mathsf{then}\ \mathsf{Mac}^2_{\mathsf{k}^{\text{ID}_1}_{\mathsf{m}}}(\langle \mathsf{n}^j\,,\, \mathsf{suc}(\pi_2(t_{\mathsf{dec}}))\rangle) \\
& \mathsf{else\ if}\ \mathsf{accept}^{\text{ID}_2}_{\mathsf{ai}}\ \mathsf{then}\ \mathsf{Mac}^2_{\mathsf{k}^{\text{ID}_2}_{\mathsf{m}}}(\langle \mathsf{n}^j\,,\, \mathsf{suc}(\pi_2(t_{\mathsf{dec}}))\rangle) \\
& \quad \cdots \\
& \mathsf{else\ UnknownId}
\end{aligned}
$$

$$
\sigma^{\mathsf{up}}_{\mathsf{ai}} \equiv \begin{cases}
\mathsf{session}^{\text{ID}_i}_{\text{N}} \mapsto \mathsf{if\ inc\text{-}accept}^{\text{ID}_i}_{\mathsf{ai}}\ \mathsf{then\ n}^j\ \mathsf{else\ session}^{\text{ID}_i}_{\text{N}} \\
\text{GUTI}^{\text{ID}_i}_{\text{N}} \mapsto \mathsf{if\ inc\text{-}accept}^{\text{ID}_i}_{\mathsf{ai}}\ \mathsf{then}\ \text{GUTI}^j\ \mathsf{else}\ \text{GUTI}^{\text{ID}_i}_{\text{N}} \\
\text{SQN}^{\text{ID}_i}_{\text{N}} \mapsto \mathsf{if\ inc\text{-}accept}^{\text{ID}_i}_{\mathsf{ai}}\ \mathsf{then\ suc}(\pi_2(t_{\mathsf{dec}}))\ \mathsf{else}\ \text{SQN}^{\text{ID}_i}_{\text{N}} \\
\mathsf{b\text{-}auth}^j_{\text{N}}, \mathsf{e\text{-}auth}^j_{\text{N}} \mapsto \mathsf{if}\ \mathsf{accept}^{\text{ID}_1}_{\mathsf{ai}}\ \mathsf{then}\ \text{ID}_1 \\
\qquad\qquad\qquad\qquad \mathsf{else\ if}\ \mathsf{accept}^{\text{ID}_2}_{\mathsf{ai}}\ \mathsf{then}\ \text{ID}_2 \\
\qquad\qquad\qquad\qquad\quad \cdots \\
\qquad\qquad\qquad\qquad \mathsf{else\ UnknownId}
\end{cases}
$$

Case $\mathsf{ai} = \text{PU}_{\text{ID}}(j,2)$.

$$
\begin{aligned}
\mathsf{accept}^{\text{ID}}_{\mathsf{ai}} &\equiv \mathsf{eq}(\mathsf{x}_{\mathsf{in}}, \mathsf{Mac}^2_{\mathsf{k}^{\text{ID}}_{\mathsf{m}}}(\langle \mathsf{b\text{-}auth}^{\text{ID}}_{\text{U}}\,,\, \text{SQN}^{\text{ID}}_{\text{U}}\rangle)) \\
t_{\mathsf{ai}} &\equiv \mathsf{if\ accept}^{\text{ID}}_{\mathsf{ai}}\ \mathsf{then\ ok\ else\ error} \\
\sigma^{\mathsf{up}}_{\mathsf{ai}} &\equiv \mathsf{e\text{-}auth}^{\text{ID}}_{\text{U}} \mapsto \mathsf{if\ accept}^{\text{ID}}_{\mathsf{ai}}\ \mathsf{then\ b\text{-}auth}^{\text{ID}}_{\text{U}}\ \mathsf{else\ fail}
\end{aligned}
$$

Case $\mathsf{ai} = \text{FN}(j)$.

$$
\begin{aligned}
\mathsf{msg}^{\text{ID}_i}_{\mathsf{ai}} &\equiv \langle \text{GUTI}^j \oplus \mathsf{f}^{\mathsf{r}}_{\mathsf{k}^{\text{ID}_i}}(\mathsf{n}^j)\,,\, \mathsf{Mac}^5_{\mathsf{k}^{\text{ID}_i}_{\mathsf{m}}}(\langle \text{GUTI}^j\,,\, \mathsf{n}^j\rangle)\rangle \\
t_{\mathsf{ai}} &\equiv \mathsf{if\ eq}(\mathsf{e\text{-}auth}^j_{\text{N}}, \text{ID}_1)\ \mathsf{then\ msg}^{\text{ID}_1}_{\mathsf{ai}} \\
&\qquad \mathsf{else\ if\ eq}(\mathsf{e\text{-}auth}^j_{\text{N}}, \text{ID}_2)\ \mathsf{then\ msg}^{\text{ID}_2}_{\mathsf{ai}} \\
&\qquad\quad \cdots \\
&\qquad \mathsf{else\ UnknownId}
\end{aligned}
$$

Case $\mathsf{ai} = \text{FU}_{\text{ID}}(j)$. Let $t_{\text{GUTI}} \equiv \pi_1(\mathsf{x}_{\mathsf{in}}) \oplus \mathsf{f}^{\mathsf{r}}_{\mathsf{k}^{\text{ID}}}(\mathsf{e\text{-}auth}^{\text{ID}}_{\text{U}})$, then:

$$
\begin{aligned}
\mathsf{accept}^{\text{ID}}_{\mathsf{ai}} &\equiv \mathsf{eq}(\pi_2(\mathsf{x}_{\mathsf{in}}), \mathsf{Mac}^5_{\mathsf{k}^{\text{ID}}_{\mathsf{m}}}(\langle t_{\text{GUTI}}\,,\, \mathsf{e\text{-}auth}^{\text{ID}}_{\text{U}}\rangle)) \;\wedge\; \neg\mathsf{eq}(\mathsf{e\text{-}auth}^{\text{ID}}_{\text{U}}, \mathsf{fail}) \\
t_{\mathsf{ai}} &\equiv \mathsf{if\ accept}^{\text{ID}}_{\mathsf{ai}}\ \mathsf{then\ ok\ else\ error} \\
\sigma^{\mathsf{up}}_{\mathsf{ai}} &\equiv \begin{cases} \mathsf{valid\text{-}guti}^{\text{ID}}_{\text{U}} \mapsto \mathsf{accept}^{\text{ID}}_{\mathsf{ai}} \\ \text{GUTI}^{\text{ID}}_{\text{U}} \mapsto \mathsf{if\ accept}^{\text{ID}}_{\mathsf{ai}}\ \mathsf{then}\ t_{\text{GUTI}}\ \mathsf{else\ UnSet} \end{cases}
\end{aligned}
$$

Convention: For every $j \in \mathbb{N}$, $\text{GUTI}^j \in \mathcal{N}$.

Figure 4.13: The Symbolic Terms and States for $\text{NS}_{\text{ID}}(j)$ and the SUPI and ASSIGN-GUTI Sub-Protocols.

Case $\mathsf{ai} = \mathrm{TU}_{\mathrm{ID}}(j,0)$.

$$
\begin{aligned}
t_{\mathsf{ai}} &\equiv \mathsf{if}\ \mathsf{valid\text{-}guti}_{\mathrm{U}}^{\mathrm{ID}}\ \mathsf{then}\ \mathrm{GUTI}_{\mathrm{U}}^{\mathrm{ID}}\ \mathsf{else}\ \mathsf{NoGuti}\\
\sigma_{\mathsf{ai}}^{\mathsf{up}} &\equiv \begin{cases} \mathsf{valid\text{-}guti}_{\mathrm{U}}^{\mathrm{ID}} \mapsto \mathsf{false} & \mathsf{e\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}} \mapsto \mathsf{fail}\\ \mathsf{s\text{-}valid\text{-}guti}_{\mathrm{U}}^{\mathrm{ID}} \mapsto \mathsf{valid\text{-}guti}_{\mathrm{U}}^{\mathrm{ID}} & \mathsf{b\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}} \mapsto \mathsf{fail}\end{cases}
\end{aligned}
$$

Case $\mathsf{ai} = \mathrm{TN}(j,0)$. Let $t_{\oplus}^{\mathrm{ID}_i} \equiv \mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}_i} \oplus \mathsf{f}_{\mathsf{k}^{\mathrm{ID}_i}}(\mathsf{n}^j)$, then:

$$
\begin{aligned}
\mathsf{msg}_{\mathsf{ai}}^{\mathrm{ID}_i} &\equiv \langle \mathsf{n}^j, t_{\oplus}^{\mathrm{ID}_i}, \mathsf{Mac}^3_{\mathsf{k}^{\mathrm{ID}_i}_{\mathsf{m}}}(\langle \mathsf{n}^j, \mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}_i}, \mathrm{GUTI}_{\mathrm{N}}^{\mathrm{ID}_i}\rangle)\rangle\\
\mathsf{accept}_{\mathsf{ai}}^{\mathrm{ID}_i} &\equiv \mathsf{eq}(\mathrm{GUTI}_{\mathrm{N}}^{\mathrm{ID}_i}, \mathsf{x_{in}}) \wedge \neg \mathsf{eq}(\mathrm{GUTI}_{\mathrm{N}}^{\mathrm{ID}_i}, \mathsf{UnSet})\\
t_{\mathsf{ai}} &\equiv \mathsf{if}\ \mathsf{accept}_{\mathsf{ai}}^{\mathrm{ID}_1}\ \mathsf{then}\ \mathsf{msg}_{\mathsf{ai}}^{\mathrm{ID}_1}\\
&\quad\ \mathsf{else\ if}\ \mathsf{accept}_{\mathsf{ai}}^{\mathrm{ID}_2}\ \mathsf{then}\ \mathsf{msg}_{\mathsf{ai}}^{\mathrm{ID}_2}\\
&\qquad \cdots\\
&\quad\ \mathsf{else}\ \mathsf{msg}_{\mathsf{ai}}^{\mathrm{ID}_{\mathsf{dum}}}\\
\sigma_{\mathsf{ai}}^{\mathsf{up}} &\equiv \begin{cases}
\mathrm{GUTI}_{\mathrm{N}}^{\mathrm{ID}_i} \mapsto \mathsf{if}\ \mathsf{accept}_{\mathsf{ai}}^{\mathrm{ID}_i}\ \mathsf{then}\ \mathsf{UnSet}\ \mathsf{else}\ \mathrm{GUTI}_{\mathrm{N}}^{\mathrm{ID}_i}\\
\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}_i} \mapsto \mathsf{if}\ \mathsf{accept}_{\mathsf{ai}}^{\mathrm{ID}_i}\ \mathsf{then}\ \mathsf{n}^j\ \mathsf{else}\ \mathsf{session}_{\mathrm{N}}^{\mathrm{ID}_i}\\
\mathsf{b\text{-}auth}_{\mathrm{N}}^{j} \mapsto \mathsf{if}\ \mathsf{accept}_{\mathsf{ai}}^{\mathrm{ID}_1}\ \mathsf{then}\ \mathrm{ID}_1\\
\qquad\qquad \mathsf{else\ if}\ \mathsf{accept}_{\mathsf{ai}}^{\mathrm{ID}_2}\ \mathsf{then}\ \mathrm{ID}_2\\
\qquad\qquad\qquad \cdots\\
\qquad\qquad \mathsf{else\ UnknownId}
\end{cases}
\end{aligned}
$$

Case $\mathsf{ai} = \mathrm{TU}_{\mathrm{ID}}(j,1)$. Let $t_{\mathrm{SQN}} \equiv \pi_2(\mathsf{x_{in}}) \oplus \mathsf{f}_{\mathsf{k}^{\mathrm{ID}}}(\pi_1(\mathsf{x_{in}}))$, then:

$$
\begin{aligned}
\mathsf{accept}_{\mathsf{ai}}^{\mathrm{ID}} &\equiv \mathsf{eq}(\pi_3(\mathsf{x_{in}}), \mathsf{Mac}^3_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle \pi_1(\mathsf{x_{in}}), t_{\mathrm{SQN}}, \mathrm{GUTI}_{\mathrm{U}}^{\mathrm{ID}}\rangle)) \wedge\ \mathsf{s\text{-}valid\text{-}guti}_{\mathrm{U}}^{\mathrm{ID}}\ \wedge\ \mathsf{range}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}, t_{\mathrm{SQN}})\\
t_{\mathsf{ai}} &\equiv \mathsf{if}\ \mathsf{accept}_{\mathsf{ai}}^{\mathrm{ID}}\ \mathsf{then}\ \mathsf{Mac}^4_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\pi_1(\mathsf{x_{in}}))\ \mathsf{else\ error}\\
\sigma_{\mathsf{ai}}^{\mathsf{up}} &\equiv \begin{cases}
\mathsf{b\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}, \mathsf{e\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}} \mapsto \mathsf{if}\ \mathsf{accept}_{\mathsf{ai}}^{\mathrm{ID}}\ \mathsf{then}\ \pi_1(\mathsf{x_{in}})\ \mathsf{else\ fail}\\
\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}} \mapsto \mathsf{if}\ \mathsf{accept}_{\mathsf{ai}}^{\mathrm{ID}}\ \mathsf{then}\ \mathsf{suc}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})\ \mathsf{else}\ \mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}
\end{cases}
\end{aligned}
$$

Case $\mathsf{ai} = \mathrm{TN}(j,1)$.

$$
\begin{aligned}
\mathsf{accept}_{\mathsf{ai}}^{\mathrm{ID}_i} &\equiv \mathsf{eq}(\mathsf{x_{in}}, \mathsf{Mac}^4_{\mathsf{k}^{\mathrm{ID}_i}_{\mathsf{m}}}(\mathsf{n}^j)) \wedge \mathsf{eq}(\mathsf{b\text{-}auth}_{\mathrm{N}}^{j}, \mathrm{ID}_i)\\
\mathsf{inc\text{-}accept}_{\mathsf{ai}}^{\mathrm{ID}_i} &\equiv \mathsf{accept}_{\mathsf{ai}}^{\mathrm{ID}_i} \wedge \mathsf{eq}(\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}_i}, \mathsf{n}^j)\\
t_{\mathsf{ai}} &\equiv \mathsf{if}\ \textstyle\bigvee_i \mathsf{accept}_{\mathsf{ai}}^{\mathrm{ID}_i}\ \mathsf{then\ ok\ else\ error}\\
\sigma_{\mathsf{ai}}^{\mathsf{up}} &\equiv \begin{cases}
\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}_i} \mapsto \mathsf{if}\ \mathsf{inc\text{-}accept}_{\mathsf{ai}}^{\mathrm{ID}_i}\ \mathsf{then}\ \mathsf{suc}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}_i})\\
\qquad\qquad\qquad\qquad \mathsf{else}\ \mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}_i}\\
\mathrm{GUTI}_{\mathrm{N}}^{\mathrm{ID}_i} \mapsto \mathsf{if}\ \mathsf{inc\text{-}accept}_{\mathsf{ai}}^{\mathrm{ID}_i}\ \mathsf{then}\ \mathrm{GUTI}^j\ \mathsf{else}\ \mathrm{GUTI}_{\mathrm{N}}^{\mathrm{ID}_i}\\
\mathsf{e\text{-}auth}_{\mathrm{N}}^{j} \mapsto \mathsf{if}\ \mathsf{accept}_{\mathsf{ai}}^{\mathrm{ID}_1}\ \mathsf{then}\ \mathrm{ID}_1\\
\qquad\qquad \mathsf{else\ if}\ \mathsf{accept}_{\mathsf{ai}}^{\mathrm{ID}_2}\ \mathsf{then}\ \mathrm{ID}_2\\
\qquad\qquad\qquad \cdots\\
\qquad\qquad \mathsf{else\ UnknownId}
\end{cases}
\end{aligned}
$$

Convention: For every $j \in \mathbb{N}$, $\mathrm{GUTI}^j \in \mathcal{N}$.

Figure 4.14: The Symbolic Terms and States the GUTI Sub-Protocol.

identities in l_{free}: for every identity $\text{ID} \in \mathcal{S}_{\mathsf{id}}$, this mapping must be such that, at any point in τ, there is at most one virtual handler which is alive and mapped to ID by θ_l. Similarly, we consider a mapping θ_r for the right side.

- Finally, we let τ_l be the action trace obtained from τ by replacing the virtual handler by the corresponding concrete identities using θ_l, and re-numbering the session numbers. We define similarly τ_r using θ_r. Then $\mathcal{R}^N_{\mathsf{ul}}$ contains the pair of action trace (τ_l, τ_r).

We define what it means for the AKA^+_N protocol to be is σ_{ul}-unlinkable.

Definition 4.5. The protocol AKA^+_N is σ_{ul}-unlinkable in any computational model satisfying the axioms Ax if, for every $(\tau_l, \tau_r) \in \mathcal{R}^N_{\mathsf{ul}}$, we can derive $\phi_{\tau_l} \sim \phi_{\tau_r}$ using Ax.

Proposition 4.1. *$\mathcal{R}^N_{ul}$ is reflexive, symmetric and transitive. Moreover, for every $\tau \in \mathit{support}(\mathcal{R}^N_{ul})$, τ is a valid action trace of AKA^+_N.*

Proof. We show this by induction over the valid action trace τ_{vh}, on virtual identities $\mathcal{S}_{\mathsf{vh}}$, used to define τ. We omit the details. ■

Most Anonymised Trace Given an action trace $\tau \in \mathsf{support}(\mathcal{R}^N_{\mathsf{ul}})$, there is a particular and unique action trace $\underline{\tau}$ which is the "most anonymised trace" corresponding to τ. Intuitively, $\underline{\tau}$ is the trace τ where we changed a user identity every time we could (i.e. every time $\text{NS}_{\text{ID}}(\_)$ appears). This is useful to prove that the AKA^+ protocol is σ_{ul}-unlinkable, as it reduces the number of cases we have to consider: we only need to show that we can derive $\phi_\tau \sim \phi_{\underline{\tau}}$ for every $\tau \in \mathsf{support}(\mathcal{R}^N_{\mathsf{ul}})$.

There is a small difficulty here: the number if identities in $\underline{\tau}$ is not the same as in τ. Therefore, on the right side we need to consider an execution of the AKA^+ protocol with more identities. More precisely, since the number of identities in $\underline{\tau}$ is upper-bounded by $|\mathcal{S}_{\mathsf{id}}| \times |\tau| = N \times |\tau|$, it is sufficient to prove that for every $\tau \in \mathsf{support}(\mathcal{R}^N_{\mathsf{ul}})$, for every $\underline{N} \geq N \times |\tau|$, there exists a derivation of:

$$\phi_\tau^{\text{AKA}^+_N} \sim \phi_{\underline{\tau}}^{\text{AKA}^+_{\underline{N}}} \tag{4.1}$$

To do this, we consider a countable subset $\mathcal{S}_{\mathsf{bid}} = \{\mathsf{A}_i \mid i \in \mathbb{N}\}$ of $\mathcal{S}^\omega_{\mathsf{id}}$. The set $\mathcal{S}_{\mathsf{bid}}$ is a set of base identities. Then, for every base identity A_i, we have copies $\mathsf{A}_i = \mathsf{A}_{i,1}, \ldots, \mathsf{A}_{i,C}, \ldots$ of A_i. The first copy $\mathsf{A}_{i,1}$ is always A_i, and all the copies are distinct function symbols. Moreover, for every $(i,j) \neq (i',j')$, the function symbols $\mathsf{A}_{i,j}$ and $\mathsf{A}_{i',j'}$ are distinct.

Definition 4.6. Given an identity $\mathsf{A}_{b,c}$, we let $\mathsf{fresh\text{-}id}(\mathsf{A}_{b,c}) = \mathsf{A}_{b,c+1}$, and given a base identity $\mathsf{A}_{b,1}$ we let $\mathsf{copies\text{-}id}_C(\mathsf{A}_{b,1}) = \{\mathsf{A}_{b,i} \mid 1 \leq i \leq C\}$. We require that all these identities are distinct:

$$\mathcal{S}^\omega_{\mathsf{id}} \;=\; \biguplus_{i,j \in \mathbb{N}} \{\mathsf{A}_{i,j}\}$$

where $\uplus$ denotes the disjoint union.

A *basic action trace* is an action trace using only base identities $\{\mathsf{A}_{b,1} \mid b \in \mathbb{N}\}$.

Definition 4.7. An action trace τ is *basic* if it only uses network action labels and user action labels $\mathsf{X}_{\text{ID}}(\_)$ where ID is a base identity, i.e. $\text{ID} \in \mathcal{S}_{\mathsf{bid}}$.

Then, for every basic action trace τ, we let $\underline{\tau}$ be the most anonymised action trace corresponding to τ.

Definition 4.8. For every basic action trace τ, we let $\underline{\tau}$ be the action trace obtained from τ by replacing, each time we encounter an action $\text{NS}_{\text{ID}}(j)$, all subsequent actions with agent ID by actions with agent $\mathsf{fresh\text{-}id}(\text{ID})$:

$$\underline{\tau} = \begin{cases} \text{NS}_{\nu\text{ID}}(j), \underline{\tau_0[\nu\text{ID}/\text{ID}]} & \text{when } \tau = \text{NS}_{\text{ID}}(j), \tau_0 \text{ and } \nu\text{ID} = \mathsf{fresh\text{-}id}(\text{ID}) \\ \mathsf{ai}, \underline{\tau_0} & \text{when } \tau = \mathsf{ai}, \tau_0 \text{ and } \mathsf{ai} \notin \{\text{NS}_{\text{ID}}(j) \mid \text{ID} \in \mathcal{S}_{\mathsf{id}}, j \in \mathbb{N}\} \end{cases}$$

Proposition 4.2. *If τ is a valid basic action trace on identities $\mathcal{S}_{id}$ then $\underline{\tau}$ is a valid action trace using less than $|\mathcal{S}_{id}| \times |\tau|$ distinct identities.*

Proof. The proof is straightforward by induction over τ. ■

We can check that for every $(\tau_l, \tau_r) \in \mathcal{R}^N_{\mathsf{ul}}$ we have $\underline{\tau_l} = \underline{\tau_r}$. Moreover, $\sim$ is a transitive relation. Therefore, instead of proving that for every $\mathcal{R}^N_{\mathsf{ul}}(\tau_l, \tau_r)$ the formula in (4.1) Ax, it is sufficient to show that for every $\tau \in \mathsf{support}(\mathcal{R}^N_{\mathsf{ul}})$, we can derive:

$$\phi_{\tau}^{\mathrm{AKA}^+_N} \sim \phi_{\underline{\tau}}^{\mathrm{AKA}^+_{\underline{N}}}$$

where $\underline{N}$ is larger than the number of distinct identities used in $\underline{\tau}$. Formally:

Proposition 4.3. *Let Ax be a set of axioms including Trans and Sym. The AKA^+_N protocol is σ_{ul}-unlinkable in any computational model satisfying some axioms Ax if for every $\tau \in \mathsf{support}(\mathcal{R}^N_{ul})$, for every $\underline{N} \geq N \times |\tau|$, there is a derivation using Ax of:*

$$\phi_{\tau}^{AKA^+_N} \sim \phi_{\underline{\tau}}^{AKA^+_{\underline{N}}}$$

Proof. Let $(\tau_l, \tau_r) \in \mathcal{R}^N_{\mathsf{ul}}$. Using Proposition 4.2, we know that $\underline{\tau_l}$ and $\underline{\tau_r}$ are valid action traces of $\mathrm{AKA}^+_{\underline{N}}$. Since $\underline{\tau_l} = \underline{\tau_r}$, and using the transitivity and symmetry axioms Trans and Sym, we get the wanted derivation:

$$\frac{\phi_{\tau_l}^{\mathrm{AKA}^+_N} \sim \phi_{\underline{\tau_l}}^{\mathrm{AKA}^+_{\underline{N}}} \qquad \phi_{\underline{\tau_r}}^{\mathrm{AKA}^+_{\underline{N}}} \sim \phi_{\tau_r}^{\mathrm{AKA}^+_N}}{\phi_{\tau_l}^{\mathrm{AKA}^+_N} \sim \phi_{\tau_r}^{\mathrm{AKA}^+_N}} \ (\mathsf{Trans} + \mathsf{Sym})^*$$

■

Notations We introduce some useful notations.

Definition 4.9. We define some functions on action traces:

- Given an action trace τ, we let $\prec_\tau$ be the restriction of $\prec$ to the set of strict prefixes of τ, i.e. $\tau_2 \prec_\tau \tau_1$ iff $\tau_2 \prec \tau_1$ and $\tau_1 \prec \tau$.
- We extend $\prec_\tau$ to symbolic actions as follows: we have $\mathsf{ai} \prec_\tau \tau_1$ (resp. $\tau_1 \prec_\tau \mathsf{ai}$) iff there exists τ_2 such that $\mathsf{h}(\tau_2) = \mathsf{ai}$ and $\tau_2 \prec_\tau \tau_1$ (resp. $\tau_1 \prec_\tau \tau_2$).

Definition 4.10. Given a basic trace τ and a basic identity $\mathrm{ID} = \mathsf{A}_{i,0}$, we let $\nu_\tau(\mathrm{ID})$ be the identity $\mathsf{A}_{i,l}$ where l is the number of occurrences of $\mathrm{NS}_{\mathrm{ID}}(\_)$ in τ.

4.6.4 Ghost Variable

To show that the AKA^+_N protocol is σ_{ul}-unlinkable, we need to know, for every identity $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$, if there was a successful SUPI session since the last $\mathrm{NS}_{\mathrm{ID}}(\_)$. To do this, we extend the set of variables Vars_σ by adding a ghost variable $\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}$ for every $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$. We also extend the symbolic state updates of $\mathrm{NS}_{\mathrm{ID}}(\_)$ and $\mathrm{PU}_{\mathrm{ID}}(j, 2)$ as follows:

- For $\mathsf{ai} = \mathrm{NS}_{\mathrm{ID}}(j)$:

$$\sigma^{\mathsf{up}}_\tau \equiv \begin{cases} \mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}} \mapsto \mathsf{false} \\ \mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}} \mapsto \mathsf{false} \end{cases}$$

- For $\mathsf{ai} = \mathrm{PU}_{\mathrm{ID}}(j, 2)$:

$$\sigma^{\mathsf{up}}_\tau \equiv \begin{cases} \mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}} \mapsto \mathsf{if}\ \mathsf{accept}^{\mathrm{ID}}_\tau\ \mathsf{then}\ \sigma^{\mathsf{in}}_\tau(\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}})\ \mathsf{else}\ \mathsf{fail} \\ \mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}} \mapsto \sigma^{\mathsf{in}}_\tau(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \vee \mathsf{accept}^{\mathrm{ID}}_\tau \end{cases}$$

Remark that the variable $\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}$ is read only to update its value. It is not used in the actual protocol. By consequence, the AKA^+_N protocol is σ_{ul}-unlinkable if and only if the extended AKA^+_N protocol is σ_{ul}-unlinkable. We extend the initial symbolic state σ_ϵ by adding $\sigma_\epsilon(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \equiv \mathsf{false}$.

4.7 Axioms

In this section, we describe the set of axioms used to prove the AKA^+ protocol σ_{ul}-unlinkability. First, we give the definitions of the non-standard *joint* cryptographic assumptions we use in Section 4.7.1, and prove relations among them in Section 4.7.2. We translate these assumptions into axioms of the logic in Section 4.7.3. Finally, we give the implementation axioms in Section 4.7.4, and some additional axioms in Section 4.7.5.

4.7.1 Joint Cryptographic Assumptions

In Section 2.5 of Chapter 2, we presented axioms for the CR-HK, EUF-MAC and PRF cryptographic assumptions. Unfortunately, we cannot use these axioms for AKA$^+$, as the hash functions of this protocol share the same secret key. Instead, we define variants of our cryptographic axioms for families of hash functions which are *jointly* CR-HK, EUF-MAC or PRF.

The functions $H, H_1, \ldots, H_l$ are jointly CR-HK if no adversary can build a collision for $H(\cdot, \mathsf{k_m})$, even if it has oracle access to $H(\cdot, \mathsf{k_m}), H_1(\cdot, \mathsf{k_m}), \ldots, H_l(\cdot, \mathsf{k_m})$.

Definition 4.11. A function H is CR-HK secure with a key jointly used by $H_1, \ldots, H_l$ if for every PPTM $\mathcal{A}$, the following quantity is negligible in η:

$$\mathbf{Pr}\big(\mathsf{k_m} : (m_1, m_2) \leftarrow \mathcal{A}^{\mathcal{O}_{H(\cdot,\mathsf{k_m})}, \mathcal{O}_{H_1(\cdot,\mathsf{k_m})}, \ldots, \mathcal{O}_{H_l(\cdot,\mathsf{k_m})}}(1^\eta), m_1 \neq m_2 \text{ and } H(m_1, \mathsf{k_m}) = H(m_2, \mathsf{k_m})\big)$$

where $\mathsf{k_m}$ is drawn uniformly in $\{0,1\}^\eta$.

Similarly, the functions $H, H_1, \ldots, H_l$ are jointly EUF-MAC if no adversary can forge a mac of $H(\cdot, \mathsf{k_m})$, even if it has oracle access to $H(\cdot, \mathsf{k_m}), H_1(\cdot, \mathsf{k_m}), \ldots, H_l(\cdot, \mathsf{k_m})$.

Definition 4.12. A function H is EUF-MAC secure with a key jointly used by $H_1, \ldots, H_l$ if for every PPTM $\mathcal{A}$, the following quantity is negligible in η:

$$\mathbf{Pr}\big(\mathsf{k_m} : (m, \sigma) \leftarrow \mathcal{A}^{\mathcal{O}_{H(\cdot,\mathsf{k_m})}, \mathcal{O}_{H_1(\cdot,\mathsf{k_m})}, \ldots, \mathcal{O}_{H_l(\cdot,\mathsf{k_m})}}(1^\eta), m \text{ not queried to } \mathcal{O}_{H(\cdot,\mathsf{k_m})} \text{ and } \sigma = H(m, \mathsf{k_m})\big)$$

where $\mathsf{k_m}$ is drawn uniformly in $\{0,1\}^\eta$.

Finally, the functions $H, H_1, \ldots, H_l$ are jointly PRF if they are simultaneously computationally indistinguishable from random functions $g, g_1, \ldots, g_l$.

Definition 4.13. Let $H_1(\cdot,\cdot), \ldots, H_n(\cdot,\cdot)$ be a finite family of keyed hash functions from $\{0,1\}^* \times \{0,1\}^\eta$ to $\{0,1\}^\eta$. The functions $H_1, \ldots, H_n$ are *Jointly Pseudo Random Functions* if, for any PPTM adversary $\mathcal{A}$ with access to oracles $\mathcal{O}_{f_1}, \ldots, \mathcal{O}_{f_n}$:

$$|\mathbf{Pr}(k : \mathcal{A}^{\mathcal{O}_{H_1(\cdot,k)}, \ldots, \mathcal{O}_{H_n(\cdot,k)}}(1^\eta) = 1) - \mathbf{Pr}(g_1, \ldots, g_n : \mathcal{A}^{\mathcal{O}_{g_1(\cdot)}, \ldots, \mathcal{O}_{g_n(\cdot)}}(1^\eta) = 1)|$$

is negligible, where:

- k is drawn uniformly in $\{0,1\}^\eta$.
- $g_1, \ldots, g_n$ are drawn uniformly in the set of all functions from $\{0,1\}^*$ to $\{0,1\}^\eta$.

4.7.2 Relations Among Cryptographic Assumptions

It is well known that we have the following relation between the standard cryptographic assumptions:

$$\text{PRF} \Rightarrow \text{EUF-MAC} \Rightarrow \text{CR-HK}$$

These relations have a joint version counterpart, which we prove below:

$$\textit{Joint PRF} \Rightarrow \textit{Joint EUF-MAC} \Rightarrow \textit{Joint CR-HK}$$

Proposition 4.4. *If the functions $H, H_1, \ldots, H_l$ are jointly PRF then H is EUF-MAC secure with a key jointly used by $H_1, \ldots, H_l$.*

Proof. The proof is almost the same than the proof showing that if a function H is a PRF then H is EUF-MAC secure, and is by reduction. If H is not EUF-MAC secure with a key jointly used by $H_1, \ldots, H_l$ then there exists an adversary $\mathcal{A}$ winning the corresponding game with a non-negligible probability. It is simple to build from $\mathcal{A}$ an adversary $\mathcal{B}$ against the joint PRF property of $H, H_1, \ldots, H_l$.

First, $\mathcal{B}$ runs the adversary $\mathcal{A}$, forwarding and logging its oracle calls. Eventually, $\mathcal{A}$ returns a pair (m, σ). Then, $\mathcal{B}$ queries the first oracle on m, which returns a value σ'. Finally, $\mathcal{B}$ returns 1 if and only if $\mathcal{A}$ never queried the first oracle on m and $\sigma' = \sigma$. Then:

- If $\mathcal{B}$ is interacting with the oracles $\mathcal{O}_{H(\cdot,\mathsf{k_m})}, \mathcal{O}_{H_1(\cdot,\mathsf{k_m})}, \ldots, \mathcal{O}_{H_l(\cdot,\mathsf{k_m})}$, its probability of returning 1 is exactly the advantage of $\mathcal{A}$ against the EUF-MAC game with key jointly used.
- If $\mathcal{B}$ is interacting with the oracles $\mathcal{O}_{g(\cdot)}, \mathcal{O}_{g_1(\cdot)}, \ldots, \mathcal{O}_{g_l(\cdot)}$ where $g, g_1, \ldots, g_l$ are random functions, then its probability of returning 1 is the probability of having $g(m) = \sigma$ knowing that m was never queried to g. Since g is a random function, this is less than $1/2^\eta$.

Since $\mathcal{A}$ has a non-negligible advantage against the EUF-MAC game with key jointly used, we deduce that $\mathcal{B}$ has a non-negligible advantage against the joint PRF game. ■

Proposition 4.5. *If H is* EUF-MAC *secure with a key jointly used by $H_1, \ldots, H_l$ then H is* CR *secure with a key jointly used by $H_1, \ldots, H_l$.*

Proof. Assume that we have an adversary $\mathcal{A}$ against the joint CR-HK game. We are going to build an adversary $\mathcal{B}$ against the joint EUF-MAC game. W.l.o.g. we can assume that:

- $\mathcal{A}$ makes at most $p(\eta)$ calls to the hash oracle for $\mathcal{O}_{\mathsf{H}_1(\cdot,\mathsf{k_m})}$, where p is a polynomial.
- $\mathcal{A}$ never calls the hash oracle $\mathcal{O}_{\mathsf{H}_1(\cdot,\mathsf{k_m})}$ on the same value twice.
- $\mathcal{A}$'s candidate collision pair (m_1, m_2) has been submitted to the oracle $\mathcal{O}_{\mathsf{H}_1(\cdot,\mathsf{k_m})}$. Moreover, m_2 is the last query to the oracle $\mathcal{O}_{\mathsf{H}_1(\cdot,\mathsf{k_m})}$.
- $\mathcal{A}$'s output is a well-formed message only when it is a valid collision pair.

We use $\mathcal{O}_{\vec{\mathsf{H}}(\cdot,\mathsf{k_m})}$ to denote the oracles $\mathcal{O}_{\mathsf{H}_1(\cdot,\mathsf{k_m})}, \ldots, \mathcal{O}_{\mathsf{H}_l(\cdot,\mathsf{k_m})}$. On input 1^η, the adversary $\mathcal{B}$ does:

- First, it guesses randomly two indices i, j in $[\![1, p(\eta)]\!]$. If $i \geq j$, it aborts.
- Then, it simulates $\mathcal{A}$, forwarding its calls to the oracles $\mathcal{O}_{\vec{\mathsf{H}}(\cdot,\mathsf{k_m})}$, with two exceptions:
  - The j-th query u_j to the oracle $\mathcal{O}_{\mathsf{H}_1(\cdot,\mathsf{k_m})}$ is not forwarded. Instead, $\mathcal{B}$ sends to $\mathcal{A}$ the result of the i-th query u_i to the oracle $\mathcal{O}_{\mathsf{H}_1(\cdot,\mathsf{k_m})}$ (i.e. $\mathsf{H}_1(u_i, \mathsf{k_m})$).
  - If there is a $j+1$-th query to $\mathcal{O}_{\mathsf{H}_1(\cdot,\mathsf{k_m})}$, $\mathcal{B}$ aborts.
- Finally, $\mathcal{B}$ gets a pair (m_1, m_2) from $\mathcal{A}$. It checks whether $m_1 = u_i$ and $m_2 = u_j$. If not, it aborts. Otherwise, it returns $(u_j, \mathsf{H}_1(u_i, \mathsf{k_m}))$.

The probability of $\mathcal{B}$ winning the game is exactly the probability of $\mathcal{B}$ winning the game and not aborting. Moreover, if $\mathcal{B}$ does not abort, $\mathcal{A}$ output a pair (m_1, m_2) which it believes is a valid collision. Therefore $\mathcal{B}$ wins if and only if (m_1, m_2) is a valid collision. We use ρ_1 for $\mathcal{B}$ random tape, and ρ_2 for $\mathcal{A}$ random tape.[7] Then we can lower-bound the probability that $\mathcal{B}$ wins as follows:

$$
\begin{aligned}
& \mathbf{Pr}\big(\rho_1, \rho_2, \mathsf{k_m} \ : \ \mathcal{B}^{\mathcal{O}_{\vec{\mathsf{H}}(\cdot,\mathsf{k_m})}}(\rho_1, \rho_2) \text{ wins}\big) \\
= \ & \mathbf{Pr}\big(\rho_1, \rho_2, \mathsf{k_m} \ : \ \mathcal{B}^{\mathcal{O}_{\vec{\mathsf{H}}(\cdot,\mathsf{k_m})}}(\rho_1, \rho_2) \text{ wins} \wedge \neg\text{abort}(\mathcal{B})\big) \\
= \ & \mathbf{Pr}\big(\rho_1, \rho_2, \mathsf{k_m} \ : \ (m_1, m_2) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathcal{B}}(\rho_1,\mathsf{k_m})}(\rho_2) \wedge \mathsf{H}_1(m_1, \mathsf{k_m}) = \mathsf{H}_1(m_2, \mathsf{k_m}) \wedge \neg\text{abort}(\mathcal{B})\big) \\
\geq \ & \mathbf{Pr}\big(\rho_1, \rho_2, \mathsf{k_m} \ : \ (m_1, m_2) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathcal{B}}(\rho_1,\mathsf{k_m})}(\rho_2) \wedge \mathsf{H}_1(m_1, \mathsf{k_m}) = \mathsf{H}_1(m_2, \mathsf{k_m}) \wedge \neg\text{abort}(\mathcal{B}) \\
& \qquad\qquad \mid \mathcal{A}^{\mathcal{O}_{\vec{\mathsf{H}}(\cdot,\mathsf{k_m})}}(\rho_2) \text{ wins}\big) \times \mathbf{Pr}\big(\rho_2, \mathsf{k_m} : \mathcal{A}^{\mathcal{O}_{\vec{\mathsf{H}}(\cdot,\mathsf{k_m})}}(\rho_2) \text{ wins}\big)
\end{aligned}
$$

Knowing that $\mathcal{A}^{\mathcal{O}_{\vec{\mathsf{H}}(\cdot,\mathsf{k_m})}}(\rho_2)$ wins, the probability over ρ_2 that $\mathcal{B}$ correctly guessed the index of $\mathcal{A}^{\mathcal{O}_{\vec{\mathsf{H}}(\cdot,\mathsf{k_m})}}(\rho_2)$'s query of m_1 to the oracle and the number of $\mathcal{A}^{\mathcal{O}_{\vec{\mathsf{H}}(\cdot,\mathsf{k_m})}}(\rho_2)$'s queries is $\frac{1}{p(\eta)^2}$. Hence:

$$
\begin{aligned}
\geq \ & \mathbf{Pr}\big(\rho_1, \rho_2, \mathsf{k_m} \ : \ (m_1, m_2) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathcal{B}}(\rho_1,\mathsf{k_m})}(\rho_2) \wedge \mathsf{H}_1(m_1, \mathsf{k_m}) = \mathsf{H}_1(m_2, \mathsf{k_m}) \wedge \neg\text{abort}(\mathcal{B}) \\
& \qquad\qquad \mid \mathcal{A}^{\mathcal{O}_{\vec{\mathsf{H}}(\cdot,\mathsf{k_m})}}(\rho_2) \text{ wins} \wedge \mathsf{guessed}(\mathcal{B})\big) \times \mathbf{Pr}\big(\rho_2, \mathsf{k_m} : \mathcal{A}^{\mathcal{O}_{\vec{\mathsf{H}}(\cdot,\mathsf{k_m})}}(\rho_2) \text{ wins}\big) \times \frac{1}{p(\eta)^2}
\end{aligned}
$$

Knowing that $\mathcal{B}$ guessed properly, and that $\mathcal{A}^{\mathcal{O}_{\vec{\mathsf{H}}(\cdot,\mathsf{k_m})}}(\rho_2)$ wins, we know that the oracles $\mathcal{O}_{\mathsf{H}}(\mathsf{k_m})$ and $\mathcal{O}_{\mathcal{B}}(\rho_1, \mathsf{k_m})$ have the same outputs on $\mathcal{A}(\rho_2)$'s queries. By consequence:

$$
\begin{aligned}
\geq \ & \mathbf{Pr}\big(\rho_1, \rho_2, \mathsf{k_m} \ : \ (m_1, m_2) \leftarrow \mathcal{A}^{\mathcal{O}_{\vec{\mathsf{H}}(\cdot,\mathsf{k_m})}}(\rho_2) \wedge \mathsf{H}_1(m_1, \mathsf{k_m}) = \mathsf{H}_1(m_2, \mathsf{k_m}) \wedge \neg\text{abort}(\mathcal{B}) \\
& \qquad\qquad \mid \mathcal{A}^{\mathcal{O}_{\vec{\mathsf{H}}(\cdot,\mathsf{k_m})}}(\rho_2) \text{ wins} \wedge \mathsf{guessed}(\mathcal{B})\big) \times \mathbf{Pr}\big(\rho_2, \mathsf{k_m} : \mathcal{A}^{\mathcal{O}_{\vec{\mathsf{H}}(\cdot,\mathsf{k_m})}}(\rho_2) \text{ wins}\big) \times \frac{1}{p(\eta)^2}
\end{aligned}
$$

[7] Of course, $\mathcal{B}$ has access to ρ_2 since it simulates $\mathcal{A}$. But it only uses it for the simulation, not for its own coin tosses.

In that case, we know that $\mathcal{B}$ does not abort and that the game is won. Therefore:

$$\geq \mathbf{Pr}\big(\rho_2, \mathsf{k_m} : \mathcal{A}^{\mathcal{O}_{\vec{\mathsf{H}}(\cdot,\mathsf{k_m})}}(\rho_2) \text{ wins}\big) \times \frac{1}{p(\eta)^2}$$

Which, by hypothesis, is non-negligible. ■

4.7.3 Cryptographic Axioms

We translate these games in the logic for the two families of functions $(\mathsf{Mac}^j)_{1\leq j\leq 5}$ and $(\mathsf{f}, \mathsf{f}^{\mathsf{r}})$. As expected, these axioms are very similar to the axioms of Section 2.5 in Chapter 2. First, some definitions.

Definition 4.14. For every ground term u, we define three set of subterms of u:

- We let $\mathsf{set\text{-}mac}^j_{\mathsf{k_m}}(u)$ be the set of Mac^j terms under key $\mathsf{k_m}$ in u:

$$\mathsf{set\text{-}mac}^j_{\mathsf{k_m}}(u) = \{m \mid \mathsf{Mac}^j_{\mathsf{k_m}}(m) \in \mathsf{st}(u)\}$$

- We let $\mathsf{strict\text{-}set\text{-}mac}^j_{\mathsf{k_m}}(u)$ be the set of mac-ed terms under key $\mathsf{k_m}$ and tag j in u appearing outside a conditional:

$$\mathsf{strict\text{-}set\text{-}mac}^j_{\mathsf{k_m}}(u) = \{m \mid \mathsf{Mac}^j_{\mathsf{k_m}}(m) \in \mathsf{strict\text{-}st}(u)\}$$

- For every $g \in \{\mathsf{f}, \mathsf{f}^{\mathsf{r}}\}$, we let $\mathsf{set\text{-}prf}^g_{\mathsf{k}}(u)$ be the set of g terms under key k in u:

$$\mathsf{set\text{-}prf}^g_{\mathsf{k}}(u) = \{m \mid g_{\mathsf{k}}(m) \in \mathsf{st}(u)\}$$

The axioms are given in Figure 4.15, and are sound under the appropriate cryptographic assumptions.

Proposition 4.6. *The axioms in Figure 4.15 are valid in any computational model where:*

CR^j	$(\mathit{Mac}^i)_{1\leq i\leq 5}$ *are jointly* CR-HK
EUF-MAC^j, P-EUF-MAC^j *and* $\text{CR-KEY}^j_{\neq}$	$(\mathit{Mac}^i)_{1\leq i\leq 5}$ *are jointly* EUF-MAC
PRF-MAC^j	$(\mathit{Mac}^i)_{1\leq i\leq 5}$ *are jointly* PRF
PRF-f *and* PRF-f^{r}	(f, f^{r}) *are jointly* PRF

Proof. The proof are exactly the same than in Section 2.5 of Chapter 2. Therefore, we omit the details. ■

Remark 4.4. Similarly to what we observed in Remark 2.4 of Chapter 2, the following axiom schema is admissible using PRF-MAC^j + Trans:

$$\frac{\vec{u}, \mathsf{if} \bigvee_{i\in I} \mathsf{eq}(m, m_i) \mathsf{\ then\ 0\ else\ n} \sim \vec{v}}{\vec{u}, \mathsf{if} \bigvee_{i\in I} \mathsf{eq}(m, m_i) \mathsf{\ then\ 0\ else\ } \mathsf{Mac}^j_{\mathsf{k_m}}(m) \sim \vec{v}} \quad \text{when} \begin{cases} \mathsf{fresh}(\mathsf{n}; \vec{u}, m) \\ \mathsf{k_m} \sqsubseteq_{\mathsf{Mac}^{-}(\_)} \vec{u}, m \\ \{m_i \mid i \in I\} = \mathsf{set\text{-}mac}^j_{\mathsf{k_m}}(\vec{u}, m) \end{cases}$$

By a notation abuse, we refer also to the axiom above as PRF-MAC^j. The same remark applies to PRF-f and $\text{PRF-f}^{\mathsf{r}}$. □

Definition 4.15. We let $\mathsf{Ax_{crypto}}$ be the set of cryptographic axioms:

$$\mathsf{Ax_{crypto}} = \text{CCA}_1 \cup \big(\text{PRF-MAC}^j\big)_{1\leq j\leq 5} \cup \text{PRF-f} \cup \text{PRF-f}^{\mathsf{r}} \cup \big(\text{EUF-MAC}^j\big)_{1\leq j\leq 5} \cup \big(\text{CR}^j\big)_{1\leq j\leq 5}$$

Proposition 4.7. *The axioms in* $\mathsf{Ax_{crypto}}$ *are valid in any computational model where the asymmetric encryption* $\{\_\}^{\_}_{\_}$ *is* IND-CCA_1 *secure and* f *and* f^{r} *(resp.* Mac^1–Mac^5*) are jointly* PRF.

Proof. For CCA_1, this is from Propositions 2.6. For the other axioms, we know using Proposition 4.4 and Proposition 4.5 that f and f^{r} (resp. Mac^1–Mac^5) are jointly EUF-MAC and CR-HK. Therefore we can conclude using Proposition 4.6. ■

$$\overline{\mathsf{Mac}^j_{\mathsf{k_m}}(m_1) = \mathsf{Mac}^j_{\mathsf{k_m}}(m_2) \to m_1 = m_2} \quad \text{when } \mathsf{k_m} \sqsubseteq_{\mathsf{Mac}^{=}(\_)} m_1, m_2 \qquad (\mathrm{CR}^j)$$

$$\overline{s = \mathsf{Mac}^j_{\mathsf{k_m}}(m) \to \bigvee_{u \in S} s = \mathsf{Mac}^j_{\mathsf{k_m}}(u)} \quad \text{when } \begin{cases} \mathsf{k_m} \sqsubseteq_{\mathsf{Mac}^{=}(\_)} s, m \\ S = \mathsf{set\text{-}mac}^j_{\mathsf{k_m}}(s, m) \end{cases} \qquad (\mathrm{EUF\text{-}MAC}^j)$$

$$\overline{s = \mathsf{Mac}^j_{\mathsf{k_m}}(m) \to \bigvee_{i \in I} b_i \wedge \bigvee_{u \in S_i} s = \mathsf{Mac}^j_{\mathsf{k_m}}(u)} \quad \text{when } \begin{cases} \mathsf{k_m} \sqsubseteq_{\mathsf{Mac}^{=}(\_)} s, m \\ (b_i)_{i \in I} \text{ is a valid CS partition} \\ \exists (s_i, m_i)_{i \in I} \text{ s.t. } \forall i \in I \\ \quad [b_i]s_i \doteq [b_i]s \wedge [b_i]m_i \doteq [b_i]m \\ \quad S_i = \mathsf{strict\text{-}set\text{-}mac}^j_{\mathsf{k_m}}(s_i, m_i) \end{cases} \qquad (\mathrm{P\text{-}EUF\text{-}MAC}^j)$$

$$\overline{\mathsf{Mac}^j_{\mathsf{k_m}}(u) = \mathsf{Mac}^j_{\mathsf{k'_m}}(v) = \mathsf{false}} \quad \text{when } \begin{cases} \mathsf{k_m}, \mathsf{k'_m} \sqsubseteq_{\mathsf{Mac}^{=}(\_)} u, v \\ \mathsf{k_m}, \mathsf{k'_m} \in \mathcal{N} \end{cases} \qquad (\mathrm{CR\text{-}KEY}^j_{\neq})$$

$$\overline{\vec{u}, \mathsf{if} \bigvee_{i \in I} \mathsf{eq}(m, m_i) \mathsf{\ then\ } 0 \mathsf{\ else\ } \mathsf{Mac}^j_{\mathsf{k_m}}(m) \sim \vec{u}, \mathsf{if} \bigvee_{i \in I} \mathsf{eq}(m, m_i) \mathsf{\ then\ } 0 \mathsf{\ else\ } \mathsf{n}} \quad \text{when } \begin{cases} \mathsf{fresh}(\mathsf{n}; \vec{u}, m) \\ \mathsf{k_m} \sqsubseteq_{\mathsf{Mac}^{=}(\_)} \vec{u}, m \\ \{m_i \mid i \in I\} = \mathsf{set\text{-}mac}^j_{\mathsf{k_m}}(\vec{u}, m) \end{cases} \qquad (\mathrm{PRF\text{-}MAC}^j)$$

$$\overline{\vec{u}, \mathsf{if} \bigvee_{i \in I} \mathsf{eq}(m, m_i) \mathsf{\ then\ } 0 \mathsf{\ else\ } g_{\mathsf{k}}(m) \sim \vec{u}, \mathsf{if} \bigvee_{i \in I} \mathsf{eq}(m, m_i) \mathsf{\ then\ } 0 \mathsf{\ else\ } \mathsf{n}} \quad \text{when } \begin{cases} \mathsf{fresh}(\mathsf{n}; \vec{u}, m) \\ \mathsf{k} \sqsubseteq_{\mathsf{f}.(\_), \mathsf{f}^{\mathsf{r}}(\_)} \vec{u}, m \\ \{m_i \mid i \in I\} = \mathsf{set\text{-}prf}^g_{\mathsf{k}}(\vec{u}, m) \end{cases} \qquad (\mathrm{PRF\text{-}g})$$

Convention: $1 \leq j \leq 5$ and $g \in \{\mathsf{f}, \mathsf{f}^{\mathsf{r}}\}$.

Figure 4.15: Axioms for Joint Cryptographic Assumptions

4.7.4 Axioms

We define the set of axioms Ax we use to prove that the AKA$^+$ protocol provides mutual authentication and σ_{ul}-unlinkability. This set of axioms contains mostly axioms we presented in Chapter 2, plus some additional axioms which are specific to the AKA$^+$ protocol.

We define the set of constants $\mathcal{S}_{\mathsf{cst}}$, which contains the set of identities $\mathcal{S}^{\omega}_{\mathsf{id}}$, the integers 0 and 1, and the special values UnSet, $\mathsf{UnknownId}$, fail, defaut and error. This set does not include all the constants of the AKA$^+$, but only the ones whose interpretations must be distinct (this is enforced by an axiom later).

Definition 4.16. We define the set $\mathcal{S}_{\mathsf{cst}}$ of constant function symbols:

$$\mathcal{S}_{\mathsf{cst}} := \mathcal{S}^{\omega}_{\mathsf{id}} \cup \{\mathsf{UnSet}, \mathsf{UnknownId}, \mathsf{fail}, \mathsf{defaut}, \mathsf{error}, 0, 1\}$$

We now define the set of axioms Ax:

Definition 4.17. Ax is the set of axioms $\mathsf{Ax} = \mathsf{Ax}_{\mathsf{struct}} \cup \mathsf{Ax}_{\mathsf{impl}} \cup \mathsf{Ax}_{\mathsf{crypto}}$, where:

- $\mathsf{Ax}_{\mathsf{struct}}$ is the set of structural axioms, which are given in Figure 2.1 and Figure 2.2.
- $\mathsf{Ax}_{\mathsf{crypto}}$ is the set of cryptographic axioms in Figure 4.15, plus the CCA_1 axiom given in Section 2.6.1.
- $\mathsf{Ax}_{\mathsf{impl}}$ is the set of implementation axioms. It includes:
  - The axioms $\mathsf{Ax}_{\langle , \rangle}$, $\mathsf{Ax}_{\mathsf{dec}}$, $\mathsf{Ax}_{\oplus}$ and $\mathsf{Ax}_{\mathsf{bool}}$ from Section 2.5.2.
  - The new axioms in Figure 4.16, which we describe below.

Description The only new axioms are the axioms in Figure 4.16. We quickly describe them:

- *The set Ax_{eq} of equality and dis-equality axioms:*

$$\frac{}{\pi_i(\langle x_1\,,x_2\,,x_3\rangle) \doteq x_i} \text{ for } i \in \{1,2,3\} \qquad \frac{}{\mathsf{eq}(\mathsf{A},\mathsf{B}) \doteq \mathsf{false}}\ {\neq}\text{-Const} \text{ for every } \mathsf{A},\mathsf{B} \in \mathcal{S}_{\mathsf{cst}} \text{ s.t. } \mathsf{A} \not\equiv \mathsf{B}$$

- *The set Ax_{len} of length axioms:*

$$\frac{\mathsf{len}(u) \doteq \mathsf{len}(s) \qquad \mathsf{len}(v) \doteq \mathsf{len}(t)}{\mathsf{len}(\langle u\,,v\rangle) \doteq \mathsf{len}(\langle s\,,t\rangle)} \qquad \frac{}{\mathsf{len}(\mathrm{ID}_1) \doteq \mathsf{len}(\mathrm{ID}_2)} \text{ for every } \mathrm{ID}_1,\mathrm{ID}_2 \in \mathcal{S}^{\omega}_{\mathsf{id}}$$

$$\frac{}{\mathsf{len}(\mathsf{suc}(\mathsf{sqn\text{-}init}^{\mathrm{ID}}_{\mathrm{U}})) \doteq \mathsf{len}(\mathsf{sqn\text{-}init}^{\mathrm{ID}}_{\mathrm{U}})} \text{ for every } \mathrm{ID} \in \mathcal{S}^{\omega}_{\mathsf{id}}$$

$$\frac{}{\mathsf{len}(\mathsf{sqn\text{-}init}^{\mathrm{ID}_1}_{\mathrm{U}}) \doteq \mathsf{len}(\mathsf{sqn\text{-}init}^{\mathrm{ID}_2}_{\mathrm{U}})} \text{ for every } \mathrm{ID}_1,\mathrm{ID}_2 \in \mathcal{S}^{\omega}_{\mathsf{id}}$$

$$\frac{}{\mathsf{len}(\mathsf{sqn\text{-}init}^{\mathrm{ID}}_{\mathrm{U}}) \doteq \mathsf{len}(\mathsf{n})} \text{ for every } \mathrm{ID} \in \mathcal{S}^{\omega}_{\mathsf{id}}, \mathsf{n} \in \mathcal{N} \qquad \frac{}{\mathsf{len}(0^x) \doteq x} \qquad \frac{}{\mathsf{len}(1^x) \doteq x}$$

$$\frac{}{\mathsf{len}(\mathsf{x}) \not\doteq 0} \text{ when } \mathsf{x} \in \mathcal{S}_{\mathsf{cst}} \qquad \frac{\mathsf{len}(u) \not\doteq 0}{\mathsf{len}(\langle u\,,v\rangle) \not\doteq 0} \qquad \frac{\mathsf{len}(v) \not\doteq 0}{\mathsf{len}(\langle u\,,v\rangle) \not\doteq 0} \qquad \frac{A \not\equiv B \qquad \mathsf{len}(A) \not\doteq 0 \qquad x \not\doteq 0}{A^x \not\doteq B^y}\ \mathsf{l\text{-}neq}$$

- *The set Ax_{inj} of injectivity axioms:*

$$\frac{}{\neg\mathsf{eq}(u,s) \wedge \mathsf{eq}(\langle u\,,v\rangle,\langle s\,,t\rangle) \doteq \mathsf{false}}\ \mathsf{EQInj}(\langle\cdot\,,\_\rangle) \qquad \frac{}{\neg\mathsf{eq}(v,t) \wedge \mathsf{eq}(\langle u\,,v\rangle,\langle s\,,t\rangle) \doteq \mathsf{false}}\ \mathsf{EQInj}(\langle\_\,,\cdot\rangle)$$

$$\frac{}{\neg\mathsf{eq}(u,v) \wedge \mathsf{eq}(\{u\}^{\mathsf{n_e}}_{\mathsf{pk(n)}},\{v\}^{\mathsf{n'_e}}_{\mathsf{pk(n')}}) \doteq \mathsf{false}}\ \mathsf{EQInj}(\{\cdot\}_{\_})$$

- *The set $\mathsf{Ax}_{\mathrm{SQN}}$ of sequence number axioms:*

$$\frac{}{\mathsf{range}(u,v) \doteq \mathsf{eq}(u,v)} \qquad \frac{}{\mathsf{suc}(u) \doteq u+1} \qquad \frac{}{\mathsf{sqn\text{-}init}^{\mathrm{ID}}_{\mathrm{N}} \leq \mathsf{sqn\text{-}init}^{\mathrm{ID}}_{\mathrm{U}}}\ \mathrm{SQN}\text{-}\mathsf{ini} \text{ for every } \mathrm{ID} \in \mathcal{S}^{\omega}_{\mathsf{id}}$$

$$\frac{}{\phi[\vec{u}] \doteq \mathsf{true}} \text{ when } \vec{u} \text{ are ground terms and } \mathrm{Th}(\mathbb{Z},0,1,+,-,=,\leq) \models \phi[\vec{x}]$$

Figure 4.16: The Set of Axiom $\mathsf{Ax}_{\mathsf{impl}} = \mathsf{Ax}_{\mathsf{ite}} \cup \mathsf{Ax}_{\mathsf{eq}} \cup \mathsf{Ax}_{\mathsf{len}} \cup \mathsf{Ax}_{\mathsf{inj}} \cup \mathsf{Ax}_{\mathrm{SQN}}$.

- The set $\mathsf{Ax}_{\mathsf{eq}}$ contains additional axioms satisfied by the equality function symbol $\mathsf{eq}(\_,\_)$. It includes properties of triples and projections, and dis-equality axioms for the element of $\mathcal{S}_{\mathsf{cst}}$. The dis-equality axioms require that all the elements of $\mathcal{S}_{\mathsf{cst}}$ must be interpreted by distinct bit-strings:

$$\frac{}{\mathsf{eq}(\mathsf{A},\mathsf{B}) \doteq \mathsf{false}}\ {\neq}\text{-Const} \qquad \text{for every } \mathsf{A},\mathsf{B} \in \mathcal{S}_{\mathsf{cst}} \text{ s.t. } \mathsf{A} \not\equiv \mathsf{B}$$

- $\mathsf{Ax}_{\mathsf{len}}$ is the set of implementation axioms on the length function $\mathsf{len}(\_)$. In particular, all identities in $\mathcal{S}^{\omega}_{\mathsf{id}}$ must have the same lengths, and not be of length 0. Similarly, sequence numbers must have the same lengths. There are also some axioms to reason on lengths, e.g.:

$$\frac{\mathsf{len}(u) \doteq \mathsf{len}(s) \qquad \mathsf{len}(v) \doteq \mathsf{len}(t)}{\mathsf{len}(\langle u\,,v\rangle) \doteq \mathsf{len}(\langle s\,,t\rangle)} \qquad \frac{\mathsf{len}(u) \not\doteq 0}{\mathsf{len}(\langle u\,,v\rangle) \not\doteq 0} \qquad \frac{\mathsf{len}(v) \not\doteq 0}{\mathsf{len}(\langle u\,,v\rangle) \not\doteq 0}$$

- The set $\mathsf{Ax}_{\mathsf{inj}}$ contains injectivity axioms for the pair and encryption. For example, for the pair, we have the left injectivity axioms:

$$\frac{}{\neg\mathsf{eq}(u,s) \wedge \mathsf{eq}(\langle u\,,v\rangle,\langle s\,,t\rangle) \doteq \mathsf{false}}\ \mathsf{EQInj}(\langle\cdot\,,\_\rangle)$$

- The set $\mathsf{Ax}_{\mathrm{SQN}}$ contains sequence numbers axioms. In particular, it requires that:

– The range and successor functions are, resp., an equality check and a by-one increment:

$$\overline{\mathsf{range}(u,v) \doteq \mathsf{eq}(u,v)} \qquad \overline{\mathsf{suc}(u) \doteq u+1}$$

– Initially, the *HN* sequence number is no larger than the *UE* sequence number.

$$\overline{\mathsf{sqn\text{-}init}_{\mathrm{N}}^{\mathrm{ID}} \leq \mathsf{sqn\text{-}init}_{\mathrm{U}}^{\mathrm{ID}}}\ \text{SQN-ini} \qquad \text{for every } \mathrm{ID} \in \mathcal{S}_{\mathsf{id}}^{\omega}$$

– For any term $\phi[]$ encoding of a boolean formula, if $\phi[\vec{u}]$ is valid in the first-order theory $\mathrm{Th}(\mathbb{Z},0,1,+,-,=,\leq)$ then $\phi[\vec{u}] \doteq \mathsf{true}$ is a valid axiom.

$$\overline{\phi[\vec{u}] \doteq \mathsf{true}} \qquad \text{when } \vec{u} \text{ are ground terms and } \mathrm{Th}(\mathbb{Z},0,1,+,-,=,\leq) \models \phi[\vec{x}\,]$$

Notations In the rest of this chapter, the set of axioms Ax is fixed, and we stop to specify that we use it: we say that we have a derivation of a formula ϕ to mean that ϕ can be deduced from Ax. Furthermore, we say that ϕ holds when there is a derivation of ϕ.

Moreover, we abuse notations and write $u = v$ instead of $u \doteq v$. We can always disambiguate using the context: if we expect a term, then $u = v$ stands for the term $\mathsf{eq}(u,v)$, whereas if a formula is expected then $u = v$ stands for $\mathsf{eq}(u,v) \sim \mathsf{true}$. We extends this to any boolean term: if b is a boolean term then we say that b holds if we can show that $b \sim \mathsf{true}$ holds. For example, $\sigma_\tau(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) \geq \sigma_\tau(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}})$ holds if we can show that $\mathsf{geq}(\sigma_\tau(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}), \sigma_\tau(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}})) \sim \mathsf{true}$.

4.7.5 ⋆ (p. 91) Additional Axioms

We present additional axioms, and show that they are logical consequences of the axioms Ax.

Definition 4.18. We let Simp denote a sequence of applications of R, FA and Dup, i.e.:

$$\frac{\vec{s} \sim \vec{t}}{\vec{u} \sim \vec{v}}\ \mathsf{Simp} \quad \text{when} \quad \frac{\vec{s} \sim \vec{t}}{\vec{u} \sim \vec{v}}\ (R + \mathsf{FA} + \mathsf{Dup})^*$$

Definition 4.19 (The indep-branch Axioms)**.** Let $\vec{u}, \vec{b}$ be ground terms, $C[]$ an if-context and $\mathsf{n}, (\mathsf{n}_i)_{i\in I}$ nonces. If $\mathsf{n}, (\mathsf{n}_i)_{i\in I}$ are distinct and such that $\mathsf{fresh}(\mathsf{n}, (\mathsf{n}_i)_{i\in I}; \vec{u}, \vec{b}, C[])$, then the following inference rule is an instance of the indep-branch axiom:

$$\frac{}{\vec{u}, C\big[\vec{b} \diamond (\mathsf{n}_i)_{i\in I}\big] \sim \vec{u}, \mathsf{n}}\ \mathsf{indep\text{-}branch}$$

Proposition 4.8. *The indep-branch axioms are a consequence of the Ax axioms.*

Proof. To prove this, we first introduce the if-context $C[]$ on the right to match the shape of the left side. We then split the proof using CS, and conclude by applying Fresh. This yields the derivation:

$$\dfrac{\dfrac{\dfrac{}{\forall i \in I, \vec{u}, \vec{b}, \mathsf{n}_i \sim \vec{u}, \vec{b}, \mathsf{n}}\ \mathsf{Fresh}}{\vec{u}, C\big[\vec{b} \diamond (\mathsf{n}_i)_{i\in I}\big] \sim \vec{u}, C\big[\vec{b} \diamond (\mathsf{n})_{i\in I}\big]}\ \mathsf{CS}^*}{\vec{u}, C\big[\vec{b} \diamond (\mathsf{n}_i)_{i\in I}\big] \sim \vec{u}, \mathsf{n}}\ R$$

■

It is often convenient to apply the FA axiom under an if-context C.

Definition 4.20. Let $\vec{v}, \vec{b}, (u_{i,j})_{i\in I, 1\leq j\leq n}, (u'_{i,j})_{i\in I, 1\leq j\leq n}$ be ground terms and C an if-context. Then the following inference rule is an instance of the $\mathsf{FA_c}$ axiom:

$$\frac{\vec{v}, \big(C\big[\vec{b} \diamond (u_{i,j})_{i\in I}\big]\big)_{1\leq j\leq n} \sim \vec{v}', \big(C\big[\vec{b}' \diamond (u'_{i,j})_{i\in I}\big]\big)_{1\leq j\leq n}}{\vec{v}, C\big[\vec{b} \diamond \big(f((u_{i,j})_{1\leq j\leq n})\big)_{i\in I}\big] \sim \vec{v}', C\big[\vec{b}' \diamond \big(f((u'_{i,j})_{1\leq j\leq n})\big)_{i\in I}\big]}\ \mathsf{FA_c}$$

Proposition 4.9. *The $\mathsf{FA_c}$ axioms are a consequence of the Ax axioms.*

Proof. First, we pull the f function outside of the if-context C using the homomorphism properties of the if_then_else_. Finally we apply the FA axiom. This yields the derivation:

$$\dfrac{\dfrac{\vec{v}, \left(C\left[\vec{b} \diamond (u_{i,j})_{i\in I}\right]\right)_{1\le j\le n} \sim \vec{v}', \left(C\left[\vec{b}' \diamond (u'_{i,j})_{i\in I}\right]\right)_{1\le j\le n}}{\vec{v}, f\left(C\left[\vec{b} \diamond (u_{i,j})_{i\in I}\right]\right)_{1\le j\le n} \sim \vec{v}', f\left(C\left[\vec{b}' \diamond (u'_{i,j})_{i\in I}\right]\right)_{1\le j\le n}}\ \mathsf{FA}}{\vec{v}, C\left[\vec{b} \diamond \left(f((u_{i,j})_{1\le j\le n})\right)_{i\in I}\right] \sim \vec{v}', C\left[\vec{b}' \diamond \left(f((u'_{i,j})_{1\le j\le n})\right)_{i\in I}\right]}\ R$$

■

Finally, the following axioms state that two encryptions with different randomness are almost never equal. This requires that the encrypted messages are not of length zero.

Proposition 4.10. *For every ground terms u, v, the following axiom is a consequence of the* Ax *axioms:*

$$\dfrac{\mathit{len}(u) \doteq \mathit{len}(v) \qquad \mathit{len}(u) \not\doteq 0}{\mathit{eq}(\{u\}^{n_e}_{\mathit{pk}(n)}, \{v\}^{n'_e}_{\mathit{pk}(n)}) \doteq \mathit{false}} \quad \mathit{when} \begin{cases} n_e \not\equiv n'_e \\ \mathit{fresh}(n_e, n'_e; u, v) \\ n \sqsubseteq_{\mathit{pk}(\cdot),\mathit{sk}(\cdot)} u, v \ \wedge\ \mathit{sk}(n) \sqsubseteq_{\mathit{dec}(\_,\cdot)} u, v \end{cases}$$

Proof. We give directly the derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\mathsf{pk(n)}, \{u\}^{n_e}_{\mathsf{pk(n)}}, \mathsf{len}(v) \sim \mathsf{pk(n)}, \{u\}^{n_e}_{\mathsf{pk(n)}}, \mathsf{len}(v)}\ \mathsf{Refl} \qquad \dfrac{\dfrac{}{\mathsf{len}(v) \doteq \mathsf{len}(v)}\ \mathsf{Refl}}{\mathsf{len}(v) \doteq \mathsf{len}(1^{\mathsf{len}(v)})}}{\mathsf{pk(n)}, \{u\}^{n_e}_{\mathsf{pk(n)}}, \{v\}^{n'_e}_{\mathsf{pk(n)}} \sim \mathsf{pk(n)}, \{u\}^{n_e}_{\mathsf{pk(n)}}, \{1^{\mathsf{len}(v)}\}^{n'_e}_{\mathsf{pk(n)}}}\ \mathsf{CCA_1}}{\{u\}^{n_e}_{\mathsf{pk(n)}}, \{v\}^{n'_e}_{\mathsf{pk(n)}} \sim \{u\}^{n_e}_{\mathsf{pk(n)}}, \{1^{\mathsf{len}(v)}\}^{n'_e}_{\mathsf{pk(n)}}}\ \mathsf{Restr}}{\mathsf{eq}(\{u\}^{n_e}_{\mathsf{pk(n)}}, \{v\}^{n'_e}_{\mathsf{pk(n)}}) \doteq \mathsf{eq}(\{u\}^{n_e}_{\mathsf{pk(n)}}, \{1^{\mathsf{len}(v)}\}^{n'_e}_{\mathsf{pk(n)}})}\ \mathsf{FA} \qquad \mathsf{eq}(\{u\}^{n_e}_{\mathsf{pk(n)}}, \{1^{\mathsf{len}(v)}\}^{n'_e}_{\mathsf{pk(n)}}) \doteq \mathsf{false}}{\mathsf{eq}(\{u\}^{n_e}_{\mathsf{pk(n)}}, \{v\}^{n'_e}_{\mathsf{pk(n)}}) \doteq \mathsf{false}}\ \mathsf{Trans}$$

To show $\mathsf{eq}(\{u\}^{n_e}_{\mathsf{pk(n)}}, \{1^{\mathsf{len}(v)}\}^{n'_e}_{\mathsf{pk(n)}}) \doteq \mathsf{false}$, we use the transitivity axiom again:

$$\dfrac{\mathsf{eq}(\{u\}^{n_e}_{\mathsf{pk(n)}}, \{1^{\mathsf{len}(v)}\}^{n'_e}_{\mathsf{pk(n)}}) \doteq \mathsf{eq}(\{0^{\mathsf{len}(u)}\}^{n_e}_{\mathsf{pk(n)}}, \{1^{\mathsf{len}(v)}\}^{n'_e}_{\mathsf{pk(n)}}) \qquad \mathsf{eq}(\{0^{\mathsf{len}(u)}\}^{n_e}_{\mathsf{pk(n)}}, \{1^{\mathsf{len}(v)}\}^{n'_e}_{\mathsf{pk(n)}}) \doteq \mathsf{false}}{\mathsf{eq}(\{u\}^{n_e}_{\mathsf{pk(n)}}, \{1^{\mathsf{len}(v)}\}^{n'_e}_{\mathsf{pk(n)}}) \doteq \mathsf{false}}\ \mathsf{Trans}$$

Now, we give the derivation of the left premise:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\mathsf{pk(n)}, \{1^{\mathsf{len}(v)}\}^{n'_e}_{\mathsf{pk(n)}}, \mathsf{len}(u) \sim \mathsf{pk(n)}, \{1^{\mathsf{len}(v)}\}^{n'_e}_{\mathsf{pk(n)}}, \mathsf{len}(u)}\ \mathsf{Refl} \qquad \dfrac{\dfrac{}{\mathsf{len}(u) \doteq \mathsf{len}(u)}\ \mathsf{Refl}}{\mathsf{len}(u) \doteq \mathsf{len}(0^{\mathsf{len}(u)})}}{\mathsf{pk(n)}, \{u\}^{n_e}_{\mathsf{pk(n)}}\{1^{\mathsf{len}(v)}\}^{n'_e}_{\mathsf{pk(n)}} \sim \mathsf{pk(n)}, \{0^{\mathsf{len}(u)}\}^{n_e}_{\mathsf{pk(n)}}\{1^{\mathsf{len}(v)}\}^{n'_e}_{\mathsf{pk(n)}}}\ \mathsf{CCA_1}}{\{u\}^{n_e}_{\mathsf{pk(n)}}\{1^{\mathsf{len}(v)}\}^{n'_e}_{\mathsf{pk(n)}} \sim \{0^{\mathsf{len}(u)}\}^{n_e}_{\mathsf{pk(n)}}\{1^{\mathsf{len}(v)}\}^{n'_e}_{\mathsf{pk(n)}}}\ \mathsf{Restr}}{\mathsf{eq}(\{u\}^{n_e}_{\mathsf{pk(n)}}, \{1^{\mathsf{len}(v)}\}^{n'_e}_{\mathsf{pk(n)}}) \doteq \mathsf{eq}(\{0^{\mathsf{len}(u)}\}^{n_e}_{\mathsf{pk(n)}}, \{1^{\mathsf{len}(v)}\}^{n'_e}_{\mathsf{pk(n)}})}\ \mathsf{FA}$$

And finally we prove the right premise $\mathsf{eq}(\{0^{\mathsf{len}(u)}\}^{n_e}_{\mathsf{pk(n)}}, \{1^{\mathsf{len}(v)}\}^{n'_e}_{\mathsf{pk(n)}}) \doteq \mathsf{false}$:

$$\dfrac{\dfrac{\dfrac{}{\mathsf{eq}(0,1) \doteq \mathsf{false}}\ \neq\text{-Const} \qquad \dfrac{}{\mathsf{len}(0) \not\doteq 0} \qquad \mathsf{len}(u) \not\doteq 0}{\mathsf{eq}(0^{\mathsf{len}(u)}, 1^{\mathsf{len}(v)}) \doteq \mathsf{false}}\ \mathsf{I\text{-}neq}}{\mathsf{eq}(\{0^{\mathsf{len}(u)}\}^{n_e}_{\mathsf{pk(n)}}, \{1^{\mathsf{len}(v)}\}^{n'_e}_{\mathsf{pk(n)}}) \doteq \mathsf{false}}\ \mathsf{EQInj}(\{\cdot\}^{\_}_{\_}) + R$$

■

4.8 Security of the AKA$^+$ Protocol

We now state the authentication and σ_{ul}-unlinkability lemmas, and sketch the proofs. The full proofs are given later, in Sections 4.9, 4.10 and 4.11.

4.8.1 Mutual Authentication of the AKA$^+$ Protocol

Authentication is modeled by a correspondence property [WL93] of the form "in any execution, if event A occurs, then event B occurred". This can be translated in the Bana-Comon indistinguishability logic.

Authentication of the User by the Network AKA$^+$ guarantees authentication of the user by the network if in any execution, if $HN(j)$ believes it authenticated UE_{ID}, then UE_{ID} stated earlier that it had initiated the protocol with $HN(j)$.

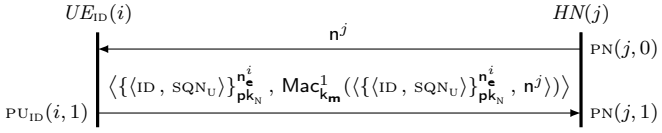
We recall that $\mathsf{e\text{-}auth}_{\text{N}}^{j}$ stores the identity of the UE authenticated by $HN(j)$, and that UE_{ID} stores in $\mathsf{b\text{-}auth}_{\text{U}}^{\text{ID}}$ the random challenge it received. Moreover, the session $HN(j)$ is uniquely identified by its random challenge n^j. Therefore, authentication of the user by the network is modeled by stating that, for any valid action trace τ, if $\sigma_\tau(\mathsf{e\text{-}auth}_{\text{N}}^{j}) = \text{ID}$ then there exists some prefix τ' of τ such that $\sigma_{\tau'}(\mathsf{b\text{-}auth}_{\text{U}}^{\text{ID}}) = \mathsf{n}^j$. Let $\preceq$ be the prefix ordering on action traces, then:

Lemma 4.1. *For every valid trace τ on $\mathcal{S}_{id}$, $\text{ID} \in \mathcal{S}_{id}$ and $j \in \mathbb{N}$, we have:*

$$\sigma_\tau(\mathit{e\text{-}auth}_{\text{N}}^{j}) = \text{ID} \rightarrow \bigvee_{\tau' \preceq \tau} \sigma_{\tau'}(\mathit{b\text{-}auth}_{\text{U}}^{\text{ID}}) = n^j$$

The key ingredients to show this lemma are *necessary conditions* for a message to be accepted by the network. Basically, a message can be accepted only if it was honestly generated by a subscriber. These necessary conditions rely on the unforgeability and collision-resistance of $(\mathsf{Mac}^j)_{1\leq j\leq 5}$.

Necessary Acceptance Conditions Using the EUF-MAC^j and CR^j axioms, we can find necessary conditions for a message to be accepted. We illustrate this on the *HN*'s second message in the SUPI sub-protocol. We depict the beginning of the execution between session $UE_{\text{ID}}(i)$ and session $HN(j)$ below:



We then prove that if a message is accepted by $HN(j)$ as coming from UE_{ID}, then the first component of this message must have been honestly generated by a session of UE_{ID}. Moreover, we know that this session received the challenge n^j.

Lemma 4.2. *Let $\text{ID} \in \mathcal{S}_{id}$ and τ be a valid trace on $\mathcal{S}_{id}$ ending with $\text{PN}(j,1)$. Then:*

$$\mathsf{accept}_\tau^{\text{ID}} \rightarrow \bigvee_{\tau_1 = \_, \text{PU}_{\text{ID}}(\_,1) \preceq \tau} \left(\pi_1(g(\phi_\tau^{\mathsf{in}})) = t_{\tau_1}^{\mathsf{enc}} \wedge g(\phi_{\tau_1}^{\mathsf{in}}) = n^j\right)$$

Proof sktech. Let t_{dec} be the term $\mathsf{dec}(\pi_1(g(\phi_\tau^{\mathsf{in}})), \mathsf{sk}_{\text{N}})$. Then $HN(j)$ accepts the last message iff the following test succeeds:

$$\underline{\pi_2(g(\phi_\tau^{\mathsf{in}})) = \mathsf{Mac}^1_{\mathsf{k_m^{ID}}}(\langle \pi_1(g(\phi_\tau^{\mathsf{in}})), \mathsf{n}^j\rangle)} \wedge \pi_1(t_{\mathsf{dec}}) = \text{ID}$$

By applying EUF-MAC^1 to the underlined part above, we know that if the test holds then $\pi_2(g(\phi_\tau^{\mathsf{in}}))$ is equal to one of the honest $\mathsf{Mac}^1_{\mathsf{k_m^{ID}}}$ subterms of $\pi_2(g(\phi_\tau^{\mathsf{in}}))$, which are the terms:

$$\left(\mathsf{Mac}^1_{\mathsf{k_m^{ID}}}(\langle t_{\tau_1}^{\mathsf{enc}}, g(\phi_{\tau_1}^{\mathsf{in}})\rangle)\right)_{\tau_1 = \_, \text{PU}_{\text{ID}}(\_,1) \prec \tau} \tag{4.2}$$

$$\left(\mathsf{Mac}^1_{\mathsf{k_m^{ID}}}(\langle \pi_1(g(\phi_{\tau_1}^{\mathsf{in}})), \mathsf{n}^{j_1}\rangle)\right)_{\tau_1 = \_, \text{PN}(j_1,1) \prec \tau} \tag{4.3}$$

Where $\prec$ is the strict version of $\preceq$. We know that $\text{PN}(j,1)$ cannot appear twice in τ. Hence for every $\tau_1 = \_, \text{PN}(j_1,1) \prec \tau$, we know that $j_1 \neq j$. Since distinct nonces are never equal, except for a negligible number of samplings, we derive that $\mathsf{eq}(\mathsf{n}^{j_1}, \mathsf{n}^j) = \mathsf{false}$. Using an axiom stating that the pair is injective and the CR^1 axiom, we can show that $\pi_2(g(\phi_\tau^{\mathsf{in}}))$ cannot by equal to one of the terms in (4.3).

Finally, for every $\tau_1 = \_, \text{PU}_{\text{ID}}(\_,1) \prec \tau$, using CR^1 and the pair injectivity axioms we derive that:

$$\left(\mathsf{Mac}^1_{\mathsf{k_m^{ID}}}(\langle \pi_1(g(\phi_\tau^{\mathsf{in}})), \mathsf{n}^j\rangle) = \mathsf{Mac}^1_{\mathsf{k_m^{ID}}}(\langle t_{\tau_1}^{\mathsf{enc}}, g(\phi_{\tau_1}^{\mathsf{in}})\rangle)\right) \rightarrow \pi_1(g(\phi_\tau^{\mathsf{in}})) = t_{\tau_1}^{\mathsf{enc}} \wedge \mathsf{n}^j = g(\phi_{\tau_1}^{\mathsf{in}})$$ ■

We prove a similar lemma for $\text{TN}(j,1)$. Lemma 4.1's proof is straightforward using these two properties.

Authentication of the Network by the User The AKA$^+$ protocol also provides authentication of the network by the user. That is, in any execution, if UE_{ID} believes it authenticated session $HN(j)$ then $HN(j)$ stated that it had initiated the protocol with UE_{ID}. Formally:

Lemma 4.3. *For every valid trace τ on $\mathcal{S}_{id}$, $\text{ID} \in \mathcal{S}_{id}$ and $j \in \mathbb{N}$, we have:*

$$\sigma_\tau(\mathit{e\text{-}auth}^{\text{ID}}_{\text{U}}) = \mathsf{n}^j \;\rightarrow\; \bigvee_{\tau' \preceq \tau} \sigma_{\tau'}(\mathit{b\text{-}auth}^j_{\text{N}}) = \text{ID}$$

This is shown using the same techniques than for Lemma 4.1.

4.8.2 σ-Unlinkability of the AKA$^+$ Protocol

Lemma 4.2 gives a necessary condition for a message to be accepted by $\text{PN}(j,1)$ as coming from ID. We can actually go further, and show that a message is accepted by $\text{PN}(j,1)$ as coming from ID *if and only if* it was honestly generated by a session of UE_{ID} which received the challenge n^j.

Lemma 4.4. *Let $\text{ID} \in \mathcal{S}_{id}$ and τ be a valid trace ending with $\text{PN}(j,1)$. There exists a derivation of:*

$$\mathsf{accept}^{\text{ID}}_\tau \leftrightarrow \bigvee_{\tau_1 = \_, \text{PU}_{\text{ID}}(\_,1) \preceq \tau} \left(g(\phi^{\mathsf{in}}_\tau) = t_{\tau_1} \wedge g(\phi^{\mathsf{in}}_{\tau_1}) = \mathsf{n}^j \right)$$

We prove similar lemmas for most actions of the AKA$^+$ protocol. Basically, these lemmas state that a message is accepted if and only if it is part of an honest execution of the protocol between UE_{ID} and HN. This allow us to replace each acceptance conditional $\mathsf{accept}^{\text{ID}}_\tau$ by a disjunction over all possible honest partial transcripts of the protocol.

We now state the σ_{ul}-unlinkability lemma:

Lemma 4.5. *The AKA^+_N protocol is σ_{ul}-unlinkable in any computational model satisfying the axioms Ax.*

Proof sktech. Using Proposition 4.3, we only need to show that for every valid basic action trace τ, there exists a derivation of $\phi_\tau \sim \phi_{\underline{\tau}}$ (where the left frame is a frame of the AKA^+_N protocol, and the right frame of the $\text{AKA}^+_{\underline{N}}$ protocol for $\underline{N}$ large enough). The full proof is long and technical, and is by induction on τ. Take a valid action trace τ, we assume by induction that there is a derivation of $\phi^{\mathsf{in}}_\tau \sim \phi^{\mathsf{in}}_{\underline{\tau}}$. We want to build a derivation of $\phi^{\mathsf{in}}_\tau, t_{\underline{\tau}} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, t_{\underline{\tau}}$ using the inference rules in Ax.

First, we rewrite t_τ using acceptance characterization lemmas, such as Lemma 4.4. This replaces each $\mathsf{accept}^{\text{ID}}_\tau$ by a case disjunction over all honest executions *on the left side*. Similarly, we rewrite $t_{\underline{\tau}}$ as a case disjunction over honest executions *on the right side*. Our goal is then to find a matching between left and right transcripts such that matched transcripts are indistinguishable. If a left and right transcript correspond to the same trace of oracle calls, this is easy. But since the left and right traces of oracle calls may differ, this is not always possible. E.g., some left transcript may not have a corresponding right transcript. When this happens, we have two possibilities: instead of a one-to-one match we build a many-to-one match, e.g. matching a left transcript to several right transcripts; or we show that some transcripts always result in a failure of the protocol. Showing the latter is complicated, as it requires to precisely track the possible values of $\text{SQN}^{\text{ID}}_{\text{U}}$ and $\text{SQN}^{\text{ID}}_{\text{N}}$ across multiple sessions of the protocol to prove that some transcripts always yield a de-synchronization between UE_{ID} and HN. ■

4.9 Mutual Authentication of the AKA$^+$ Protocol

We now prove that the AKA$^+$ protocol provides mutual authentication. This section is organized as follows: we state some useful properties and necessary acceptance conditions in Section 4.9.1 (we postpone the proofs of the necessary acceptance conditions to Section 4.9.5); then, we prove authentication of the user by the network in Section 4.9.2, and authentication of the network by the user in Section 4.9.3; finally, we prove that we actually have *injective* authentication of the network by the user in Section 4.9.4.

4.9.1 Invariants and Necessary Acceptance Conditions

We start by proving some properties of the AKA$^+$ protocol. First, we show that the sequence numbers are always of the same length. This is an easy consequence of the length axioms.

Proposition 4.11. *For every valid action traces τ, τ' on $\mathcal{S}_{id}$, $\mathrm{ID}_1, \mathrm{ID}_2 \in \mathcal{S}_{id}$ and $n \in \mathcal{N}$:*

$$\mathit{len}(\sigma^{\mathit{in}}_{\tau}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}_1})) = \mathit{len}(\sigma^{\mathit{in}}_{\tau}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}_2})) \qquad \mathit{len}(\sigma^{\mathit{in}}_{\tau}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}_1})) = \mathit{len}(\sigma^{\mathit{in}}_{\tau'}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}_1})) \qquad \mathit{len}(\sigma^{\mathit{in}}_{\tau}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}_1})) = \mathit{len}(n)$$

Proof. We show only the first equality, as the proofs of the other two equalities are similar. First, we prove by induction over τ that for every $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$, there exists an if-context C, terms $\vec{b}$ and integers $(k_i)_i$ such that:

$$\sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) = C[\vec{b} \diamond (\mathsf{suc}^{k_i}(\mathsf{sqn\text{-}init}_{\mathrm{U}}^{\mathrm{ID}}))_i)]$$

Therefore, let C_1, C_2, $\vec{b}_1, \vec{b}_2$ and $(k_i^1)_i, (k_j^2)_j$ be such that:

$$\sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}_1}) = C_1[\vec{b}_1 \diamond (\mathsf{suc}^{k_i^1}(\mathsf{sqn\text{-}init}_{\mathrm{U}}^{\mathrm{ID}_1}))_i] \qquad \sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}_2})) = C_2[\vec{b}_2 \diamond (\mathsf{suc}^{k_j^2}(\mathsf{sqn\text{-}init}_{\mathrm{U}}^{\mathrm{ID}_2}))_j]$$

Using the axioms in $\mathsf{Ax}_{\mathsf{len}}$, we show that for every i, i', j, j':

$$\mathsf{len}(\mathsf{suc}^{k_i^1}(\mathsf{sqn\text{-}init}_{\mathrm{U}}^{\mathrm{ID}_1})) = \mathsf{len}(\mathsf{suc}^{k_{i'}^1}(\mathsf{sqn\text{-}init}_{\mathrm{U}}^{\mathrm{ID}_1})) = \mathsf{len}(\mathsf{suc}^{k_j^2}(\mathsf{sqn\text{-}init}_{\mathrm{U}}^{\mathrm{ID}_2})) = \mathsf{len}(\mathsf{suc}^{k_{j'}^2}(\mathsf{sqn\text{-}init}_{\mathrm{U}}^{\mathrm{ID}_2}))$$

Therefore, using R we have a derivation of:

$$\mathsf{len}(\sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}_1})) = \mathsf{len}(\sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}_2}))$$ ■

The following proposition states that n_{N} appears only in the *HN* public key $\mathsf{pk}(\mathsf{n}_{\mathrm{N}})$ and secret key $\mathsf{sk}(\mathsf{n}_{\mathrm{N}})$, and that for every $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$, the keys k^{ID} and $\mathsf{k}_{\mathsf{m}}^{\mathrm{ID}}$ appear only in key position in Mac^1–Mac^5. These properties will be useful to apply the cryptographic axioms later.

Proposition 4.12 (Invariant (INV-KEY))**.** *For all valid action trace τ on $\mathcal{S}_{id} = \{\mathrm{ID}_1, \ldots, \mathrm{ID}_N\}$, we have:*

$$\begin{aligned} & n_{\mathrm{N}} \sqsubseteq_{pk(\cdot),sk(\cdot)} \phi_\tau \ \wedge\ sk(n_{\mathrm{N}}) \sqsubseteq_{dec(\_,\cdot)} \phi_\tau \\ \forall 1 \le i \le N, \quad & k_m^{\mathrm{ID}_i} \sqsubseteq_{Mac_{\cdot}(\_)} \phi_\tau \\ \forall 1 \le i \le N, \quad & k^{\mathrm{ID}_i} \sqsubseteq_{f_{\cdot}(\_),fr_{\cdot}(\_)} \phi_\tau \end{aligned}$$

Proof. The proof is straightforward by induction on τ. ■

The following proposition states that if a user ID has no valid temporary identity at instant τ_2 (i.e. $\sigma_{\tau_2}(\mathrm{GUTI}_{\mathrm{U}}^{\mathrm{ID}}) = \mathsf{UnSet}$), and if every ASSIGN-GUTI sub-protocol session run by ID between the instants τ_2 and τ_i failed (i.e. for every $\tau_1 = \_, \mathrm{FU}_{\mathrm{ID}}(j_1)$ such that $\tau_2 \prec_\tau \tau_1 \prec_\tau \tau_i$, we have $\neg\mathsf{accept}_{\tau_1}^{\mathrm{ID}}$), then ID does not have a valid temporary identity at instant τ_i (i.e. $\sigma^{\mathsf{in}}_{\tau_i}(\mathrm{GUTI}_{\mathrm{U}}^{\mathrm{ID}}) = \mathsf{UnSet}$). Formally:

Proposition 4.13. *For every valid action trace τ on $\mathcal{S}_{id}$, for every $\tau_2 \prec_\tau \tau_i$ and $\mathrm{ID} \in \mathcal{S}_{id}$, we have:*

$$\sigma_{\tau_2}(\mathrm{GUTI}_{\mathrm{U}}^{\mathrm{ID}}) = \mathit{UnSet} \wedge \bigwedge_{\substack{\tau_1 = \_, \mathrm{FU}_{\mathrm{ID}}(j_1) \\ \tau_2 \prec_\tau \tau_1 \prec_\tau \tau_i}} \neg\mathsf{accept}_{\tau_1}^{\mathrm{ID}} \rightarrow \sigma^{\mathit{in}}_{\tau_i}(\mathrm{GUTI}_{\mathrm{U}}^{\mathrm{ID}}) = \mathit{UnSet}$$

Proof. The proof is straightforward by induction on τ_i. ■

We let $\mathsf{session}$ be the (partial) function mapping an action label ai to its network session number j.

Definition 4.21. We define the partial $\mathsf{session}$ function:

$$\mathsf{session}(\mathsf{ai}) = j \text{ when } \mathsf{ai} = \mathrm{X}(j, \_) \text{ where } \mathrm{X} \in \{\mathrm{PN}, \mathrm{TN}, \mathrm{FN}\}$$

We let $\mathsf{s\text{-}started}_j(\tau)$ be the predicate holding exactly on action traces where the j-th session of the network started, i.e. where $\mathsf{session}(\mathsf{ai}) = j$ for some ai appearing in τ.

Definition 4.22. For every action trace τ, we let $\mathsf{s\text{-}started}_j(\tau)$ be true if and only if there exists $\mathsf{ai} \in \tau$ such that $\mathsf{session}(\mathsf{ai}) = j$.

We now describe some properties of AKA$^+$. They are formally defined and shown after.

- The property **(A1)** states that the *HN* challenge n^j cannot appear in the frame ϕ_τ if the session j has not started yet. Formally, if $\neg\mathsf{s\text{-}started}_j(\tau)$ then $n^j \notin \mathsf{st}(\phi_\tau)$.
- The properties **(A2)** and **(A3)** give conditions under which some user sequence number has changed.
- **(A4)** expresses the fact that two different users $\mathrm{ID}_1, \mathrm{ID}_2$ can never have the same temporary identities on the server side. This is intuitive, as the server samples temporary identities uniformly at random, and should never assign the same identity to two different users.
- **(A5)**, **(A6)** and **(A7)** state that if the network accepts a message, then there is no ambiguity on the sender. That is, for every $\mathrm{ID}_0 \neq \mathrm{ID}_1$, we cannot have $\mathsf{accept}_\tau^{\mathrm{ID}_0}$ and $\mathsf{accept}_\tau^{\mathrm{ID}_1}$ simultaneously.
- Finally, **(A8)** says that if the user ID believes he authenticated the session j of the network (i.e. $\sigma_\tau^{\mathsf{in}}(\mathsf{e\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}) = \mathsf{n}^j$), then it must have received the challenge n^j when he started his current session (i.e. $\sigma_\tau^{\mathsf{in}}(\mathsf{b\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}) = \mathsf{n}^j$).

Proposition 4.14. *Let $\tau = \_, \mathsf{ai}$ be a valid action trace on $\mathcal{S}_{\mathsf{id}}$, then:*

1. ***(A1)*** *If $\neg\mathsf{s\text{-}started}_j(\tau)$ then $n^j \notin \mathsf{st}(\phi_\tau)$.*
2. ***(A2)*** *For all $\tau_0 = \_, \mathrm{PU}_{\mathrm{ID}}(j_0, 2) \preceq \tau$ and $\tau_1 = \_, \mathrm{PU}_{\mathrm{ID}}(j_1, 2) \preceq \tau$, if $\tau_0 \neq \tau_1$ then:*
$$\sigma_{\tau_0}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) \neq \sigma_{\tau_1}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})$$
3. ***(A3)*** *For every $\tau_0 = \_, \mathrm{PU}_{\mathrm{ID}}(j_0, 2)$, $\tau_1 = \_, \mathrm{PU}_{\mathrm{ID}}(j_1, 1)$ such that $\tau_1 \prec_\tau \tau_0$, if $j_0 \neq j_1$ then:*
$$\sigma_{\tau_0}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) \neq \mathsf{suc}(\sigma_{\tau_1}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}))$$
4. ***(A4)*** *For every $\mathrm{ID}_0, \mathrm{ID}_1 \in \mathcal{S}_{\mathsf{id}}$ such that $\mathrm{ID}_0 \neq \mathrm{ID}_1$:*
$$\left(\sigma_\tau^{\mathsf{in}}(\mathrm{GUTI}_{\mathrm{N}}^{\mathrm{ID}_0}) \neq \mathsf{UnSet} \wedge \sigma_\tau^{\mathsf{in}}(\mathrm{GUTI}_{\mathrm{N}}^{\mathrm{ID}_1}) \neq \mathsf{UnSet}\right) \rightarrow \sigma_\tau^{\mathsf{in}}(\mathrm{GUTI}_{\mathrm{N}}^{\mathrm{ID}_0}) \neq \sigma_\tau^{\mathsf{in}}(\mathrm{GUTI}_{\mathrm{N}}^{\mathrm{ID}_1})$$
5. ***(A5)*** , ***(A6)*** , ***(A7)*** *If $\mathsf{ai} = \mathrm{PN}(j,1), \mathrm{TN}(j,0)$ or $\mathrm{TN}(j,1)$, then for every $\mathrm{ID}_0 \neq \mathrm{ID}_1$,*
$$\left(\neg\mathsf{accept}_\tau^{\mathrm{ID}_0}\right) \vee \left(\neg\mathsf{accept}_\tau^{\mathrm{ID}_1}\right)$$
6. ***(A8)*** *For every $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}, j \in \mathbb{N}$, $\sigma_\tau^{\mathsf{in}}(\mathsf{e\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}) = \mathsf{n}^j \rightarrow \sigma_\tau^{\mathsf{in}}(\mathsf{b\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}) = \mathsf{n}^j$.*

Proof. All these properties are simple to show:

- **(A1)** is trivial by induction over τ.
- **(A2)** and **(A3)** both follow from the fact that if $\tau = \_, \mathrm{PU}_{\mathrm{ID}}(j, 1)$ then $\sigma_\tau(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) \equiv \mathsf{suc}(\sigma_\tau^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}))$, and therefore $\sigma_\tau(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) > \sigma_\tau^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})$.
- **(A5)** and **(A7)** follow easily from the unforgeability axioms EUF-MAC, and the collision resistance axioms CR-KEY$_{\neq}$.
- To prove **(A4)**, we first observe that for every $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$, we initially have $\sigma_\epsilon(\mathrm{GUTI}_{\mathrm{N}}^{\mathrm{ID}}) \equiv \mathsf{UnSet}$, and that the only value we store in $\mathrm{GUTI}_{\mathrm{N}}^{\mathrm{ID}}$ are UnSet or GUTI^i for some $i \in \mathbb{N}$. Therefore it is easy to show that for every $\tau_n \prec \tau$:
$$\sigma_{\tau_n}^{\mathsf{in}}(\mathrm{GUTI}_{\mathrm{N}}^{\mathrm{ID}}) \neq \mathsf{UnSet} \rightarrow \bigvee_{i \in \mathcal{S}} \sigma_{\tau_n}^{\mathsf{in}}(\mathrm{GUTI}_{\mathrm{N}}^{\mathrm{ID}}) = \mathrm{GUTI}^i$$
where $\mathcal{S} \subseteq \mathbb{N}$ is the set of network session number appearing in τ. Moreover, we can only store GUTI^i in $\mathrm{GUTI}_{\mathrm{N}}^{\mathrm{ID}}$ at $\mathrm{PN}(i,1)$ or $\mathrm{TN}(i,1)$, and by validity τ cannot contain both $\mathrm{PN}(i,1)$ and $\mathrm{TN}(i,1)$. We conclude observing that we cannot have $\mathsf{accept}_{\tau_n}^{\mathrm{ID}_0}$ and $\mathsf{accept}_{\tau_n}^{\mathrm{ID}_1}$ if $\tau_n = \_, \mathrm{PN}(i,1)$ or $\_, \mathrm{TN}(i,1)$ using **(A5)** and **(A7)**. The result follows.
- **(A6)** is a consequence of **(A4)**.
- **(A8)** follows from the fact that whenever a new session of the protocol is started, we reset both $\mathsf{b\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}$ and $\mathsf{e\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}$. Then $\mathsf{e\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}$ is either set to fail or to $\mathsf{b\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}$. ■

We can now state and prove our first necessary acceptance conditions.

Lemma 4.6. *Let $\tau = \_, ai$ be a valid action trace on $\mathcal{S}_{id}$, then:*

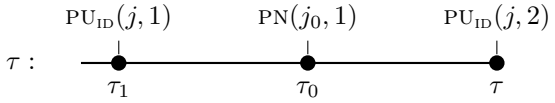
1. ***(Acc1)*** *If $ai = \mathrm{PN}(j,1)$, then for every $\mathrm{ID} \in \mathcal{S}_{id}$:*

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \rightarrow \bigvee_{\tau_0 = \_, \mathrm{PU}_{\mathrm{ID}}(j_0,1) \prec \tau} \left(\pi_1(g(\phi^{in}_{\tau})) = \{ \langle \mathrm{ID}, \sigma^{in}_{\tau_0}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) \rangle \}^{n^{j_0}_{e}}_{pk_{\mathrm{N}}} \wedge g(\phi^{in}_{\tau_0}) = n^j \right)$$

2. ***(Acc2)*** *If $ai = \mathrm{PU}_{\mathrm{ID}}(j,2)$. Let $\tau_1 = \_, \mathrm{PU}_{\mathrm{ID}}(j,1) \prec \tau$. Then:*

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \rightarrow \bigvee_{\substack{\tau_0 = \_, \mathrm{PN}(j_0,1) \\ \tau_1 \prec \tau \, \tau_0}} \mathsf{accept}^{\mathrm{ID}}_{\tau_0} \wedge g(\phi^{in}_{\tau_1}) = n^{j_0} \wedge \pi_1(g(\phi^{in}_{\tau_0})) = \{ \langle \mathrm{ID}, \sigma^{in}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) \rangle \}^{n^{j}_{e}}_{pk_{\mathrm{N}}}$$

To help the reader, we graphically represents how the instants τ_1, τ_0 and τ are situated:[8]



3. ***(Acc3)*** *If $ai = \mathrm{TU}_{\mathrm{ID}}(j,1)$ then:*

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \rightarrow \bigvee_{\substack{\tau_0 = \_, \mathrm{TN}(j_0,0) \\ \tau_0 \prec \tau}} \left(\begin{array}{l} \mathsf{accept}^{\mathrm{ID}}_{\tau_0} \wedge \pi_1(g(\phi^{in}_{\tau})) = n^{j_0} \wedge \pi_2(g(\phi^{in}_{\tau})) = \sigma^{in}_{\tau_0}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \oplus f_k(n^{j_0}) \\ \wedge \sigma^{in}_{\tau}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{in}_{\tau_0}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \end{array} \right)$$

4. ***(Acc4)*** *If $ai = \mathrm{TN}(j,1)$ then:*

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \rightarrow \bigvee_{\tau_0 = \_, \mathrm{TU}_{\mathrm{ID}}(\_,1) \prec \tau} \mathsf{accept}^{\mathrm{ID}}_{\tau_0} \wedge \pi_1(g(\phi^{in}_{\tau_0})) = n^j$$

Proof. The proof of this lemma is given later, in Section 4.9.5. ■

4.9.2 Authentication of the User by the Network

We now prove that the AKA$^+$ protocol provides authentication of the user by the network. Remark that the lemma below subsumes Lemma 4.1.

Lemma 4.7. *For every valid action trace τ on $\mathcal{S}_{id}$, the AKA$^+$ protocol provides authentication of the user by the network:*

$$\forall \mathrm{ID} \in \mathcal{S}_{id}, j \in \mathbb{N}, \;\; \sigma_{\tau}(\textit{e-auth}^{j}_{\mathrm{N}}) = \mathrm{ID} \rightarrow \bigvee_{\tau' \preceq \tau} \sigma_{\tau'}(\textit{b-auth}^{\mathrm{ID}}_{\mathrm{U}}) = n^j$$

Moreover, if $\tau = \_, \mathrm{TN}(j,1)$ then:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \rightarrow \bigvee_{\tau_0 = \_, \mathrm{TU}_{\mathrm{ID}}(\_,1) \prec \tau} \sigma_{\tau_0}(\textit{b-auth}^{\mathrm{ID}}_{\mathrm{U}}) = n^j$$

Proof ⋆(p. 97). We prove this by induction on τ. First, for $\tau = \epsilon$ we have that for every $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$, $\sigma_{\tau}(\mathsf{e\text{-}auth}^{j}_{\mathrm{N}}) \equiv \mathsf{fail} \neq \mathrm{ID}$ by axiom $\neq$-Const. Therefore the property holds.

Let $\tau = \_, \mathsf{ai}$. Let $j \in \mathbb{N}$ be a session number. Remark that if $\sigma^{\mathsf{up}}_{\tau}(\mathsf{e\text{-}auth}^{j}_{\mathrm{N}}) = \bot$, and if the authentication property holds just before the instant τ, i.e.:

$$\forall \mathrm{ID} \in \mathcal{S}_{\mathsf{id}}, \;\; \sigma^{\mathsf{in}}_{\tau}(\mathsf{e\text{-}auth}^{j}_{\mathrm{N}}) = \mathrm{ID} \rightarrow \bigvee_{\tau' \prec \tau} \sigma^{\mathsf{in}}_{\tau'}(\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^j$$

then the authentication property for j holds at instant τ. Therefore we only need to consider the action labels $\mathsf{ai} = \mathrm{PN}(j,1)$ and $\mathsf{ai} = \mathrm{TN}(j,1)$.

[8] We will often use such pictures in this chapter. They are particularly useful when more than two instants are being considered simultaneously. Some conventions: the horizontal line represents the action trace whose name is on the left, before the semi-column (e.g. "τ :" here); instants are represented in their order of appearance at the bottom of the horizontal line; at the top of the line, we indicate (when it is known) the last action of an instant (e.g. here τ_1 ends by $\mathrm{PU}_{\mathrm{ID}}(j,1)$).

- **Case ai $= \mathbf{PN}(j,1)$:** Let $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$. Using $\neq$-Const, we get that $\sigma_\tau(\mathsf{e\text{-}auth}^j_{\mathrm{N}}) = \mathrm{ID} \rightarrow \mathsf{accept}^{\mathrm{ID}}_\tau$. Using **(Acc1)** of Proposition 4.14, we deduce that:

$$\sigma_\tau(\mathsf{e\text{-}auth}^j_{\mathrm{N}}) = \mathrm{ID} \rightarrow \bigvee_{\tau_0 = \_, \mathrm{PU}_{\mathrm{ID}}(j_0,1) \prec \tau} g(\phi^{\mathsf{in}}_{\tau_0}) = \mathsf{n}^j \tag{4.4}$$

By validity of τ, there exists τ_2 such that $\tau_2 = \_, \mathrm{PN}(j,0) \prec \tau$. Let $\tau_0 \prec_\tau \tau_2$, we have $\neg\mathsf{s\text{-}started}_j(\tau_0)$. Using **(A1)**, we get that $n^j \notin \mathsf{st}(\phi^{\mathsf{in}}_{\tau_0})$. It follows from axiom $=$-ind that $g(\phi^{\mathsf{in}}_{\tau_0}) \neq \mathsf{n}^j$. Hence:

$$\bigvee_{\tau_0 = \_, \mathrm{PU}_{\mathrm{ID}}(j_0,1) \prec \tau} g(\phi^{\mathsf{in}}_{\tau_0}) = \mathsf{n}^j \quad \leftrightarrow \quad \bigvee_{\substack{\tau_0 = \_, \mathrm{PU}_{\mathrm{ID}}(j_0,1) \\ \tau_2 \prec_\tau \tau_0 \prec \tau}} g(\phi^{\mathsf{in}}_{\tau_0}) = \mathsf{n}^j \tag{4.5}$$

Let τ_0 be such that $\tau_2 \prec_\tau \tau_0 \prec \tau$ and $\tau_0 = \_, \mathrm{PU}_{\mathrm{ID}}(j_0,1)$. Since $\sigma_{\tau_0}(\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) \equiv g(\phi^{\mathsf{in}}_{\tau_0})$, we have:

$$g(\phi^{\mathsf{in}}_{\tau_0}) = \mathsf{n}^j \rightarrow \sigma_{\tau_0}(\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^j$$

Hence putting (4.4) and (4.5) together, we get:

$$\sigma_\tau(\mathsf{e\text{-}auth}^j_{\mathrm{N}}) = \mathrm{ID} \rightarrow \bigvee_{\substack{\tau_0 = \_, \mathrm{PU}_{\mathrm{ID}}(j_0,1) \\ \tau_2 \prec_\tau \tau_0 \prec \tau}} \sigma_{\tau_0}(\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^j$$

Since $\{\tau_0 \mid \tau_0 = \_, \mathrm{PU}_{\mathrm{ID}}(j_0,1) \wedge \tau_2 \prec_\tau \tau_0 \prec \tau\}$ is a subset of $\{\tau_0 \mid \tau_0 \preceq \tau\}$, we deduce that:

$$\sigma_\tau(\mathsf{e\text{-}auth}^j_{\mathrm{N}}) = \mathrm{ID} \rightarrow \rightarrow \bigvee_{\tau_0 \preceq \tau} \sigma^{\mathsf{in}}_{\tau_0}(\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^j$$

- **Case ai $= \mathbf{TN}(j,1)$:** This case is similar to the previous one. First, we check that $\neq$-Const implies that $\sigma_\tau(\mathsf{e\text{-}auth}^j_{\mathrm{N}}) = \mathrm{ID} \rightarrow \mathsf{accept}^{\mathrm{ID}}_\tau$. Moreover, using **(Acc4)**, we know that:

$$\mathsf{accept}^{\mathrm{ID}}_\tau \rightarrow \bigvee_{\tau_0 = \_, \mathrm{TU}_{\mathrm{ID}}(\_,1) \prec \tau} \mathsf{accept}_{\tau_0} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau_0})) = \mathsf{n}^j$$

Moreover, for every $\tau_0 = \_, \mathrm{TU}_{\mathrm{ID}}(\_,1) \prec \tau$, we have:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau_0} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau_0})) = \mathsf{n}^j \rightarrow \sigma_{\tau_0}(\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^j$$

Hence:

$$\begin{aligned} \mathsf{accept}^{\mathrm{ID}}_\tau &\rightarrow \bigvee_{\tau_0 = \_, \mathrm{TU}_{\mathrm{ID}}(\_,1) \prec \tau} \mathsf{accept}_{\tau_0} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau_0})) = \mathsf{n}^j \\ &\rightarrow \bigvee_{\tau_0 = \_, \mathrm{TU}_{\mathrm{ID}}(\_,1) \prec \tau} \sigma_{\tau_0}(\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^j \\ &\rightarrow \bigvee_{\tau_0 \preceq \tau} \sigma^{\mathsf{in}}_{\tau_0}(\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^j \end{aligned}$$

■

4.9.3 Authentication of the Network by the User

We prove that the AKA$^+$ protocols provides authentication of the network by the user. We actually prove the stronger result that for any valid action trace τ, if the authentication of UE_{ID} succeeded at instant τ (i.e. $\sigma^{\mathsf{in}}_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) \neq \mathsf{fail}$), then there exists some $j \in \mathbb{N}$ such that UE_{ID} authenticated $HN(j)$.

Lemma 4.8. *For all valid action trace τ on $\mathcal{S}_{id}$, the AKA$^+$ protocol provides authentication of the network by the user. Formally, for every $\mathrm{ID} \in \mathcal{S}_{id}$ and $j \in \mathbb{N}$, we let:*

$$\mathit{suc\text{-}auth}_\tau(\mathrm{ID}) \equiv \sigma_\tau(\mathit{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) \neq \mathit{fail} \qquad \mathit{auth}_\tau(\mathrm{ID}, j) \equiv \sigma_\tau(\mathit{b\text{-}auth}^j_{\mathrm{N}}) = \mathrm{ID} \wedge n^j = \sigma_\tau(\mathit{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}})$$

Then:

$$\forall \mathrm{ID} \in \mathcal{S}_{id},\ \ \mathit{suc\text{-}auth}_\tau(\mathrm{ID}) \rightarrow \bigvee_{\mathit{s\text{-}started}_j(\tau)} \mathit{auth}_\tau(\mathrm{ID}, j)$$

Proof ⋆(p. 100). We prove this by induction on τ. First, for $\tau = \epsilon$ we have that for every $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$, $\sigma_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) \equiv \mathsf{fail}$. Therefore the property holds. Let $\tau = \tau_0, \mathsf{ai}$, and assume by induction that:

$$\forall\, \mathrm{ID} \in \mathcal{S}_{\mathsf{id}},\ \ \mathsf{suc\text{-}auth}_{\tau_0}(\mathrm{ID}) \ \rightarrow\ \bigvee_{\mathsf{s\text{-}started}_j(\tau_0)} \mathsf{auth}_{\tau_0}(\mathrm{ID}, j)$$

If for every j_0 and ID we have:

$$\sigma^{\mathsf{up}}_\tau(\mathsf{b\text{-}auth}^{j_0}_{\mathrm{N}}) = \bot \qquad\qquad \sigma^{\mathsf{up}}_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \bot$$

then, by induction hypothesis, we have authentication of the network by the user at τ. Therefore it only remains to show that authentication holds for τ in the cases where ai is equal to $\mathrm{PN}(j,1)$, $\mathrm{PU}_{\mathrm{ID}}(j,1)$, $\mathrm{PU}_{\mathrm{ID}}(j,2)$, $\mathrm{TU}_{\mathrm{ID}}(j,0)$, $\mathrm{TN}(j,0)$ or $\mathrm{TU}_{\mathrm{ID}}(j,1)$.

Before starting the case disjunction, remark that if we can prove that for every $\mathrm{ID}_0 \in \mathcal{S}_{\mathsf{id}}$ and $j_0 \in \mathbb{N}$:

$$(\mathsf{suc\text{-}auth}_\tau(\mathrm{ID}_0) \ \wedge\ \mathsf{auth}_\tau(\mathrm{ID}_0, j_0)) \ \leftrightarrow\ (\mathsf{suc\text{-}auth}_{\tau_0}(\mathrm{ID}_0) \ \wedge\ \mathsf{auth}_{\tau_0}(\mathrm{ID}_0, j_0)) \tag{4.6}$$

Then we can directly conclude by applying the induction hypothesis. We now do a case disjunction on ai.

- **Cases $\mathsf{ai} = \mathbf{PU}_{\mathbf{ID}}(j,1)$ and $\mathsf{ai} = \mathbf{TU}_{\mathbf{ID}}(j,0)$.** In both cases, we have $\sigma_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) \equiv \mathsf{fail}$, and therefore the property trivially holds for ID. Besides, for every $\mathrm{ID}_0 \neq \mathrm{ID}$ and $j_0 \in \mathbb{N}$, (4.6) holds.
- **Case $\mathsf{ai} = \mathbf{TU}_{\mathbf{ID}}(j,1)$.** For all $\mathrm{ID}_0 \neq \mathrm{ID}$ and for all $j_0 \in \mathbb{N}$, we easily check that (4.6) holds. It only remains to show that:

  $$\mathsf{suc\text{-}auth}_\tau(\mathrm{ID}) \ \rightarrow\ \bigvee_{\mathsf{s\text{-}started}_i(\tau)} \mathsf{auth}_\tau(\mathrm{ID}, i)$$

  Let $\mathsf{k} \equiv \mathsf{k}^{\mathrm{ID}}$. We observe that:

  $$\begin{aligned}
  \mathsf{suc\text{-}auth}_\tau(\mathrm{ID}) \ &\rightarrow\ \sigma_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) \neq \mathsf{fail} \\
  &\rightarrow\ \mathsf{accept}^{\mathrm{ID}}_\tau \\
  &\rightarrow\ \bigvee_{\tau_0 = _, \mathrm{TN}(j_0,0) \prec \tau} \left(\begin{array}{l} \mathsf{accept}^{\mathrm{ID}}_{\tau_0} \ \wedge\ \pi_1(g(\phi^{\mathsf{in}}_\tau)) = \mathsf{n}^{j_0} \ \wedge \\ \pi_2(g(\phi^{\mathsf{in}}_\tau)) = \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \oplus \mathsf{f}_{\mathsf{k}}(\mathsf{n}^{j_0}) \end{array} \right) \qquad \text{(by } \mathbf{(Acc3)}\text{)}
  \end{aligned}$$

  Let $\tau_0 = \mathrm{TN}(j_0, 0)$ such that $\tau_0 \prec_\tau \tau$. Then:

  $$\pi_1(g(\phi^{\mathsf{in}}_\tau)) = \mathsf{n}^{j_0} \wedge \mathsf{accept}^{\mathrm{ID}}_\tau \ \rightarrow\ \sigma_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^{j_0}$$

  Moreover using **(A7)** we know that $\mathsf{accept}^{\mathrm{ID}}_{\tau_0} \rightarrow \sigma_{\tau_0}(\mathsf{b\text{-}auth}^{j_0}_{\mathrm{N}}) = \mathrm{ID}$. Using the validity of τ, we can easily show that for all $\tau_0 \prec \tau' \preceq \tau$ we have $\sigma^{\mathsf{up}}_{\tau'}(\mathsf{b\text{-}auth}^{j_0}_{\mathrm{N}}) \equiv \bot$. We deduce that $\mathsf{accept}^{\mathrm{ID}}_{\tau_0} \rightarrow \sigma_\tau(\mathsf{b\text{-}auth}^{j_0}_{\mathrm{N}}) = \mathrm{ID}$. Hence:

  $$\mathsf{suc\text{-}auth}_\tau(\mathrm{ID}) \ \rightarrow\ \bigvee_{\tau_0 = \_, \mathrm{TN}(j_0,0) \prec \tau} \mathsf{auth}_\tau(\mathrm{ID}, j_0) \ \rightarrow\ \bigvee_{\mathsf{s\text{-}started}_{j_0}(\tau)} \mathsf{auth}_\tau(\mathrm{ID}, j_0)$$

- **Case $\mathsf{ai} = \mathbf{PU}_{\mathbf{ID}}(j,2)$.** For all $\mathrm{ID}_0 \neq \mathrm{ID}$ and for all $j_0 \in \mathbb{N}$, we check that (4.6) holds. It remains to prove that:

  $$\mathsf{suc\text{-}auth}_\tau(\mathrm{ID}) \ \rightarrow\ \bigvee_{\mathsf{s\text{-}started}_j(\tau)} \mathsf{auth}_\tau(\mathrm{ID}, j)$$

  First, we observe that:

  $$\begin{aligned}
  \mathsf{suc\text{-}auth}_\tau(\mathrm{ID}) \ &\rightarrow\ \mathsf{accept}^{\mathrm{ID}}_\tau \\
  &\rightarrow\ \bigvee_{\substack{\tau_0 = _, \mathrm{PN}(j_0,1) \\ \tau_1 = _, \mathrm{PU}_{\mathrm{ID}}(j,1) \\ \tau_1 \prec_\tau \tau_0}} \left(\begin{array}{l} \mathsf{accept}^{\mathrm{ID}}_{\tau_0} \ \wedge\ g(\phi^{\mathsf{in}}_{\tau_1}) = \mathsf{n}^{j_0} \ \wedge \\ \pi_1(g(\phi^{\mathsf{in}}_{\tau_0})) = \{\!|\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle|\!\}^{\mathsf{n}^j_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} \end{array} \right) \qquad \text{(by } \mathbf{(Acc2)}\text{)}
  \end{aligned}$$

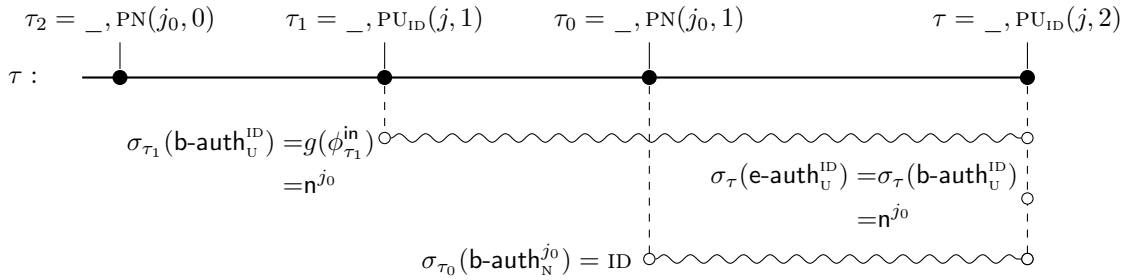
  Let $\tau_0 = \_, \mathrm{PN}(j_0, 1)$, $\tau_1 = \_, \mathrm{PU}_{\mathrm{ID}}(j, 1)$ such that $\tau_1 \prec_\tau \tau_0$. Let $\tau_2 = \_, \mathrm{PN}(j_0, 0)$, by validity of τ we know that $\tau_2 \prec_\tau \tau_0$. Moreover, if $\tau_1 \prec_\tau \tau_2$ then by **(A1)** we have $\mathsf{n}^{j_0} \notin \mathsf{st}(\phi^{\mathsf{in}}_{\tau_1})$, and therefore using $=$-ind we obtain that $g(\phi^{\mathsf{in}}_{\tau_1}) \neq \mathsf{n}^{j_0}$. Hence:

  $$\mathsf{suc\text{-}auth}_\tau(\mathrm{ID}) \ \rightarrow\ \bigvee_{\substack{\tau_0 = \_, \mathrm{PN}(j_0,1) \\ \tau_1 = \_, \mathrm{PU}_{\mathrm{ID}}(j,1) \\ \tau_2 = \_, \mathrm{PN}(j_0,0) \\ \tau_2 \prec_\tau \tau_1 \prec_\tau \tau_0}} \overbrace{\mathsf{accept}^{\mathrm{ID}}_\tau \wedge \mathsf{accept}^{\mathrm{ID}}_{\tau_0} \wedge g(\phi^{\mathsf{in}}_{\tau_1}) = \mathsf{n}^{j_0} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau_0})) = \{\!|\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle|\!\}^{\mathsf{n}^j_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}}}^{\psi^{\tau_1}_{\tau_2,\tau_0}}$$

We know that $g(\phi^{\mathsf{in}}_{\tau_1}) = \mathsf{n}^{j_0} \rightarrow \sigma_{\tau_1}(\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^{j_0}$, and that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau_0} \rightarrow \sigma_{\tau_0}(\mathsf{b\text{-}auth}^{j_0}_{\mathrm{N}}) = \mathrm{ID} \qquad \mathsf{accept}^{\mathrm{ID}}_{\tau} \rightarrow \sigma_{\tau}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma_{\tau}(\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}})$$

We represent graphically all the information we have below:



It follows that $\psi^{\tau_1}_{\tau_2,\tau_0} \rightarrow \mathsf{auth}_{\tau}(\mathrm{ID}, j_0)$. Hence:

$$\mathsf{suc\text{-}auth}_{\tau}(\mathrm{ID}) \rightarrow \bigvee_{\substack{\tau_0 = \_, \mathrm{PN}(j_0,1) \\ \tau_1 = \_, \mathrm{PU}_{\mathrm{ID}}(j,1) \\ \tau_2 = \_, \mathrm{PN}(j_0,0) \\ \tau_2 \prec_\tau \tau_1 \prec_\tau \tau_0}} \mathsf{auth}_{\tau}(\mathrm{ID}, j_0) \rightarrow \bigvee_{\mathsf{s\text{-}started}_{j_0}(\tau)} \mathsf{auth}_{\tau}(\mathrm{ID}, j_0)$$

- **Case $\mathsf{ai} = \mathrm{PN}(j, 1)$.** For all $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$ and for all $j_0 \in \mathbb{N}$ such that $j_0 \neq j$ we have:

$$\mathsf{suc\text{-}auth}_{\tau}(\mathrm{ID}) \equiv \mathsf{suc\text{-}auth}_{\tau_0}(\mathrm{ID}) \qquad \mathsf{auth}_{\tau}(\mathrm{ID}, j_0) \equiv \mathsf{auth}_{\tau_0}(\mathrm{ID}, j_0)$$

Hence (4.6) holds. It only remains the case where $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$ and $j_0 = j$. By validity of τ we know that $\sigma^{\mathsf{in}}_{\tau}(\mathsf{b\text{-}auth}^{j}_{\mathrm{N}}) \equiv \mathsf{fail}$. From $\neq$-Const it follows that $\sigma^{\mathsf{in}}_{\tau}(\mathsf{b\text{-}auth}^{j}_{\mathrm{N}}) \neq \mathrm{ID}$, and therefore $\mathsf{auth}_{\tau_0}(\mathrm{ID}, j) \leftrightarrow \mathsf{false}$.
To conclude this case, we only need to show that $(\mathsf{suc\text{-}auth}_{\tau}(\mathrm{ID}) \wedge \mathsf{auth}_{\tau}(\mathrm{ID}, j)) \leftrightarrow \mathsf{false}$. We recall that $\mathsf{suc\text{-}auth}_{\tau}(\mathrm{ID}) \equiv \sigma_{\tau}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) \neq \mathsf{fail}$. The only instants that can set $\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}$ to something other than fail are $\mathrm{PU}_{\mathrm{ID}}(\_, 2)$ and $\mathrm{TU}_{\mathrm{ID}}(\_, 1)$. Formally, we show by induction on τ that:

$$\sigma_{\tau}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) \neq \mathsf{fail} \rightarrow \bigvee_{\substack{\tau_0 \preceq \tau \\ \tau_0 = \_, \mathrm{PU}_{\mathrm{ID}}(\_,2) \\ \vee\ \tau_0 = \mathrm{TU}_{\mathrm{ID}}(\_,1)}} \mathsf{accept}^{\mathrm{ID}}_{\tau_0} \wedge \sigma_{\tau}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma_{\tau_0}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) \tag{4.7}$$

Therefore we only have to prove that for any τ_0 in the disjunction above, we have:

$$\left(\mathsf{suc\text{-}auth}_{\tau}(\mathrm{ID}) \wedge \mathsf{auth}_{\tau}(\mathrm{ID}, j) \wedge \mathsf{accept}^{\mathrm{ID}}_{\tau_0} \wedge \sigma_{\tau}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma_{\tau_0}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}})\right) \leftrightarrow \mathsf{false}$$

We have two cases:

– Let $\tau_0 = \_, \mathrm{PU}_{\mathrm{ID}}(j_0, 2) \preceq \tau$. By validity of τ, we know that there exists $\tau_2 \prec_\tau \tau_0$ such that $\tau_2 = \_, \mathrm{PU}_{\mathrm{ID}}(j_0, 1)$. By **(Acc2)**:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau_0} \rightarrow \bigvee_{\substack{\tau_1 = \_, \mathrm{PN}(j_1,1) \\ \tau_2 \prec_\tau \tau_1 \prec_\tau \tau_0}} g(\phi^{\mathsf{in}}_{\tau_2}) = \mathsf{n}^{j_1}$$

Moreover, $\mathsf{accept}^{\mathrm{ID}}_{\tau_0} \wedge g(\phi^{\mathsf{in}}_{\tau_2}) = \mathsf{n}^{j_1} \rightarrow \sigma_{\tau_0}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^{j_1}$. Therefore:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau_0} \wedge \sigma_{\tau}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma_{\tau_0}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) \rightarrow \bigvee_{\substack{\tau_1 = \_, \mathrm{PN}(j_1,1) \\ \tau_2 \prec_\tau \tau_1 \prec_\tau \tau_0}} \sigma_{\tau}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^{j_1}$$

Since $\tau_0 \prec \tau$ we know that for every $\tau_1 = \_, \mathrm{PN}(j_1, 1) \in \{\tau_1 \mid \tau_2 \prec \tau_1 \prec_\tau \tau_0\}$, $j_1 \neq j$. Using $=$-ind we deduce that $\mathsf{n}^{j_1} \neq \mathsf{n}^{j}$. Since $\mathsf{auth}_{\tau}(\mathrm{ID}, j) \rightarrow \mathsf{n}^{j} = \sigma_{\tau}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}})$, we obtain that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau_0} \wedge \sigma_{\tau}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma_{\tau_0}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{auth}_{\tau}(\mathrm{ID}, j) \rightarrow \bigvee_{\substack{\tau_1 = \_, \mathrm{PN}(j_1,1) \\ \tau_2 \prec_\tau \tau_1 \prec_\tau \tau_0}} \mathsf{n}^{j} = \mathsf{n}^{j_1}$$
$$\rightarrow \mathsf{false}$$

– Let $\tau_0 = \_, \mathrm{TU}_{\mathrm{ID}}(j_0, 1) \preceq \tau$. We do a similar reasoning. By **(Acc3)**:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau_0} \rightarrow \bigvee_{\substack{\tau_1 = \_, \mathrm{TN}(j_1, 0) \\ \tau_1 \prec_\tau \tau_0}} \pi_1(g(\phi^{\mathsf{in}}_{\tau_0})) = \mathsf{n}^{j_1}$$

Remark that $\mathsf{accept}^{\mathrm{ID}}_{\tau_0} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau_0})) = \mathsf{n}^{j_1} \rightarrow \sigma_{\tau_0}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^{j_1}$. Hence:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau_0} \wedge \sigma_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma_{\tau_0}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) \rightarrow \bigvee_{\substack{\tau_1 = \_, \mathrm{TN}(j_1, 0) \\ \tau_1 \prec_\tau \tau_0}} \sigma_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^{j_1}$$

By validity of τ, we know that for every $\tau_1 = \_, \mathrm{TN}(j_1, 0)$ such that $\tau_1 \prec_\tau \tau_0$, we have $j_1 \neq j$, and by consequence $\mathsf{n}^{j_1} \neq \mathsf{n}^j$. Since $\mathsf{auth}_\tau(\mathrm{ID}, j) \rightarrow \mathsf{n}^j = \sigma_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}})$, we obtain that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau_0} \wedge \sigma_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma_{\tau_0}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{auth}_\tau(\mathrm{ID}, j) \rightarrow \bigvee_{\substack{\tau_1 = \_, \mathrm{TN}(j_1, 0) \\ \tau_1 \prec_\tau \tau_0}} \mathsf{n}^j = \mathsf{n}^{j_1}$$
$$\rightarrow \mathsf{false}$$

- **Case $\mathbf{ai} = \mathrm{TN}(j, 0)$.** Again, for all $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$ and for all $j_0 \in \mathbb{N}$ such that $j_0 \neq j$:

$$\mathsf{suc\text{-}auth}_\tau(\mathrm{ID}) \equiv \mathsf{suc\text{-}auth}_{\tau_0}(\mathrm{ID}) \qquad \mathsf{auth}_\tau(\mathrm{ID}, j_0) \equiv \mathsf{auth}_{\tau_0}(\mathrm{ID}, j_0)$$

Hence (4.6) holds. It only remains the case $j_0 = j$. We know that $\sigma^{\mathsf{in}}_\tau(\mathsf{b\text{-}auth}^j_{\mathrm{N}}) \equiv \mathsf{fail}$, therefore $\mathsf{suc\text{-}auth}_{\tau_0}(\mathrm{ID}, j) = \mathsf{false}$, which in turn implies that:

$$(\mathsf{suc\text{-}auth}_{\tau_0}(\mathrm{ID}) \wedge \mathsf{auth}_{\tau_0}(\mathrm{ID}, j)) \leftrightarrow \mathsf{false}$$

Moreover:

$$\mathsf{auth}_\tau(\mathrm{ID}, j) \rightarrow \mathsf{n}^j = \sigma_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}})$$

Remark that $\sigma_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) \equiv \sigma^{\mathsf{in}}_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}})$. Using **(A1)** we easily show that n^j does not appear in $\mathsf{st}(\sigma^{\mathsf{in}}_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}))$. Therefore $\neg\mathsf{auth}_\tau(\mathrm{ID}, j)$ by $=$-ind. ■

Using Lemma 4.8, we can prove Lemma 4.3, which we recall below:

Lemma (4.3). *For every valid action trace τ on $\mathcal{S}_{\mathsf{id}}$, $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$ and $j \in \mathbb{N}$, we have:*

$$\sigma_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^j \rightarrow \bigvee_{\tau' \preceq \tau} \sigma_{\tau'}(\mathsf{b\text{-}auth}^j_{\mathrm{N}}) = \mathrm{ID}$$

Proof. Let τ be a valid action trace. First, observe that $\sigma_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^j$ implies that $\sigma_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) \neq \mathsf{fail}$. Therefore, using Lemma 4.8 we get that:

$$\begin{aligned} \sigma_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^j &\rightarrow \sigma_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) \neq \mathsf{fail} \\ &\rightarrow \mathsf{suc\text{-}auth}_\tau(\mathrm{ID}) \\ &\rightarrow \bigvee_{\mathsf{s\text{-}started}_{j'}(\tau)} \mathsf{auth}_\tau(\mathrm{ID}, j') \qquad \text{(By Lemma 4.8)} \end{aligned}$$

Since $(\mathsf{n}^j = \sigma_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{n}^{j'} = \sigma_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}})) \leftrightarrow \mathsf{false}$ if $j \neq j'$:

$$\begin{aligned} &\rightarrow \sigma_\tau(\mathsf{b\text{-}auth}^j_{\mathrm{N}}) = \mathrm{ID} \\ &\rightarrow \bigvee_{\tau' \preceq \tau} \sigma_{\tau'}(\mathsf{b\text{-}auth}^j_{\mathrm{N}}) = \mathrm{ID} \end{aligned}$$

■

4.9.4 Injective Authentication of the Network by the User

We actually can show that the authentication of the network by the user is *injective*.

Lemma 4.9. *For every valid action trace τ on $\mathcal{S}_{\mathsf{id}}$, the AKA$^+$ protocol provides* injective *authentication of the network by the user. Formally, for every $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$ and $j \in \mathbb{N}$, we define the formula:*

$$\mathit{inj\text{-}auth}_\tau(\mathrm{ID}, j) \equiv \mathit{auth}_\tau(\mathrm{ID}, j) \wedge \bigwedge_{\substack{i \neq j \\ \mathsf{s\text{-}started}_i(\tau)}} \neg \mathit{auth}_\tau(\mathrm{ID}, i)$$

Then:

$$\forall \mathrm{ID} \in \mathcal{S}_{\mathsf{id}}, \quad \mathit{suc\text{-}auth}_\tau(\mathrm{ID}) \rightarrow \bigvee_{\mathsf{s\text{-}started}_j(\tau)} \mathit{inj\text{-}auth}_\tau(\mathrm{ID}, j)$$

Proof. First, we show that for $\text{ID} \in \mathcal{S}_{\mathsf{id}}$ and $i_0, i_1 \in \mathbb{N}$ with $i_0 \neq i_1$:

$$\mathsf{suc\text{-}auth}_\tau(\text{ID}) \rightarrow (\neg \mathsf{auth}_\tau(\text{ID}, i_0) \vee \neg \mathsf{auth}_\tau(\text{ID}, i_1)) \tag{4.8}$$

Indeed:

$$\mathsf{suc\text{-}auth}_\tau(\text{ID}) \wedge \mathsf{auth}_\tau(\text{ID}, i_0) \wedge \mathsf{auth}_\tau(\text{ID}, i_1) \;\rightarrow\; \mathsf{n}^{i_0} = \sigma_\tau(\mathsf{e\text{-}auth}_{\text{U}}^{\text{ID}}) \wedge \mathsf{n}^{i_1} = \sigma_\tau(\mathsf{e\text{-}auth}_{\text{U}}^{\text{ID}})$$

Using $=$-ind, we know that $\mathsf{n}^{i_1} \neq \mathsf{n}^{i_0}$. Therefore:

$$\mathsf{n}^{i_0} = \sigma_\tau(\mathsf{e\text{-}auth}_{\text{U}}^{\text{ID}}) \wedge \mathsf{n}^{i_1} = \sigma_\tau(\mathsf{e\text{-}auth}_{\text{U}}^{\text{ID}}) \rightarrow \mathsf{false}$$

This concludes the proof of (4.8). From Lemma 4.8 we know that:

$$\forall \text{ID} \in \mathcal{S}_{\mathsf{id}},\;\; \mathsf{suc\text{-}auth}_\tau(\text{ID}) \;\rightarrow\; \bigvee_{\mathsf{s\text{-}started}_j(\tau)} \mathsf{auth}_\tau(\text{ID}, j)$$

Moreover, using (4.8) we have that for every $\text{ID} \in \mathcal{S}_{\mathsf{id}}, j \in \mathbb{N}$:

$$\mathsf{suc\text{-}auth}_\tau(\text{ID}) \wedge \mathsf{auth}_\tau(\text{ID}, j) \;\rightarrow\; \bigwedge_{\substack{i \neq j \\ \mathsf{s\text{-}started}_j(\tau)}} \neg \mathsf{auth}_\tau(\text{ID}, i)$$

We deduce that:

$$\forall \text{ID} \in \mathcal{S}_{\mathsf{id}},\;\; \mathsf{suc\text{-}auth}_\tau(\text{ID}) \;\rightarrow\; \bigvee_{\mathsf{s\text{-}started}_j(\tau)} \mathsf{inj\text{-}auth}_\tau(\text{ID}, j)$$ ■

Finally, we prove that ID authenticated j_0 at τ if and only if $\mathsf{n}^{j_0} = \sigma_\tau(\mathsf{e\text{-}auth}_{\text{U}}^{\text{ID}})$.

Proposition 4.15. *For every valid action trace τ, for every $j_0 \in \mathbb{N}$:*

$$\mathit{inj\text{-}auth}_\tau(\text{ID}, j_0) \;\leftrightarrow\; \mathit{n}^{j_0} = \sigma_\tau(\mathit{e\text{-}auth}_{\text{U}}^{\text{ID}})$$

Proof. To do this we show both directions. The first direction is trivial:

$$\mathsf{inj\text{-}auth}_\tau(\text{ID}, j_0) \;\rightarrow\; \mathsf{auth}_\tau(\text{ID}, j_0) \;\rightarrow\; \mathsf{n}^{j_0} = \sigma_\tau^{\mathsf{in}}(\mathsf{e\text{-}auth}_{\text{U}}^{\text{ID}})$$

We now prove the converse direction:

$$\begin{aligned} \mathsf{n}^{j_0} = \sigma_\tau^{\mathsf{in}}(\mathsf{e\text{-}auth}_{\text{U}}^{\text{ID}}) &\rightarrow \mathsf{suc\text{-}auth}_\tau(\text{ID}) && \text{(Using } =\text{-}\mathsf{ind}) \\ &\rightarrow \bigvee_{\mathsf{s\text{-}started}_{j_1}(\tau)} \mathsf{inj\text{-}auth}_\tau(\text{ID}, j_1) && \text{(Lemma 4.9)} \end{aligned}$$

We conclude by observing that for every $j_1 \neq j_0$:

$$\begin{aligned} \mathsf{n}^{j_0} = \sigma_\tau(\mathsf{e\text{-}auth}_{\text{U}}^{\text{ID}}) \wedge \mathsf{inj\text{-}auth}_\tau(\text{ID}, j_1) &\rightarrow \mathsf{n}^{j_0} = \sigma_\tau(\mathsf{e\text{-}auth}_{\text{U}}^{\text{ID}}) \wedge \mathsf{n}^{j_1} = \sigma_\tau(\mathsf{e\text{-}auth}_{\text{U}}^{\text{ID}}) \\ &\rightarrow \mathsf{false} && \text{(Using } =\text{-}\mathsf{ind}) \end{aligned}$$ ■

4.9.5 ⋆ (p. 104) Proof of Lemma 4.6

*Proof of **(Acc1)**.* Let $\mathsf{ai} = \text{PN}(j, 1)$ and $\mathsf{k_m} \equiv \mathsf{k}_{\mathsf{m}}^{\text{ID}}$. Recall that:

$$\mathsf{accept}_\tau^{\text{ID}} \;\equiv\; \mathsf{eq}(\pi_1(\mathsf{dec}(\pi_1(g(\phi_\tau^{\mathsf{in}})), \mathsf{sk}_{\text{N}})), \text{ID}) \wedge \mathsf{eq}(\pi_2(g(\phi_\tau^{\mathsf{in}})), \mathsf{Mac}_{\mathsf{k_m}}^1(\langle \pi_1(g(\phi_\tau^{\mathsf{in}})), \mathsf{n}^j \rangle))$$

We apply the P-EUF-MAC^1 axiom (invariant (INV-KEY) guarantees that the syntactic side-conditions hold):

$$\begin{aligned} \mathsf{accept}_\tau^{\text{ID}} \;&\rightarrow\; \pi_2(g(\phi_\tau^{\mathsf{in}})) = \mathsf{Mac}_{\mathsf{k_m}}^1(\langle \pi_1(g(\phi_\tau^{\mathsf{in}})), \mathsf{n}^j \rangle) \\ &\rightarrow \bigvee_{\tau_0 = \_, \text{PU}_{\text{ID}}(j_0, 1) \prec \tau} \pi_2(g(\phi_\tau^{\mathsf{in}})) = \mathsf{Mac}_{\mathsf{k_m}}^1(\langle \{\langle \text{ID}, \sigma_{\tau_0}^{\mathsf{in}}(\text{SQN}_{\text{U}}^{\text{ID}}) \rangle\}_{\mathsf{pk}_{\text{N}}}^{\mathsf{n_e}^{j_0}}, g(\phi_{\tau_0}^{\mathsf{in}}) \rangle) \end{aligned}$$

Finally, we use CR^1, $\mathsf{EQInj}(\langle \_, \cdot \rangle)$ and $\mathsf{EQInj}(\langle \cdot, \_ \rangle)$ to show that for every $\tau_0 = \_, \text{PU}_{\text{ID}}(j_0, 1) \prec \tau$:

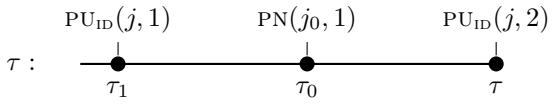
$$\mathsf{Mac}_{\mathsf{k_m}}^1(\langle \pi_1(g(\phi_\tau^{\mathsf{in}})), \mathsf{n}^j \rangle) = \mathsf{Mac}_{\mathsf{k_m}}^1(\langle \{\langle \text{ID}, \sigma_{\tau_0}^{\mathsf{in}}(\text{SQN}_{\text{U}}^{\text{ID}}) \rangle\}_{\mathsf{pk}_{\text{N}}}^{\mathsf{n_e}^{j_0}}, g(\phi_{\tau_0}^{\mathsf{in}}) \rangle) \;\rightarrow$$

$$\pi_1(g(\phi_\tau^{\mathsf{in}})) = \{\langle \text{ID}, \sigma_{\tau_0}^{\mathsf{in}}(\text{SQN}_{\text{U}}^{\text{ID}}) \rangle\}_{\mathsf{pk}_{\text{N}}}^{\mathsf{n_e}^{j_0}} \wedge \mathsf{n}^j = g(\phi_{\tau_0}^{\mathsf{in}})$$ ■

Proof of ***(Acc2)***. If $\mathsf{ai} = \mathrm{PU}_{\mathrm{ID}}(j,2)$. Let $\mathsf{k_m} \equiv \mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}$. Recall that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \;\equiv\; g(\phi^{\mathsf{in}}_{\tau}) = \mathsf{Mac}^2_{\mathsf{k_m}}(\langle \sigma^{\mathsf{in}}_{\tau}(\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}})\,,\, \sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle)$$

Graphically, we are in the situation:



Part 1 We are going to apply the P-EUF-MAC^2 axiom. We let:

$$S = \{\tau_0 \mid \tau_0 = \_, \mathrm{PN}(j_0,1) \prec \tau\}$$

and for all $S_0 \subseteq S$ we let:

$$b_{S_0} = \Big(\bigwedge_{\tau_0 \in S_0} \mathsf{accept}^{\mathrm{ID}}_{\tau_0}\Big) \wedge \Big(\bigwedge_{\tau_0 \in \overline{S_0}} \neg \mathsf{accept}^{\mathrm{ID}}_{\tau_0}\Big)$$

Then $(b_{S_0})_{S_0 \subseteq S}$ is a valid CS partition. It is straightforward to check that for every $S_0 \subseteq S$, for every $\tau_0 = \_, \mathrm{PN}(j_0,1) \prec \tau$, if $\tau_0 \in S_0$ then we can rewrite $[b_{S_0}]t_{\tau_0}$ into a term $[b_{S_0}]t^{S_0}_{\tau_0}$ by removing the branch corresponding to $\mathsf{accept}^{\mathrm{ID}}_{\tau_0}$. Therefore:

$$\mathsf{Mac}^2_{\mathsf{k_m}}(\langle \mathsf{n}^{j_0}\,,\, \mathsf{suc}(\pi_2(\mathsf{dec}(\pi_1(g(\phi^{\mathsf{in}}_{\tau_0})), \mathsf{sk}_{\mathrm{N}})))\rangle) \in \mathsf{set\text{-}mac}^2_{\mathsf{k_m}}(t^{S_0}_{\tau_0}) \text{ if and only if } \tau_0 \in S_0$$

Hence by applying the P-EUF-MAC^2 axiom we get that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \to \bigvee_{S_0 \subseteq S} b_{S_0} \wedge \bigvee_{\tau_0 \in S_0} g(\phi^{\mathsf{in}}_{\tau}) = \mathsf{Mac}^2_{\mathsf{k_m}}(\langle \mathsf{n}^{j_0}\,,\, \mathsf{suc}(\pi_2(\mathsf{dec}(\pi_1(g(\phi^{\mathsf{in}}_{\tau_0})), \mathsf{sk}_{\mathrm{N}})))\rangle)$$

For $S_0 = \emptyset$, we have:

$$\neg\Big(\bigvee_{\tau_0 \in S_0} g(\phi^{\mathsf{in}}_{\tau}) = \mathsf{Mac}^2_{\mathsf{k_m}}(\langle \mathsf{n}^{j}\,,\, \mathsf{suc}(\pi_2(\mathsf{dec}(\pi_1(g(\phi^{\mathsf{in}}_{\tau_0})), \mathsf{sk}_{\mathrm{N}})))\rangle)\Big)$$

Hence:

$$\begin{aligned}
\mathsf{accept}^{\mathrm{ID}}_{\tau} \;&\to\; \bigvee_{\substack{S_0 \subseteq S \\ S_0 \neq \emptyset}} b_{S_0} \wedge \bigvee_{\tau_0 \in S_0} g(\phi^{\mathsf{in}}_{\tau}) = \mathsf{Mac}^2_{\mathsf{k_m}}(\langle \mathsf{n}^{j}\,,\, \mathsf{suc}(\pi_2(\mathsf{dec}(\pi_1(g(\phi^{\mathsf{in}}_{\tau_0})), \mathsf{sk}_{\mathrm{N}})))\rangle) \\
&\to \bigvee_{\substack{S_0 \subseteq S \\ S_0 \neq \emptyset}} \bigvee_{\tau_0 \in S_0} \mathsf{accept}^{\mathrm{ID}}_{\tau_0} \;\wedge\; g(\phi^{\mathsf{in}}_{\tau}) = \mathsf{Mac}^2_{\mathsf{k_m}}(\langle \mathsf{n}^{j}\,,\, \mathsf{suc}(\pi_2(\mathsf{dec}(\pi_1(g(\phi^{\mathsf{in}}_{\tau_0})), \mathsf{sk}_{\mathrm{N}})))\rangle) \\
&\to \bigvee_{\tau_0 = _, \mathrm{PN}(j_0,1)\atop \tau_0 \prec \tau} \mathsf{accept}^{\mathrm{ID}}_{\tau_0} \;\wedge\; g(\phi^{\mathsf{in}}_{\tau}) = \mathsf{Mac}^2_{\mathsf{k_m}}(\langle \mathsf{n}^{j}\,,\, \mathsf{suc}(\pi_2(\mathsf{dec}(\pi_1(g(\phi^{\mathsf{in}}_{\tau_0})), \mathsf{sk}_{\mathrm{N}})))\rangle) \\
&\to \bigvee_{\tau_0 = _, \mathrm{PN}(j_0,1)\atop \tau_0 \prec \tau} \mathsf{accept}^{\mathrm{ID}}_{\tau_0} \wedge \begin{array}{l} \langle \sigma^{\mathsf{in}}_{\tau}(\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}})\,,\, \sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle \;=\; \\ \langle \mathsf{n}^{j}\,,\, \mathsf{suc}(\pi_2(\mathsf{dec}(\pi_1(g(\phi^{\mathsf{in}}_{\tau_0})), \mathsf{sk}_{\mathrm{N}})))\rangle \end{array} \qquad (\mathrm{CR}^2) \\
&\to \bigvee_{\tau_0 = _, \mathrm{PN}(j_0,1)\atop \tau_0 \prec \tau} \begin{array}{l} \mathsf{accept}^{\mathrm{ID}}_{\tau_0} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^{j} \\ \wedge\; \sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{suc}(\pi_2(\mathsf{dec}(\pi_1(g(\phi^{\mathsf{in}}_{\tau_0})), \mathsf{sk}_{\mathrm{N}}))) \end{array} \qquad \begin{pmatrix} \mathsf{EQInj}(\langle _\,,\, \cdot \rangle) \\ \text{and } \mathsf{EQInj}(\langle \cdot\,,\, _ \rangle) \end{pmatrix}
\end{aligned}$$

Part 2 It only remains to show that we can restrict ourselves to the τ_0 such that $\tau_1 \prec_{\tau} \tau_0$. Using **(Acc1)** we know that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau_0} \;\to\; \bigvee_{\tau' = \_, \mathrm{PU}_{\mathrm{ID}}(j',1)\atop \tau' \prec_{\tau} \tau_0} \pi_1(g(\phi^{\mathsf{in}}_{\tau_0})) = \{\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_{\tau'}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j'}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} \wedge g(\phi^{\mathsf{in}}_{\tau'}) = \mathsf{n}^{j_0}$$

Let $\tau' = \_, \mathrm{PU}_{\mathrm{ID}}(j', 1)$ such that $\tau' \prec_\tau \tau_0$. We now show that if $j' \neq j$ then the tests fail, which proves the impossibility of replaying an old message here. Assume $j' \neq j$, then:

$$\begin{aligned}
&\sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{suc}(\pi_2(\mathsf{dec}(\pi_1(g(\phi^{\mathsf{in}}_{\tau_0})), \mathsf{sk}_{\mathrm{N}}))) \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau_0})) = \{\langle \mathrm{ID}\,, \sigma^{\mathsf{in}}_{\tau'}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j'}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} \\
\rightarrow\ &\sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau'}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})) \\
\rightarrow\ &\mathsf{false} \qquad \text{(By } \textbf{(A3)}\text{)}
\end{aligned}$$

We deduce that:

$$\mathsf{accept}^{\mathrm{ID}}_\tau \rightarrow \bigvee_{\substack{\tau_0 = \_, \mathrm{PN}(j_0,1) \\ \tau_1 \prec_\tau \tau_0}} \mathsf{accept}^{\mathrm{ID}}_{\tau_0} \wedge g(\phi^{\mathsf{in}}_{\tau_1}) = \mathsf{n}^{j_0} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau_0})) = \{\langle \mathrm{ID}\,, \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} \qquad \blacksquare$$

Proof of ***(Acc3)***. Let $\mathsf{ai} = \mathrm{TU}_{\mathrm{ID}}(j, 1)$ and $\mathsf{k} \equiv \mathsf{k}^{\mathrm{ID}}$. We know that:

$$\mathsf{accept}^{\mathrm{ID}}_\tau \rightarrow \pi_3(g(\phi^{\mathsf{in}}_\tau)) = \mathsf{Mac}^3_{\mathsf{k_m}}(\langle \pi_1(g(\phi^{\mathsf{in}}_\tau))\,, \pi_2(g(\phi^{\mathsf{in}}_\tau)) \oplus \mathsf{f_k}(\pi_1(g(\phi^{\mathsf{in}}_\tau)))\,, \sigma^{\mathsf{in}}_\tau(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}})\rangle)$$

We are going to apply the P-EUF-MAC^3 axiom. We let S be the set of terms:

$$S = \{\tau_0 \mid \tau_0 = \_, \mathrm{TN}(j_0, 1) \prec \tau\}$$

and for all $S_0 \subseteq S$ we let:

$$b_{S_0} = \Big(\bigwedge_{\tau_0 \in S_0} \mathsf{accept}^{\mathrm{ID}}_{\tau_0}\Big) \wedge \Big(\bigwedge_{\tau_0 \in \overline{S_0}} \neg\mathsf{accept}^{\mathrm{ID}}_{\tau_0}\Big)$$

Then $(b_{S_0})_{S_0 \subseteq S}$ is a valid CS partition. It is straightforward to check that for every $S_0 \subseteq S$, for every $\tau_0 = \_, \mathrm{TN}(j_0, 1) \prec \tau$, if $\tau_0 \in S$ then we can rewrite $[b_{S_0}]t_{\tau_0}$ into a term $[b_{S_0}]t^{S_0}_{\tau_0}$ by removing the branch corresponding to $\mathsf{accept}^{\mathrm{ID}}_{\tau_0}$. Therefore:

$$\mathsf{Mac}^3_{\mathsf{k_m}}(\langle \mathsf{n}^{j_0}\,, \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})\,, \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}})\rangle) \in \mathsf{set\text{-}mac}^3_{\mathsf{k_m}}(t^{S_0}_{\tau_0}) \text{ if and only if } \tau_0 \in S_0$$

Hence by applying the P-EUF-MAC^3 axiom we get that:

$$\mathsf{accept}^{\mathrm{ID}}_\tau \rightarrow \bigvee_{S_0 \subseteq S} b_{S_0} \wedge \bigvee_{\tau_0 \in S_0} \pi_3(g(\phi^{\mathsf{in}}_\tau)) = \mathsf{Mac}^3_{\mathsf{k_m}}(\langle \mathsf{n}^{j_0}\,, \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})\,, \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}})\rangle)$$

By CR^3, $\mathsf{EQInj}(\langle \_\,, \cdot\rangle)$ and $\mathsf{EQInj}(\langle \cdot\,, \_\rangle)$ we know that for every $\tau_0 = \_, \mathrm{TN}(j_0, 1) \in S$:

$$\begin{aligned}
&\mathsf{Mac}^3_{\mathsf{k_m}}(\langle \pi_1(g(\phi^{\mathsf{in}}_\tau))\,, \pi_2(g(\phi^{\mathsf{in}}_\tau)) \oplus \mathsf{f_k}(\pi_1(g(\phi^{\mathsf{in}}_\tau)))\,, \sigma^{\mathsf{in}}_\tau(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}})\rangle) = \mathsf{Mac}^3_{\mathsf{k_m}}(\langle \mathsf{n}^{j_0}\,, \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})\,, \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}})\rangle) \\
\rightarrow\ &\pi_1(g(\phi^{\mathsf{in}}_\tau)) = \mathsf{n}^{j_0} \wedge \pi_2(g(\phi^{\mathsf{in}}_\tau)) \oplus \mathsf{f_k}(\pi_1(g(\phi^{\mathsf{in}}_\tau))) = \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \wedge \sigma^{\mathsf{in}}_\tau(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}})
\end{aligned}$$

Using the idempotence of the $\oplus$ we know that:

$$\big(\pi_1(g(\phi^{\mathsf{in}}_\tau)) = \mathsf{n}^{j_0} \wedge \pi_2(g(\phi^{\mathsf{in}}_\tau)) \oplus \mathsf{f_k}(\pi_1(g(\phi^{\mathsf{in}}_\tau))) = \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})\big) \rightarrow \pi_2(g(\phi^{\mathsf{in}}_\tau)) = \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \oplus \mathsf{f_k}(\mathsf{n}^{j_0})$$

Moreover, remark that if $S_0 \cap S_{\mathrm{N}} = \emptyset$, we have:

$$\neg \bigvee_{S_0 \subseteq S} b_{S_0} \wedge \bigvee_{\tau_0 \in S_0} \pi_3(g(\phi^{\mathsf{in}}_\tau)) = \mathsf{Mac}^3_{\mathsf{k_m}}(\langle \mathsf{n}^{j_0}\,, \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})\,, \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}})\rangle)$$

Putting everything together, we get that:

$$\begin{aligned}
\mathsf{accept}^{\mathrm{ID}}_\tau \rightarrow\ &\bigvee_{\substack{S_0 \subseteq S \\ S_0 \neq \emptyset}} b_{S_0} \wedge \bigvee_{\tau_0 \in S_0} \pi_3(g(\phi^{\mathsf{in}}_\tau)) = \mathsf{Mac}^3_{\mathsf{k_m}}(\langle \mathsf{n}^{j_0}\,, \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})\,, \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}})\rangle) \\
\rightarrow\ &\bigvee_{\substack{S_0 \subseteq S \\ S_0 \neq \emptyset}} \bigvee_{\tau_0 \in S_0} \mathsf{accept}^{\mathrm{ID}}_{\tau_0} \wedge \pi_3(g(\phi^{\mathsf{in}}_\tau)) = \mathsf{Mac}^3_{\mathsf{k_m}}(\langle \mathsf{n}^{j_0}\,, \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})\,, \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}})\rangle) \\
\rightarrow\ &\bigvee_{\tau_0 = _, \mathrm{TN}(j_0,0) \prec \tau} \mathsf{accept}^{\mathrm{ID}}_{\tau_0} \wedge \pi_3(g(\phi^{\mathsf{in}}_\tau)) = \mathsf{Mac}^3_{\mathsf{k_m}}(\langle \mathsf{n}^{j_0}\,, \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})\,, \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}})\rangle) \\
\rightarrow\ &\bigvee_{\tau_0 = _, \mathrm{TN}(j_0,0) \prec \tau} \mathsf{accept}^{\mathrm{ID}}_{\tau_0} \wedge \pi_1(g(\phi^{\mathsf{in}}_\tau)) = \mathsf{n}^{j_0} \wedge \pi_2(g(\phi^{\mathsf{in}}_\tau)) = \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \oplus \mathsf{f_k}(\mathsf{n}^{j_0}) \\
&\qquad\qquad \wedge \sigma^{\mathsf{in}}_\tau(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}})
\end{aligned} \qquad \blacksquare$$

*Proof of **(Acc4)**.* We are going to apply the P-EUF-MAC^4 axiom. We let $S = \{\tau_0 \mid \tau_0 = \_, \mathrm{TU}_{\mathrm{ID}}(j_0, 1) \prec \tau\}$, and for all $S_0 \subseteq S$ we let :

$$b_{S_0} = \bigwedge_{\tau_0 \in S_0} \mathsf{accept}^{\mathrm{ID}}_{\tau_0} \wedge \bigwedge_{\tau_0 \in \overline{S_0}} \neg \mathsf{accept}^{\mathrm{ID}}_{\tau_0}$$

Then $(b_{S_0})_{S_0 \subseteq S}$ is a valid CS partition. It is straightforward to check that for every $S_0 \subseteq S$, for every $\tau_0 = \_, \mathrm{TU}_{\mathrm{ID}}(j_0, 1) \prec \tau$:

$$[b_{S_0}]t_{\tau_0} = \begin{cases} [b_{S_0}]\mathsf{Mac}^4_{\mathsf{k_m}}(\pi_1(g(\phi^{\mathsf{in}}_{\tau_0}))) & \text{if } \tau_0 \in S_0 \\ [b_{S_0}]\mathsf{error} & \text{if } \tau_0 \in \overline{S_0} \end{cases}$$

Hence by applying the P-EUF-MAC^4 axiom we get that:

$$g(\phi^{\mathsf{in}}_\tau) = \mathsf{Mac}^4_{\mathsf{k_m}}(\mathsf{n}^j) \rightarrow \bigvee_{S_0 \subseteq S} b_{S_0} \wedge \bigvee_{\tau_0 \in S_0} g(\phi^{\mathsf{in}}_\tau) = \mathsf{Mac}^4_{\mathsf{k_m}}(\pi_1(g(\phi^{\mathsf{in}}_{\tau_0})))$$

Remark that for $S_0 = \emptyset$, we have:

$$\neg\Big(b_{S_0} \wedge \textstyle\bigvee_{\tau_0 \in S_0} g(\phi^{\mathsf{in}}_\tau) = \mathsf{Mac}^4_{\mathsf{k_m}}(\pi_1(g(\phi^{\mathsf{in}}_{\tau_0})))\Big)$$

Hence:

$$g(\phi^{\mathsf{in}}_\tau) = \mathsf{Mac}^4_{\mathsf{k_m}}(\mathsf{n}^j) \rightarrow \bigvee_{\substack{S_0 \subseteq S \\ S_0 \neq \emptyset}} b_{S_0} \wedge \bigvee_{\tau_0 \in S_0} g(\phi^{\mathsf{in}}_\tau) = \mathsf{Mac}^4_{\mathsf{k_m}}(\pi_1(g(\phi^{\mathsf{in}}_{\tau_0})))$$

Let $S_0 \subseteq S$ with $S_0 \neq \emptyset$, and let $\tau_0 \in S_0$. Using the CR^4 axiom we know that:

$$g(\phi^{\mathsf{in}}_\tau) = \mathsf{Mac}^4_{\mathsf{k_m}}(\underline{\mathsf{n}^j}) \ \wedge \ g(\phi^{\mathsf{in}}_\tau) = \mathsf{Mac}^4_{\mathsf{k_m}}(\underline{\pi_1(g(\phi^{\mathsf{in}}_{\tau_0}))}) \ \rightarrow \ \underline{\pi_1(g(\phi^{\mathsf{in}}_{\tau_0})) = \mathsf{n}^j}$$

Therefore:

$$\begin{aligned} g(\phi^{\mathsf{in}}_\tau) = \mathsf{Mac}^4_{\mathsf{k_m}}(\mathsf{n}^j) \rightarrow & \bigvee_{\substack{S_0 \subseteq S \\ S_0 \neq \emptyset}} b_{S_0} \wedge \bigvee_{\tau_0 \in S_0} g(\phi^{\mathsf{in}}_\tau) = \mathsf{Mac}^4_{\mathsf{k_m}}(\pi_1(g(\phi^{\mathsf{in}}_{\tau_0}))) \\ \rightarrow & \bigvee_{\substack{S_0 \subseteq S \\ S_0 \neq \emptyset}} b_{S_0} \wedge \bigvee_{\tau_0 \in S_0} \pi_1(g(\phi^{\mathsf{in}}_{\tau_0})) = \mathsf{n}^j \end{aligned}$$

And using the fact that $b_{S_0} \rightarrow \mathsf{accept}^{\mathrm{ID}}_{\tau_0}$:

$$g(\phi^{\mathsf{in}}_\tau) = \mathsf{Mac}^4_{\mathsf{k_m}}(\mathsf{n}^j) \rightarrow \bigvee_{\tau_0 = \_, \mathrm{TU}_{\mathrm{ID}}(\_,1) \prec \tau} \mathsf{accept}_{\tau_0} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau_0})) = \mathsf{n}^j$$

■

4.10 Acceptance Condition Characterizations

In this section, we prove acceptance characterizations, i.e. *necessary and sufficient* conditions for a message to be accepted by the user or the network. This section is organized as follows: we start by showing some properties of the AKA$^+$ protocol, which we use to obtain a first set of acceptance characterizations in Section 4.10.1 and Section 4.10.3; then, using these conditions, we prove in Section 4.10.5 that the temporary identity $\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}$ is concealed until the subscriber starts a session of the GUTI sub-protocol; finally, using the GUTI concealment property, we prove stronger acceptance characterizations in Section 4.10.6.

4.10.1 A First Acceptance Condition Characterization

Before proving our first acceptance characterizations, we show two properties of the AKA$^+$ protocol.

The property **(B1)** states that the user and network sequence numbers are increasing, i.e. for every valid action trace τ, and for every prefixes τ_1, τ_0 of τ such that $\tau_0 \preceq \tau_1$, we have:

$$\sigma_{\tau_0}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) \leq \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) \qquad \sigma_{\tau_0}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \leq \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$$

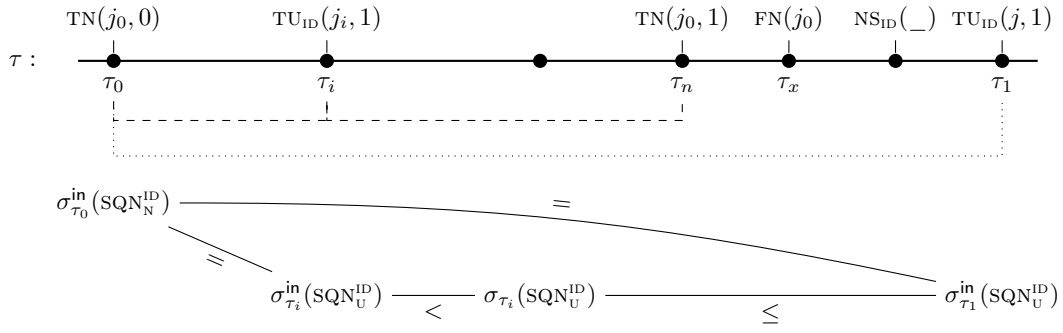


Figure 4.17: Graphical Representation of the Proof of Proposition 4.16

The property **(B2)** is more complex. Let j_0 be a network session that authenticated a user at instant τ (i.e. $\sigma_\tau(\mathsf{e\text{-}auth}^{j_0}_{\mathrm{N}}) \neq \mathsf{UnknownId}$), and let ID be a user. We assume that ID has been reseted since the session j_0, and that ID already ran a full session of the AKA$^+$ protocol: formally, $\tau = \mathrm{FU}_{\mathrm{ID}}(\_)$ and $\mathrm{FN}(j_0) \prec_\tau \mathrm{NS}_{\mathrm{ID}}(\_)$. Then either the *HN* session j_0 did not authenticate ID, or the current value value of $\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}$ is not n^{j_0}. In both case we have $\neg\mathsf{inj\text{-}auth}_\tau(\mathrm{ID}, j_0)$.

We show **(B2)** by contradiction, by proving that if it does not hold, then there is a sequence number *inconsistency*. In that case, we prove that there exists an instant τ_1 such that $\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) < \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$. We describe in an informal fashion how this is done. First, we prove that when a message is accepted, the user and network sequence numbers must be equal between some instants of the protocol executions (we prove two such equalities). Moreover, the sequence number are not decreasing **(B1)**, and the user increments his sequence number at the instant $\mathrm{TU}_{\mathrm{ID}}(\_, 1)$ if it accepts. This allows us to obtain the situation depicted in Figure 4.17. We will use this proof technique multiple times in this chapter.

Proposition 4.16. *For every valid action trace $\tau = \_, \mathsf{ai}$ on $\mathcal{S}_{\mathsf{id}}$ and identity $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$:*

- ***(B1)** For every $\tau_0 \preceq \tau_1 \preceq \tau$, for every $\mathrm{X} \in \{\mathrm{U}, \mathrm{N}\}$, we have $\sigma_{\tau_0}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{X}}) \leq \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{X}})$.*
- ***(B2)** If $\mathsf{ai} = \mathrm{FU}_{\mathrm{ID}}(j)$ then for every and $j_0 \in \mathbb{N}$, if $\mathrm{FN}(j_0) \prec_\tau \mathrm{NS}_{\mathrm{ID}}(\_)$ then:*

$$\sigma_\tau(\mathit{e\text{-}auth}^{j_0}_{\mathrm{N}}) \neq \mathit{UnknownId} \rightarrow \neg\mathit{inj\text{-}auth}_\tau(\mathrm{ID}, j_0)$$

Proof ⋆(p. 106). Let $\tau = \_, \mathsf{ai}$ be a valid action trace and $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$. The property **(B1)** is straightforward by induction over τ_1. Therefore, we focus on **(B2)**.

Let $\tau_x = \_, \mathrm{FN}(j_0) \prec \tau$. We do a case disjunction on the sub-protocol used by the user:

- If there exists τ_1 s.t. $\tau_1 = \_, \mathrm{TU}_{\mathrm{ID}}(j, 1) \prec \tau$. By validity of τ, there exists $\tau_n \prec \tau_x$ with $\tau_n = \_, \mathrm{PN}(j_0, 1)$ or $\_, \mathrm{TN}(j_0, 1)$. We can check that $\tau_n \prec \tau_x \prec \tau_1 \prec \tau$.
  Assume that $\tau_n = \_, \mathrm{PN}(j_0, 1)$. The sub-protocols used by the user and the network are different. In that case, it is very easy to show that we cannot have authentication. To prove this formally, observe first that $\mathsf{inj\text{-}auth}_\tau(\mathrm{ID}, j_0) \rightarrow \mathsf{accept}^{\mathrm{ID}}_{\tau_1}$. Therefore, using **(Acc3)**:

$$\mathsf{inj\text{-}auth}_\tau(\mathrm{ID}, j_0) \rightarrow \bigvee_{\substack{\tau_2 = \_, \mathrm{TN}(j_2, 0) \\ \tau_2 \prec \tau_1}} \sigma_{\tau_1}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^{j_2}$$

  For every $\tau_2 = \_, \mathrm{TN}(j_2, 0) \prec \tau_1$, we know that $j_2 \neq j_0$ (since $\tau_n = \_, \mathrm{PN}(j_0, 1)$). Hence:

$$\mathsf{inj\text{-}auth}_\tau(\mathrm{ID}, j_0) \rightarrow \sigma_{\tau_1}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) \neq \mathsf{n}^{j_0} \rightarrow \mathsf{false}$$

  Which is what we wanted.

  Now, assume that $\tau_n = \_, \mathrm{TN}(j_0, 1)$. We give a graphical representation of this case in Figure 4.17. The idea is that $\mathsf{inj\text{-}auth}_\tau(\mathrm{ID}, j_0)$ implies that $UE_{\mathrm{ID}}(j)$ must have accepted $HN(j_0)$ at instant τ_1. But since $HN(j_0)$ ran the GUTI sub-protocol at instant τ_n which is *before* τ_1, is must have accepted messages from a prior $UE_{\mathrm{ID}}(j_i)$ session (with $j_i \neq j$). It follows that $HN(j_0)$ must have accepted two different UE_{ID} sessions, j_i and j. This will yield a contradiction on sequence numbers.

We now prove this formally. First, observe that $\sigma_{\tau_n}(\mathsf{e\text{-}auth}_{\mathrm{N}}^{j_0}) \neq \mathsf{fail}$ and $\sigma_{\tau}(\mathsf{e\text{-}auth}_{\mathrm{N}}^{j_0}) = \sigma_{\tau_n}(\mathsf{e\text{-}auth}_{\mathrm{N}}^{j_0})$. Moreover, it is straightforward to check that for every valid action trace τ':

$$\mathsf{inj\text{-}auth}_{\tau'}(\mathrm{ID}, j_0) \wedge \sigma_{\tau'}(\mathsf{e\text{-}auth}_{\mathrm{N}}^{j_0}) \neq \mathsf{UnknownId} \wedge \sigma_{\tau'}(\mathsf{e\text{-}auth}_{\mathrm{N}}^{j_0}) \neq \mathsf{fail} \\ \rightarrow \sigma_{\tau'}(\mathsf{e\text{-}auth}_{\mathrm{N}}^{j_0}) = \sigma_{\tau'}(\mathsf{b\text{-}auth}_{\mathrm{N}}^{j_0})$$

Hence we deduce that:

$$\mathsf{inj\text{-}auth}_{\tau}(\mathrm{ID}, j_0) \wedge \sigma_{\tau}(\mathsf{e\text{-}auth}_{\mathrm{N}}^{j_0}) \neq \mathsf{UnknownId} \rightarrow \sigma_{\tau}(\mathsf{e\text{-}auth}_{\mathrm{N}}^{j_0}) = \sigma_{\tau}(\mathsf{b\text{-}auth}_{\mathrm{N}}^{j_0})$$

Since $\mathsf{inj\text{-}auth}_{\tau}(\mathrm{ID}, j_0) \rightarrow \sigma_{\tau}(\mathsf{b\text{-}auth}_{\mathrm{N}}^{j_0}) = \mathrm{ID}$, we get that:

$$\mathsf{inj\text{-}auth}_{\tau}(\mathrm{ID}, j_0) \wedge \sigma_{\tau}(\mathsf{e\text{-}auth}_{\mathrm{N}}^{j_0}) \neq \mathsf{UnknownId} \rightarrow \sigma_{\tau}(\mathsf{e\text{-}auth}_{\mathrm{N}}^{j_0}) = \mathrm{ID}$$

Moreover, $\sigma_{\tau}(\mathsf{e\text{-}auth}_{\mathrm{N}}^{j_0}) = \mathrm{ID} \rightarrow \mathsf{accept}_{\tau_n}^{\mathrm{ID}}$. Using **(Acc4)** on τ_n:

$$\mathsf{accept}_{\tau_n}^{\mathrm{ID}} \rightarrow \bigvee_{\tau_i = \_, \mathrm{TU}_{\mathrm{ID}}(j_i,1) \prec \tau_n} \mathsf{accept}_{\tau_i}^{\mathrm{ID}} \wedge \pi_1(g(\phi_{\tau_i}^{\mathsf{in}})) = \mathsf{n}^{j_0}$$

Let $\tau_0 = \mathrm{TN}(j_0, 0)$ and $\tau_i = \_, \mathrm{TU}_{\mathrm{ID}}(j_i, 1) \prec \tau_n$. Observe that $\tau_i \neq \tau_1$. Using **(Acc3)**, we get that:

$$\mathsf{accept}_{\tau_i}^{\mathrm{ID}} \wedge \pi_1(g(\phi_{\tau_i}^{\mathsf{in}})) = \mathsf{n}^{j_0} \rightarrow \mathsf{range}(\sigma_{\tau_i}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}), \sigma_{\tau_0}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}})) \rightarrow \sigma_{\tau_i}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) = \sigma_{\tau_0}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}})$$

Recall that $\mathsf{inj\text{-}auth}_{\tau}(\mathrm{ID}, j_0) \rightarrow \mathsf{accept}_{\tau_1}^{\mathrm{ID}}$. Moreover, $\mathsf{inj\text{-}auth}_{\tau}(\mathrm{ID}, j_0) \rightarrow \pi_1(g(\phi_{\tau_1}^{\mathsf{in}})) = \mathsf{n}^{j_0}$. Hence using **(Acc3)** again we get:

$$\mathsf{accept}_{\tau_1}^{\mathrm{ID}} \wedge \pi_1(g(\phi_{\tau_1}^{\mathsf{in}})) = \mathsf{n}^{j_0} \rightarrow \mathsf{range}(\sigma_{\tau_1}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}), \sigma_{\tau_0}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}})) \rightarrow \sigma_{\tau_1}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) = \sigma_{\tau_0}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}})$$

Putting everything together:

$$\mathsf{inj\text{-}auth}_{\tau}(\mathrm{ID}, j_0) \wedge \sigma_{\tau}(\mathsf{e\text{-}auth}_{\mathrm{N}}^{j_0}) \neq \mathsf{UnknownId} \rightarrow \begin{pmatrix} \sigma_{\tau_i}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) = \sigma_{\tau_0}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) \\ \wedge \ \sigma_{\tau_1}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) = \sigma_{\tau_0}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) \end{pmatrix} \\ \rightarrow \ \sigma_{\tau_1}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) = \sigma_{\tau_i}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})$$

Finally, $\mathsf{accept}_{\tau_i}^{\mathrm{ID}} \rightarrow \sigma_{\tau_i}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) < \sigma_{\tau_i}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})$, and using **(B1)** we know that $\sigma_{\tau_i}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) \leq \sigma_{\tau_1}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})$. We deduce that:

$$\mathsf{inj\text{-}auth}_{\tau}(\mathrm{ID}, j_0) \wedge \sigma_{\tau}(\mathsf{e\text{-}auth}_{\mathrm{N}}^{j_0}) \neq \mathsf{UnknownId} \rightarrow \sigma_{\tau_1}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) = \sigma_{\tau_i}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) < \sigma_{\tau_1}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) \\ \rightarrow \mathsf{false}$$

This concludes this case.

- If there exists $\tau_1 = \_, \mathrm{PU}_{\mathrm{ID}}(j, 2) \prec \tau$. Let $\tau_3 = \_, \mathrm{PU}_{\mathrm{ID}}(j, 1) \prec \tau_1$, we know that $\tau_x \prec \tau_3$. Remark that $\mathsf{inj\text{-}auth}_{\tau}(\mathrm{ID}, j_0) \rightarrow \mathsf{accept}_{\tau_1}^{\mathrm{ID}}$, and using **(Acc2)** we easily get that:

$$\mathsf{accept}_{\tau_1}^{\mathrm{ID}} \rightarrow \bigvee_{\substack{\tau_2 = \_, \mathrm{PN}(j_2,1) \\ \tau_3 \prec \tau_2 \prec \tau_1}} \sigma_{\tau_1}(\mathsf{e\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}) = \mathsf{n}^{j_2}$$

Since no ID action occurred between τ_1 and τ, we have $\sigma_{\tau_1}(\mathsf{e\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}) = \sigma_{\tau}(\mathsf{e\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}})$. Moreover, $\mathsf{inj\text{-}auth}_{\tau}(\mathrm{ID}, j_0) \rightarrow \sigma_{\tau}(\mathsf{e\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}) = \mathsf{n}^{j_0}$. Finally, for every $\tau_2 = \_, \mathrm{PN}(j_2, 1)$ such that $\tau_3 \prec \tau_2 \prec \tau_1$, since $\tau_x \prec \tau_3$ we know that $j_2 \neq j_0$. It follows that:

$$\mathsf{inj\text{-}auth}_{\tau}(\mathrm{ID}, j_0) \rightarrow \bigvee_{\substack{\tau_2 = \_, \mathrm{PN}(j_2,1) \\ \tau_3 \prec \tau_2 \prec \tau_1}} \mathsf{n}^{j_0} = \mathsf{n}^{j_2} \rightarrow \mathsf{false}$$ ■

We now prove a first acceptance characterization:

Lemma 4.10. *For every valid action trace $\tau = \_, a i$ on $\mathcal{S}_{id}$ and identity $\mathrm{ID} \in \mathcal{S}_{id}$:*

- ***(Equ1)*** *If* $\mathsf{ai} = \mathrm{FU}_{\mathrm{ID}}(j)$. *For every* $\tau_0 = \_, \mathrm{FN}(j_0) \prec \tau$, *we let:*

$$\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_0}_{\mathsf{u}:\tau} \equiv \begin{pmatrix} \mathsf{inj\text{-}auth}_\tau(\mathrm{ID}, j_0) \wedge \sigma^{\mathsf{in}}_\tau(\mathsf{e\text{-}auth}^{j_0}_{\mathrm{N}}) \neq \mathsf{UnknownId} \\ \wedge\ \pi_1(g(\phi^{\mathsf{in}}_\tau)) = \mathrm{GUTI}^{j_0} \oplus \mathsf{f}^{\mathsf{r}}_{\mathsf{k}}(\mathsf{n}^{j_0}) \wedge\ \pi_2(g(\phi^{\mathsf{in}}_\tau)) = \mathsf{Mac}^5_{\mathsf{k_m}}(\langle \mathrm{GUTI}^{j_0}\,,\, \mathsf{n}^{j_0}\rangle) \end{pmatrix}$$

Then:

$$\mathsf{accept}^{\mathrm{ID}}_\tau \leftrightarrow \bigvee_{\substack{\tau_0 = \_, \mathrm{FN}(j_0) \prec \tau \\ \tau_0 \not\prec_\tau \mathrm{NS}_{\mathrm{ID}}(\_)}} \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_0}_{\mathsf{u}:\tau}$$

Proof ★*(p. 107).* Using Lemma 4.9 we know that:

$$\mathsf{suc\text{-}auth}_\tau(\mathrm{ID}) \rightarrow \bigvee_{\mathsf{s\text{-}started}_{j_0}(\tau)} \mathsf{inj\text{-}auth}_\tau(\mathrm{ID}, j_0)$$

Let $\mathsf{k} \equiv \mathsf{k}_{\mathrm{ID}}$ and $\mathsf{k_m} \equiv \mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}$. Since:

$$\mathsf{accept}^{\mathrm{ID}}_\tau \equiv \mathsf{suc\text{-}auth}_\tau(\mathrm{ID}) \wedge \underbrace{\pi_2(g(\phi^{\mathsf{in}}_\tau)) = \mathsf{Mac}^5_{\mathsf{k_m}}(\langle \pi_1(g(\phi^{\mathsf{in}}_\tau)) \oplus \mathsf{f}^{\mathsf{r}}_{\mathsf{k}}(\sigma^{\mathsf{in}}_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}))\,,\, \sigma^{\mathsf{in}}_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}})\rangle)}_{\mathsf{EQMac}}$$

And since $\mathsf{inj\text{-}auth}_\tau(\mathrm{ID}, j_0) \rightarrow \mathsf{suc\text{-}auth}_\tau(\mathrm{ID})$ we have:

$$\begin{aligned} \mathsf{accept}^{\mathrm{ID}}_\tau &\leftrightarrow \bigvee_{\mathsf{s\text{-}started}_{j_0}(\tau)} \mathsf{inj\text{-}auth}_\tau(\mathrm{ID}, j_0) \wedge \mathsf{EQMac} \\ &\leftrightarrow \bigvee_{\mathsf{s\text{-}started}_{j_0}(\tau)} \mathsf{inj\text{-}auth}_\tau(\mathrm{ID}, j_0) \wedge \pi_2(g(\phi^{\mathsf{in}}_\tau)) = \mathsf{Mac}^5_{\mathsf{k_m}}(\langle \pi_1(g(\phi^{\mathsf{in}}_\tau)) \oplus \mathsf{f}^{\mathsf{r}}_{\mathsf{k}}(\mathsf{n}^{j_0})\,,\, \mathsf{n}^{j_0}\rangle) \end{aligned}$$

Using the P-EUF-MAC^5 and CR^5 axioms, it is easy to show that for every $j_0 \in \mathbb{N}$:

$$\pi_2(g(\phi^{\mathsf{in}}_\tau)) = \mathsf{Mac}^5_{\mathsf{k_m}}(\langle \pi_1(g(\phi^{\mathsf{in}}_\tau)) \oplus \mathsf{f}^{\mathsf{r}}_{\mathsf{k}}(\mathsf{n}^{j_0})\,,\, \mathsf{n}^{j_0}\rangle) \rightarrow \begin{cases} \begin{pmatrix} \pi_1(g(\phi^{\mathsf{in}}_\tau)) \oplus \mathsf{f}^{\mathsf{r}}_{\mathsf{k}}(\mathsf{n}^{j_0}) = \mathrm{GUTI}^{j_0} \\ \wedge\ \sigma^{\mathsf{in}}_\tau(\mathsf{e\text{-}auth}^{j_0}_{\mathrm{N}}) \neq \mathsf{UnknownId} \end{pmatrix} & \text{if } \mathrm{FN}(j_0) \in \tau \\ \mathsf{false} & \text{otherwise} \end{cases}$$

Hence:

$$\begin{aligned} \mathsf{accept}^{\mathrm{ID}}_\tau &\leftrightarrow \bigvee_{\tau_0 = \_, \mathrm{FN}(j_0) \prec \tau} \begin{pmatrix} \mathsf{inj\text{-}auth}_\tau(\mathrm{ID}, j_0) \wedge \sigma^{\mathsf{in}}_\tau(\mathsf{e\text{-}auth}^{j_0}_{\mathrm{N}}) \neq \mathsf{UnknownId} \\ \wedge\ \pi_1(g(\phi^{\mathsf{in}}_\tau)) = \mathrm{GUTI}^{j_0} \oplus \mathsf{f}^{\mathsf{r}}_{\mathsf{k}}(\mathsf{n}^{j_0}) \wedge \pi_2(g(\phi^{\mathsf{in}}_\tau)) = \mathsf{Mac}^5_{\mathsf{k_m}}(\langle \mathrm{GUTI}^{j_0}\,,\, \mathsf{n}^{j_0}\rangle) \end{pmatrix} \\ &\leftrightarrow \bigvee_{\tau_0 = \_, \mathrm{FN}(j_0) \prec \tau} \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_0}_{\mathsf{u}:\tau} \end{aligned}$$

We conclude using **(B2)**:

$$\mathsf{accept}^{\mathrm{ID}}_\tau \leftrightarrow \bigvee_{\substack{\tau_0 = \_, \mathrm{FN}(j_0) \prec \tau \\ \tau_0 \not\prec_\tau \mathrm{NS}_{\mathrm{ID}}(\_)}} \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_0}_{\mathsf{u}:\tau}$$ ■

Using this acceptance characterization, we prove additional properties of the protocol:

- **(B3)**: if the user has a valid temporary identity (i.e. $\sigma_\tau(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}})$), then the variable $\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}$ is not unset.
- **(B4)**: if the network sequence number for ID increased between two instants τ_2 and τ_1, then this increase has been recorded by the variable $\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}$: there must exists an instant τ_x between τ_2 and τ_1 such that $\sigma^{\mathsf{in}}_{\tau_1}(\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}) = \mathsf{n}^{j_x}$, where τ_x ends by $\mathrm{TN}(j_x, 0)$, $\mathrm{TN}(j_x, 1)$ or $\mathrm{PN}(j_x, 1)$.
- **(B5)**: the network sequence number is always smaller than the user sequence number: for every ID, we have $\sigma_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \leq \sigma_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$.
- **(B6)**: if τ_0 is the last reset of user ID (i.e. $\tau_0 = \_, \mathrm{NS}_{\mathrm{ID}}(\_) \prec \tau$ and $\tau_0 \not\prec_\tau \mathrm{NS}_{\mathrm{ID}}(\_)$), and if ID is synced at an instant τ_1 between τ_0 and τ, then the *network* sequence number at instant τ_1 is greater than the *user* sequence number at the time of the reset (i.e. at τ_0).
- **(B7)**: if no ASSIGN-GUTI session took place since the last reset of user ID, then ID has no valid temporary identity.

Proposition 4.17. *For every valid action trace $\tau = \_, ai$ on $\mathcal{S}_{id}$ and identity $\mathrm{ID} \in \mathcal{S}_{id}$:*

- ***(B3)*** $\sigma_\tau(\mathsf{valid\text{-}guti}_{\mathrm{U}}^{\mathrm{ID}}) \rightarrow \sigma_\tau(\mathrm{GUTI}_{\mathrm{U}}^{\mathrm{ID}}) \neq \mathit{UnSet}$.
- ***(B4)*** *For every $\tau_2 \prec_\tau \tau_1$:*

$$\sigma_{\tau_2}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) < \sigma_{\tau_1}^{in}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) \rightarrow \bigvee_{\substack{\tau_2 \prec_\tau \tau_x \prec_\tau \tau_1 \\ \tau_x = \_, \mathrm{TN}(j_x,0), \_, \mathrm{TN}(j_x,1) \text{ or } \_, \mathrm{PN}(j_x,1)}} \sigma_{\tau_1}^{in}(\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}}) = \mathsf{n}^{j_x}$$

- ***(B5)*** $\sigma_\tau(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) \leq \sigma_\tau(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})$.
- ***(B6)*** *For every $\tau_0 \prec_\tau \tau_1$ such that $\tau_0 = \_, \mathrm{NS}_{\mathrm{ID}}(\_)$ or ϵ, and such that $\tau_0 \not\prec_\tau \mathrm{NS}_{\mathrm{ID}}(\_)$, we have:*

$$\sigma_{\tau_1}(\mathsf{sync}_{\mathrm{U}}^{\mathrm{ID}}) \rightarrow \sigma_{\tau_1}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) > \sigma_{\tau_0}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})$$

- ***(B7)*** *If for all $\tau' \preceq \tau$ such that $\tau' \not\prec_\tau \mathrm{NS}_{\mathrm{ID}}(\_)$ we have $\tau' \neq \_, \mathrm{FU}_{\mathrm{ID}}(\_)$, then:*

$$\sigma_\tau(\mathsf{valid\text{-}guti}_{\mathrm{U}}^{\mathrm{ID}}) \rightarrow \mathit{false}$$

4.10.2 ⋆ (p. 111) Proof of Proposition 4.17

*Proof of **(B3)**.* We show this by induction over τ. If $\tau = \epsilon$, we know from Definition 4.4 that $\sigma_\epsilon(\mathsf{valid\text{-}guti}_{\mathrm{U}}^{\mathrm{ID}}) \equiv \mathsf{false}$ and $\sigma_\epsilon(\mathrm{GUTI}_{\mathrm{X}}^{\mathrm{ID}}) \equiv \mathsf{UnSet}$. Therefore the property holds. Let $\tau = \tau_0, \mathsf{ai}$, assume by induction that the property holds for τ_0. If ai is different from $\mathrm{TU}_{\mathrm{ID}}(j,0), \mathrm{PU}_{\mathrm{ID}}(j,1)$ and $\mathrm{FU}(j)$ then $\sigma_\tau^{\mathsf{up}}(\mathsf{valid\text{-}guti}_{\mathrm{U}}^{\mathrm{ID}}) \equiv \sigma_\tau^{\mathsf{up}}(\mathrm{GUTI}_{\mathrm{U}}^{\mathrm{ID}}) \equiv \bot$, in which case we conclude immediately by induction hypothesis. We have three cases remaining:

- If $\mathsf{ai} = \mathrm{TU}_{\mathrm{ID}}(j,0)$ or $\mathsf{ai} = \mathrm{PU}_{\mathrm{ID}}(j,1)$ then $\sigma_\tau^{\mathsf{up}}(\mathrm{GUTI}_{\mathrm{U}}^{\mathrm{ID}}) \equiv \mathsf{false}$. Therefore the property holds.
- If $\mathsf{ai} = \mathrm{FU}(j)$, using **(Equ1)** we can check that:

$$\mathsf{accept}_\tau^{\mathrm{ID}} \rightarrow \bigvee_{\substack{\tau_1 = \_, \mathrm{FN}(j_0) \prec \tau \\ \tau_1 \not\prec \tau \mathrm{NS}_{\mathrm{ID}}(\_)}} \sigma_\tau(\mathrm{GUTI}_{\mathrm{U}}^{\mathrm{ID}}) = \mathrm{GUTI}^{j_0} \rightarrow \sigma_\tau(\mathrm{GUTI}_{\mathrm{U}}^{\mathrm{ID}}) \neq \mathsf{UnSet}$$

We conclude by observing that $\sigma_\epsilon(\mathsf{valid\text{-}guti}_{\mathrm{U}}^{\mathrm{ID}}) \equiv \mathsf{accept}_\tau^{\mathrm{ID}}$. ■

*Proof of **(B4)**.* We prove this directly. Intuitively, this holds because if $\sigma_{\tau_2}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) < \sigma_{\tau_1}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}})$ then we know that $\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}$ was updated between τ_2 and τ_1. Moreover, if such an update occurs at $\tau_x = \_, \mathrm{PN}(j_x,1)$ or $\mathrm{TN}(j_x,1)$ then $\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}}$ has to be equal to n^{j_x} after the update. Finally, the fact that $\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}}$ is equal to n^{j_x} for some τ_x between τ_2 and τ_1 with $\tau_x = \_, \mathrm{TN}(j_x,0)$, $\_, \mathrm{TN}(j_x,1)$ or $\_, \mathrm{PN}(j_x,1)$ is an invariant of the protocol. Now we give the formal proof.

First, we remark that $\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}$ is updated only at $\mathrm{PN}(\_,1)$ and $\mathrm{TN}(\_,1)$. Moreover, each update either left $\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}$ unchanged or increments it by at least one. Finally, it is updated at $\tau_x \prec \tau$ if and only if $\mathsf{inc\text{-}accept}_{\tau_x}^{\mathrm{ID}}$ holds. If follows that:

$$\sigma_{\tau_2}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) < \sigma_{\tau_1}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) \rightarrow \bigvee_{\substack{\tau_2 \prec_\tau \tau_x \prec_\tau \tau_1 \\ \tau_x = , \_, \mathrm{TN}(j_x,1) \text{ or } \_, \mathrm{PN}(j_x,1)}} \mathsf{inc\text{-}accept}_{\tau_x}^{\mathrm{ID}}$$

We know that for every $\tau_2 \prec_\tau \tau_x \prec_\tau \tau_1$, if:

- $\tau_x = , \_, \mathrm{PN}(j_x,1)$ then $\mathsf{inc\text{-}accept}_{\tau_x}^{\mathrm{ID}} \rightarrow \sigma_{\tau_x}(\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}}) = \mathsf{n}^{j_x}$.
- $\tau_x = \_, \mathrm{TN}(j_x,1)$ then since $\mathsf{inc\text{-}accept}_{\tau_x}^{\mathrm{ID}} \equiv \mathsf{inc\text{-}accept}_{\tau_x}^{\mathrm{ID}} \wedge \sigma_{\tau_x}^{\mathsf{in}}(\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}}) = \mathsf{n}^{j_x}$, we know that $\mathsf{inc\text{-}accept}_{\tau_x}^{\mathrm{ID}} \rightarrow \sigma_{\tau_x}^{\mathsf{in}}(\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}}) = \mathsf{n}^{j_x}$. Besides, since $\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}}$ is not updated at $\mathrm{TN}(j_x,1)$ we deduce that $\mathsf{inc\text{-}accept}_{\tau_x}^{\mathrm{ID}} \rightarrow \sigma_{\tau_x}(\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}}) = \mathsf{n}^{j_x}$.

Hence:

$$\sigma_{\tau_2}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) < \sigma_{\tau_1}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) \rightarrow \bigvee_{\substack{\tau_2 \prec_\tau \tau_x \prec_\tau \tau_1 \\ \tau_x = , \_, \mathrm{TN}(j_x,1) \text{ or } \_, \mathrm{PN}(j_x,1)}} \sigma_{\tau_x}(\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}}) = \mathsf{n}^{j_x} \tag{4.9}$$

Let $\tau_2 \prec_\tau \tau_x \prec_\tau \tau_1$ such that $\tau_x = , \_, \mathrm{TN}(j_x,1)$ or $\_, \mathrm{PN}(j_x,1)$. Now, we prove by induction over τ' such that $\tau_x \preceq \tau' \prec \tau_1$ that:

$$\sigma_{\tau_x}(\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}}) = \mathsf{n}^{j_x} \rightarrow \bigvee_{\substack{\tau_x \preceq \tau_n \preceq \tau' \\ \tau_n = \_, \mathrm{TN}(j_n,0), \_, \mathrm{TN}(j_n,1) \text{ or } \_, \mathrm{PN}(j_n,1)}} \sigma_{\tau'}(\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}}) = \mathsf{n}^{j_n}$$

If $\tau' = \tau_x$ this is obvious. For the inductive case, we do a disjunction over the final action of τ'. If $\mathsf{session}^{\text{ID}}_{\text{N}}$ is not updated then we conclude by induction, otherwise we are in one of the following cases:

- If $\tau' = \_, \text{TN}(j', 0)$ then we do a case disjunction on $\mathsf{accept}^{\text{ID}}_{\tau'}$:

$$\neg\mathsf{accept}^{\text{ID}}_{\tau'} \rightarrow \sigma_{\tau'}(\mathsf{session}^{\text{ID}}_{\text{N}}) = \sigma^{\mathsf{in}}_{\tau'}(\mathsf{session}^{\text{ID}}_{\text{N}}) \tag{4.10}$$

  Hence:

$$\begin{aligned}
&\neg\mathsf{accept}^{\text{ID}}_{\tau'} \wedge \sigma_{\tau_x}(\mathsf{session}^{\text{ID}}_{\text{N}}) = \mathsf{n}^{j_x} \\
&\quad\rightarrow \bigvee_{\substack{\tau_x \preceq \tau_n \prec \tau' \\ \tau_n = _,\text{TN}(j_n,0), _,\text{TN}(j_n,1) \text{ or } _,\text{PN}(j_n,1)}} \sigma_{\tau'}(\mathsf{session}^{\text{ID}}_{\text{N}}) = \mathsf{n}^{j_n} && \text{(By induction hypothesis and (4.10))} \\
&\quad\rightarrow \bigvee_{\substack{\tau_x \preceq \tau_n \preceq \tau' \\ \tau_n = _,\text{TN}(j_n,0), _,\text{TN}(j_n,1) \text{ or } _,\text{PN}(j_n,1)}} \sigma_{\tau'}(\mathsf{session}^{\text{ID}}_{\text{N}}) = \mathsf{n}^{j_n} && \text{(Relaxing the condition } \tau_n \prec \tau')
\end{aligned}$$

  Moreover,

$$\mathsf{accept}^{\text{ID}}_{\tau'} \rightarrow \sigma_{\tau'}(\mathsf{session}^{\text{ID}}_{\text{N}}) = \mathsf{n}^{j'} \rightarrow \bigvee_{\substack{\tau_x \preceq \tau_n \preceq \tau' \\ \tau_n = \_,\text{TN}(j_n,0), \_,\text{TN}(j_n,1) \text{ or } \_,\text{PN}(j_n,1)}} \sigma_{\tau'}(\mathsf{session}^{\text{ID}}_{\text{N}}) = \mathsf{n}^{j_n}$$

  This concludes this case.

- If $\tau_n = \_, \text{PN}(j_n, 1)$ then the proof is the same than in the previous case, but doing a case disjunction over $\mathsf{inc\text{-}accept}^{\text{ID}}_{\tau'}$.

Let $\tau_0{}'$ be such that $\tau_1 = \tau_0{}', \mathsf{ai}_1$. By applying the induction hypothesis to $\tau_0{}'$, we get:

$$\sigma_{\tau_x}(\mathsf{session}^{\text{ID}}_{\text{N}}) = \mathsf{n}^{j_x} \rightarrow \bigvee_{\substack{\tau_x \preceq \tau_n \preceq \tau_0{}' \\ \tau_n = \_,\text{TN}(j_n,0), \_,\text{TN}(j_n,1) \text{ or } \_,\text{PN}(j_n,1)}} \sigma_{\tau_0{}'}(\mathsf{session}^{\text{ID}}_{\text{N}}) = \mathsf{n}^{j_n} \rightarrow \bigvee_{\substack{\tau_x \preceq \tau_n \prec \tau_1 \\ \tau_n = \_,\text{TN}(j_n,0), \_,\text{TN}(j_n,1) \text{ or } \_,\text{PN}(j_n,1)}} \sigma^{\mathsf{in}}_{\tau_1}(\mathsf{session}^{\text{ID}}_{\text{N}}) = \mathsf{n}^{j_n}$$

We conclude using (4.9) and the property above. ■

Proof of ***(B5)***. We prove this by induction over τ. For $\tau = \epsilon$, from Definition 4.4 we know that $\sigma_\epsilon(\text{SQN}^{\text{ID}}_{\text{U}}) \equiv \mathsf{sqn\text{-}init}^{\text{ID}}_{\text{U}}$ and $\sigma_\epsilon(\text{SQN}^{\text{ID}}_{\text{N}}) \equiv \mathsf{sqn\text{-}init}^{\text{ID}}_{\text{N}}$. Using SQN-ini, we know that $\mathsf{sqn\text{-}init}^{\text{ID}}_{\text{N}} \leq \mathsf{sqn\text{-}init}^{\text{ID}}_{\text{U}}$.

For the inductive case, let $\tau = \tau_0, \mathsf{ai}$ and assume that the property holds for τ_0. We have three cases:

- If ai is such that $\text{SQN}^{\text{ID}}_{\text{N}}$ is not updated. Using **(B1)** we know that $\sigma_\tau(\text{SQN}^{\text{ID}}_{\text{U}}) \geq \sigma_{\tau_0}(\text{SQN}^{\text{ID}}_{\text{U}})$, and we conclude by applying the induction hypothesis.
- If $\mathsf{ai} = \text{PN}(j, 1)$, then we do a case disjunction on $\mathsf{inc\text{-}accept}^{\text{ID}}_{\tau}$. If it is true then:

$$\begin{aligned}
\mathsf{inc\text{-}accept}^{\text{ID}}_{\tau} &\rightarrow \bigvee_{\tau_0 = _,\text{PU}_{\text{ID}}(j_0,1) \prec \tau} \sigma_\tau(\text{SQN}^{\text{ID}}_{\text{N}}) = \sigma^{\mathsf{in}}_{\tau_0}(\text{SQN}^{\text{ID}}_{\text{U}}) && \text{(By (Acc1))} \\
&\rightarrow \bigvee_{\tau_0 = _,\text{PU}_{\text{ID}}(j_0,1) \prec \tau} \sigma_\tau(\text{SQN}^{\text{ID}}_{\text{N}}) = \sigma^{\mathsf{in}}_{\tau_0}(\text{SQN}^{\text{ID}}_{\text{U}}) \wedge \sigma^{\mathsf{in}}_{\tau_0}(\text{SQN}^{\text{ID}}_{\text{U}}) \leq \sigma_\tau(\text{SQN}^{\text{ID}}_{\text{U}}) && \text{(By (B1))} \\
&\rightarrow \sigma_\tau(\text{SQN}^{\text{ID}}_{\text{N}}) \leq \sigma_\tau(\text{SQN}^{\text{ID}}_{\text{U}})
\end{aligned}$$

  If $\mathsf{inc\text{-}accept}^{\text{ID}}_{\tau}$ is false then $\neg\mathsf{inc\text{-}accept}^{\text{ID}}_{\tau} \rightarrow \sigma_\tau(\text{SQN}^{\text{ID}}_{\text{N}}) = \sigma^{\mathsf{in}}_{\tau}(\text{SQN}^{\text{ID}}_{\text{N}})$, and we conclude by applying the induction hypothesis.
- If $\mathsf{ai} = \text{TN}(j, 1)$, then we do a case disjunction on $\mathsf{inc\text{-}accept}^{\text{ID}}_{\tau}$. First we handle the case where it is true. We summarize graphically this case in Figure 4.18. Let $\tau_2 = \_, \text{TN}(j, 0) \prec \tau$. We know that $\mathsf{inc\text{-}accept}^{\text{ID}}_{\tau} \rightarrow \sigma^{\mathsf{in}}_{\tau}(\mathsf{session}^{\text{ID}}_{\text{N}}) = \mathsf{n}^{j}$. Moreover:

$$\begin{aligned}
\sigma^{\mathsf{in}}_{\tau}(\mathsf{session}^{\text{ID}}_{\text{N}}) = \mathsf{n}^{j} \rightarrow &\bigwedge_{\substack{\tau_2 \prec \tau_1 \prec \tau \\ \tau_1 = _,\text{TN}(j_x,0), _,\text{TN}(j_x,1) \text{ or } _,\text{PN}(j_x,1)}} \sigma^{\mathsf{in}}_{\tau}(\mathsf{session}^{\text{ID}}_{\text{N}}) \neq \mathsf{n}^{j_x} \\
&\rightarrow \sigma_{\tau_2}(\text{SQN}^{\text{ID}}_{\text{N}}) \leq \sigma^{\mathsf{in}}_{\tau}(\text{SQN}^{\text{ID}}_{\text{N}}) && \text{(Using the contrapositive of (B4))} \\
&\rightarrow \sigma_{\tau_2}(\text{SQN}^{\text{ID}}_{\text{N}}) = \sigma^{\mathsf{in}}_{\tau}(\text{SQN}^{\text{ID}}_{\text{N}}) && \text{(Using (B1))}
\end{aligned}$$

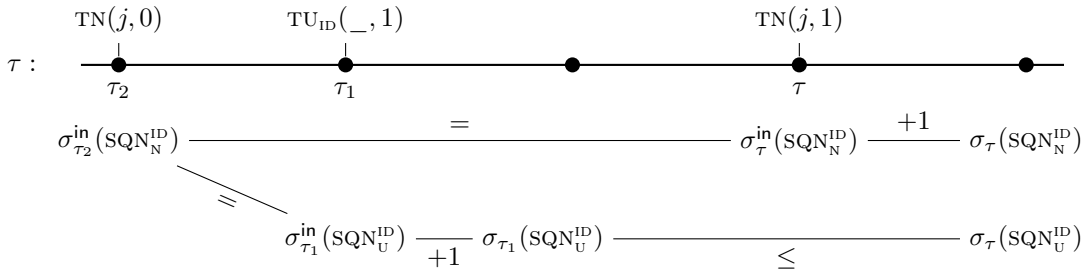


Figure 4.18: Graphical Representation Used in the Proof of **(B5)**.

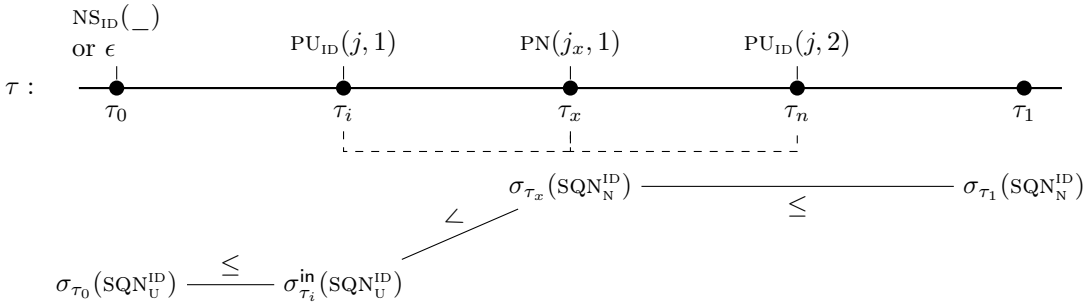


Figure 4.19: Graphical Representation Used in the Proof of **(B6)**.

We know that $\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau} \rightarrow \mathsf{accept}^{\mathrm{ID}}_{\tau}$. Moreover, using **(Acc3)** and **(Acc4)**, we check that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \rightarrow \bigvee_{\substack{\tau_1 = \_, \mathrm{TU}_{\mathrm{ID}}(\_,1) \\ \tau_2 \prec \tau_1 \prec \tau}} \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$$

Besides, $\mathsf{accept}^{\mathrm{ID}}_{\tau} \rightarrow \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) + 1$, and using **(B1)** we know that $\sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) \leq \sigma_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$. Finally, $\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau} \rightarrow \sigma_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = \sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) + 1$. Putting everything together:

$$\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau} \rightarrow \sigma_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \leq \sigma_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$$

Which is what we wanted.

If $\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau}$ is false then $\neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau} \rightarrow \sigma_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = \sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$, and we conclude by applying the induction hypothesis. ■

Proof of ***(B6)***. First, observe that:

$$\sigma_{\tau_1}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \rightarrow \bigvee_{\substack{\tau_n = \_, \mathrm{PU}_{\mathrm{ID}}(j,2) \\ \tau_0 \prec \tau_n \prec \tau_1}} \mathsf{accept}^{\mathrm{ID}}_{\tau_n} \tag{4.11}$$

Let $\tau_n = \_, \mathrm{PU}_{\mathrm{ID}}(j,2)$ such that $\tau_0 \prec \tau_n \prec \tau_1$. Let $\tau_i = \_, \mathrm{PU}_{\mathrm{ID}}(j,1)$ such that $\tau_i \prec \tau_n$. We know that $\tau_i \prec \tau_0$. We give a graphical summary of this proof in Figure 4.19. First, we apply **(Acc2)**:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau_n} \rightarrow \bigvee_{\substack{\tau_x = \_, \mathrm{PN}(j_x,1) \\ \tau_i \prec \tau_x \prec \tau_n}} \mathsf{accept}^{\mathrm{ID}}_{\tau_x} \wedge g(\phi^{\mathsf{in}}_{\tau_i}) = \mathsf{n}^{j_x} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau_x})) = \{\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_{\tau_i}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^j_{\epsilon}}_{\mathsf{pk}_{\mathrm{N}}} \tag{4.12}$$

Let $\tau_x = \_, \mathrm{PN}(j_x,1)$ such that $\tau_i \prec \tau_x \prec \tau_n$. Using **(B1)**, we get that $\sigma_{\tau_0}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) \leq \sigma^{\mathsf{in}}_{\tau_i}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$ and that $\sigma_{\tau_x}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \leq \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$. There are two cases, depending on whether we have $\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_x}$.

- We know that $\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_x} \rightarrow \sigma_{\tau_x}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = \sigma^{\mathsf{in}}_{\tau_i}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) + 1 > \sigma^{\mathsf{in}}_{\tau_i}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$. Putting everything together, we get that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau_n} \wedge \mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_x} \rightarrow \sigma_{\tau_0}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) < \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$$

- We know that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau_x} \wedge \neg \mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_x} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau_x})) = \{\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_{\tau_i}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^j_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} \rightarrow \sigma^{\mathsf{in}}_{\tau_i}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) < \sigma^{\mathsf{in}}_{\tau_x}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$$

  Moreover, $\neg \mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_x} \rightarrow \sigma^{\mathsf{in}}_{\tau_x}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = \sigma_{\tau_x}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$. We recall that $\sigma_{\tau_0}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) \leq \sigma^{\mathsf{in}}_{\tau_i}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$ and that $\sigma_{\tau_x}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \leq \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$. Therefore:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau_x} \wedge \neg \mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_x} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau_x})) = \{\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_{\tau_i}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^j_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} \rightarrow \sigma_{\tau_0}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) < \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$$

Using (4.12) and the two cases above, we get that $\mathsf{accept}^{\mathrm{ID}}_{\tau_n} \rightarrow \sigma_{\tau_0}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) < \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$.

Since this is true for all $\tau_n = \_, \mathrm{PU}_{\mathrm{ID}}(j,2)$ such that $\tau_0 \prec \tau_n \prec \tau_1$, we deduce from (4.11) that

$$\sigma_{\tau_1}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \rightarrow \sigma_{\tau_0}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) < \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$$

■

*Proof of **(B7)**.* Let $\tau_{\mathrm{NS}} = \epsilon$ or $\mathrm{NS}_{\mathrm{ID}}(\_)$ be such that $\tau_{\mathrm{NS}} \preceq \tau$ and $\tau_{\mathrm{NS}} \not\preceq_\tau \mathrm{NS}_{\mathrm{ID}}(\_)$. We show by induction over τ' with $\tau_{\mathrm{NS}} \preceq \tau' \preceq \tau$ that $\sigma_{\tau'}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \equiv \mathsf{false}$.

For $\tau' = \tau_{\mathrm{NS}}$, this is true using from Definition 4.4 if if $\tau_{\mathrm{NS}} = \epsilon$, and from the protocol term definitions if $\tau_{\mathrm{NS}} = \mathrm{NS}_{\mathrm{ID}}(\_)$. The inductive case is straightforward. ■

4.10.3 A Full Set of Acceptance Condition Characterizations

We now design acceptance condition characterizations for all relevant action labels.

Lemma 4.11. *For every valid action trace $\tau = \_, \mathsf{ai}$ on $\mathcal{S}_{\mathsf{id}}$ and identity $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$:*

- ***(Equ2)** If $\mathsf{ai} = \mathrm{PU}_{\mathrm{ID}}(j,2)$. Let $\tau_2 = \_, \mathrm{PU}_{\mathrm{ID}}(j,1)$ such that $\tau_2 \prec \tau$. For every $\tau_1 = \_, \mathrm{PN}(j_1,1)$, let:*

$$\mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \equiv \begin{pmatrix} g(\phi^{\mathsf{in}}_{\tau}) = \mathsf{Mac}^2_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle \mathsf{n}^{j_1}\,,\, \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))\rangle) \\ \wedge\, g(\phi^{\mathsf{in}}_{\tau_2}) = \mathsf{n}^{j_1} \;\wedge\; \pi_1(g(\phi^{\mathsf{in}}_{\tau_1})) = \{\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^j_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} \end{pmatrix}$$

  Then:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \leftrightarrow \bigvee_{\substack{\tau_1 = \_, \mathrm{PN}(j_1,1) \\ \tau_2 \prec_\tau \tau_1}} \mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau}$$

- ***(Equ3)** If $\mathsf{ai} = \mathrm{PN}(j,1)$. Then:*

$$\begin{aligned} \mathsf{accept}^{\mathrm{ID}}_{\tau} &\leftrightarrow \bigvee_{\substack{\tau_1 = \_, \mathrm{PU}_{\mathrm{ID}}(j_1,1) \\ \tau_1 \prec \tau}} \begin{pmatrix} g(\phi^{\mathsf{in}}_{\tau_1}) = \mathsf{n}^j \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau})) = \{\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j_1}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} \\ \wedge\, \pi_2(g(\phi^{\mathsf{in}}_{\tau})) = \mathsf{Mac}^1_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle \{\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j_1}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}}\,,\, g(\phi^{\mathsf{in}}_{\tau_1})\rangle) \end{pmatrix} \\ &\leftrightarrow \bigvee_{\substack{\tau_1 = \_, \mathrm{PU}_{\mathrm{ID}}(j_1,1) \\ \tau_1 \prec \tau}} g(\phi^{\mathsf{in}}_{\tau_1}) = \mathsf{n}^j \wedge g(\phi^{\mathsf{in}}_{\tau}) = t_{\tau_1} \end{aligned}$$

- ***(Equ4)** If $\mathsf{ai} = \mathrm{TU}_{\mathrm{ID}}(j,1)$. For every $\tau_1 = \_, \mathrm{TN}(j_0,0)$ such that $\tau_1 \prec \tau$, we let:*

$$\mathsf{c\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau} \equiv \begin{pmatrix} \pi_3(g(\phi^{\mathsf{in}}_{\tau})) = \mathsf{Mac}^3_{\mathsf{k}_{\mathsf{m}}}(\langle \mathsf{n}^{j_0}\,,\, \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})\,,\, \sigma^{\mathsf{in}}_{\tau}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}})\rangle) \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{s\text{-}valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \\ \wedge\, \mathsf{range}(\sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}), \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})) \wedge g(\phi^{\mathsf{in}}_{\tau_1}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau})) = \mathsf{n}^{j_0} \\ \wedge\, \pi_2(g(\phi^{\mathsf{in}}_{\tau})) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \oplus f_k(\mathsf{n}^{j_0}) \wedge \sigma^{\mathsf{in}}_{\tau}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \end{pmatrix}$$

  Then:

$$\left(\mathsf{c\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau} \rightarrow \mathsf{accept}^{\mathrm{ID}}_{\tau_1}\right)_{\substack{\tau_1 = \_, \mathrm{TN}(j_0,0) \\ \tau_1 \prec \tau}} \qquad \mathsf{accept}^{\mathrm{ID}}_{\tau} \leftrightarrow \bigvee_{\substack{\tau_1 = \_, \mathrm{TN}(j_0,0) \\ \tau_1 \prec \tau}} \mathsf{c\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau}$$

- ***(Equ5)** If $\mathsf{ai} = \mathrm{TN}(j,1)$. Let $\tau_1 = \_, \mathrm{TN}(j,0)$ such that $\tau_1 \prec \tau$, and let $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$. Then:*

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \leftrightarrow \bigvee_{\substack{\tau_i = \_, \mathrm{TU}_{\mathrm{ID}}(j_i,1) \\ \tau_1 \prec_\tau \tau_i}} \mathsf{c\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_i} \wedge g(\phi^{\mathsf{in}}_{\tau}) = \mathsf{Mac}^4_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\mathsf{n}^j)$$

4.10.4 ⋆ (p. 114) Proof of Lemma 4.11

*Proof of **(Equ2)**.* Using **(Acc2)** we know that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \leftrightarrow \bigvee_{\substack{\tau_1=\_,\mathrm{PN}(j_1,1)\\ \tau_2\prec\tau\,\tau_1}} \mathsf{accept}^{\mathrm{ID}}_{\tau} \wedge g(\phi^{\mathsf{in}}_{\tau_2}) = \mathsf{n}^{j_1} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau_1})) = \{\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}}$$

$$\leftrightarrow \bigvee_{\substack{\tau_1=\_,\mathrm{PN}(j_1,1)\\ \tau_2\prec\tau\,\tau_1}} \begin{pmatrix} g(\phi^{\mathsf{in}}_{\tau}) = \mathsf{Mac}^{2}_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle \mathsf{n}^{j_1}\,,\, \sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle) \wedge g(\phi^{\mathsf{in}}_{\tau_2}) = \mathsf{n}^{j_1} \\ \wedge\, \pi_1(g(\phi^{\mathsf{in}}_{\tau_1})) = \{\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} \end{pmatrix}$$

Since $\sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) \equiv \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))$:

$$\leftrightarrow \bigvee_{\substack{\tau_1=\_,\mathrm{PN}(j_1,1)\\ \tau_2\prec\tau\,\tau_1}} \begin{pmatrix} g(\phi^{\mathsf{in}}_{\tau}) = \mathsf{Mac}^{2}_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle \mathsf{n}^{j_1}\,,\, \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))\rangle) \wedge g(\phi^{\mathsf{in}}_{\tau_2}) = \mathsf{n}^{j_1} \\ \wedge\, \pi_1(g(\phi^{\mathsf{in}}_{\tau_1})) = \{\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} \end{pmatrix}$$

$$\leftrightarrow \bigvee_{\substack{\tau_1=\_,\mathrm{PN}(j_1,1)\\ \tau_2\prec\tau\,\tau_1}} \mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau}$$

■

*Proof of **(Equ3)**.* Using **(Acc1)** it is easy to check that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \leftrightarrow \bigvee_{\tau_1=\_,\mathrm{PU_{ID}}(j_1,1)\prec\tau} \begin{pmatrix} g(\phi^{\mathsf{in}}_{\tau_1}) = \mathsf{n}^{j} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau})) = \{\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j_1}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} \\ \wedge\, \pi_2(g(\phi^{\mathsf{in}}_{\tau})) = \mathsf{Mac}^{1}_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle \pi_1(g(\phi^{\mathsf{in}}_{\tau}))\,,\, \mathsf{n}^{j}\rangle) \end{pmatrix}$$

Which can be rewritten as follows (we identify above, using waves and dots, which equalities are used, and which terms are rewritten):

$$\leftrightarrow \bigvee_{\tau_1=\_,\mathrm{PU_{ID}}(j_1,1)\prec\tau} \begin{pmatrix} g(\phi^{\mathsf{in}}_{\tau_1}) = \mathsf{n}^{j} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau})) = \{\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j_1}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} \\ \wedge\, \pi_2(g(\phi^{\mathsf{in}}_{\tau})) = \mathsf{Mac}^{1}_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle \{\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j_1}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}}\,,\, g(\phi^{\mathsf{in}}_{\tau_1})\rangle) \end{pmatrix}$$

First, observe that:

$$\{\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j_1}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} = \pi_1(t_{\tau_1}) \qquad \mathsf{Mac}^{1}_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle \{\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j_1}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}}\,,\, g(\phi^{\mathsf{in}}_{\tau_1})\rangle) = \pi_2(t_{\tau_1})$$

We conclude using the injectivity of the pair. ■

*Proof of **(Equ4)**.* Using **(Acc3)** we know that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \leftrightarrow \bigvee_{\substack{\tau_1=\_,\mathrm{TN}(j_0,0)\\ \tau_1\prec\tau}} \begin{pmatrix} \mathsf{accept}^{\mathrm{ID}}_{\tau} \wedge \mathsf{accept}^{\mathrm{ID}}_{\tau_1} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau})) = \mathsf{n}^{j_0} \\ \wedge\, \pi_2(g(\phi^{\mathsf{in}}_{\tau})) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \oplus \mathsf{f}_{\mathsf{k}}(\mathsf{n}^{j_0}) \wedge \sigma^{\mathsf{in}}_{\tau}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \end{pmatrix}$$

Inlining the definition of $\mathsf{accept}^{\mathrm{ID}}_{\tau_1}$:

$$\leftrightarrow \bigvee_{\substack{\tau_1=\_,\mathrm{TN}(j_0,0)\\ \tau_1\prec\tau}} \begin{pmatrix} \mathsf{accept}^{\mathrm{ID}}_{\tau} \wedge g(\phi^{\mathsf{in}}_{\tau_1}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \wedge \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \neq \mathsf{UnSet} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau})) = \mathsf{n}^{j_0} \\ \wedge\, \pi_2(g(\phi^{\mathsf{in}}_{\tau})) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \oplus \mathsf{f}_{\mathsf{k}}(\mathsf{n}^{j_0}) \wedge \sigma^{\mathsf{in}}_{\tau}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \end{pmatrix}$$

Inlining the definition of $\mathsf{accept}^{\mathrm{ID}}_{\tau}$:

$$\leftrightarrow \bigvee_{\substack{\tau_1=\_,\mathrm{TN}(j_0,0)\\ \tau_1\prec\tau}} \begin{pmatrix} \pi_3(g(\phi^{\mathsf{in}}_{\tau})) = \mathsf{Mac}^{3}_{\mathsf{k}_{\mathsf{m}}}(\langle \pi_1(g(\phi^{\mathsf{in}}_{\tau}))\,,\, \pi_2(g(\phi^{\mathsf{in}}_{\tau})) \oplus \mathsf{f}_{\mathsf{k}}(\pi_1(g(\phi^{\mathsf{in}}_{\tau})))\,,\, \sigma^{\mathsf{in}}_{\tau}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}})\rangle) \\ \wedge\, \sigma^{\mathsf{in}}_{\tau}(\mathsf{s\text{-}valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{range}(\sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}), \pi_2(g(\phi^{\mathsf{in}}_{\tau})) \oplus \mathsf{f}_{\mathsf{k}}(\pi_1(g(\phi^{\mathsf{in}}_{\tau})))) \\ g(\phi^{\mathsf{in}}_{\tau_1}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \wedge \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \neq \mathsf{UnSet} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau})) = \mathsf{n}^{j_0} \\ \wedge\, \pi_2(g(\phi^{\mathsf{in}}_{\tau})) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \oplus \mathsf{f}_{\mathsf{k}}(\mathsf{n}^{j_0}) \wedge \sigma^{\mathsf{in}}_{\tau}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \end{pmatrix}$$

We rewrite $\pi_1(g(\phi^{\mathsf{in}}_\tau))$ into n^{j_0}:

$$\leftrightarrow \bigvee_{\substack{\tau_1 = \_, \mathrm{TN}(j_0,0) \\ \tau_1 \prec \tau}} \begin{pmatrix} \pi_3(g(\phi^{\mathsf{in}}_\tau)) = \mathsf{Mac}^3_{\mathsf{k_m}}(\langle \underline{\mathsf{n}^{j_0}}, \underline{\pi_2(g(\phi^{\mathsf{in}}_\tau)) \oplus \mathsf{f_k}(\mathsf{n}^{j_0})}, \sigma^{\mathsf{in}}_\tau(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}})\rangle) \\ \wedge\, \sigma^{\mathsf{in}}_\tau(\mathsf{s\text{-}valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{range}(\sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}), \underline{\pi_2(g(\phi^{\mathsf{in}}_\tau)) \oplus \mathsf{f_k}(\mathsf{n}^{j_0})}) \\ g(\phi^{\mathsf{in}}_{\tau_1}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \wedge \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \neq \mathsf{UnSet} \wedge \underline{\pi_1(g(\phi^{\mathsf{in}}_\tau)) = \mathsf{n}^{j_0}} \\ \wedge\, \underline{\pi_2(g(\phi^{\mathsf{in}}_\tau)) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \oplus \mathsf{f_k}(\mathsf{n}^{j_0})} \wedge \sigma^{\mathsf{in}}_\tau(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \end{pmatrix}$$

We rewrite $\pi_2(g(\phi^{\mathsf{in}}_\tau)) \oplus \mathsf{f_k}(\mathsf{n}^{j_0})$ into $\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$:

$$\leftrightarrow \bigvee_{\substack{\tau_1 = \_, \mathrm{TN}(j_0,0) \\ \tau_1 \prec \tau}} \begin{pmatrix} \pi_3(g(\phi^{\mathsf{in}}_\tau)) = \mathsf{Mac}^3_{\mathsf{k_m}}(\langle \mathsf{n}^{j_0}, \underline{\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})}, \sigma^{\mathsf{in}}_\tau(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}})\rangle) \\ \wedge\, \sigma^{\mathsf{in}}_\tau(\mathsf{s\text{-}valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{range}(\sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}), \underline{\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})}) \\ \wedge\, g(\phi^{\mathsf{in}}_{\tau_1}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \wedge \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \neq \mathsf{UnSet} \wedge \pi_1(g(\phi^{\mathsf{in}}_\tau)) = \mathsf{n}^{j_0} \\ \wedge\, \underline{\pi_2(g(\phi^{\mathsf{in}}_\tau)) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \oplus \mathsf{f_k}(\mathsf{n}^{j_0})} \wedge \sigma^{\mathsf{in}}_\tau(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \end{pmatrix} \tag{4.13}$$

Let $\tau_2 = \_, \mathrm{TU}_{\mathrm{ID}}(j_0, 0) \prec \tau$. By validity of τ, there are no user ID actions between τ_2 and τ, and therefore it is easy to check that $\sigma^{\mathsf{in}}_\tau(\mathsf{s\text{-}valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \rightarrow \sigma^{\mathsf{in}}_{\tau_2}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}})$, and that $\sigma^{\mathsf{in}}_\tau(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}})$. Moreover, using **(B3)** we know that $\sigma^{\mathsf{in}}_{\tau_2}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \rightarrow \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) \neq \mathsf{UnSet}$. Therefore $\sigma^{\mathsf{in}}_\tau(\mathsf{s\text{-}valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \rightarrow \sigma^{\mathsf{in}}_\tau(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) \neq \mathsf{UnSet}$. It follows that:

$$\left(\sigma^{\mathsf{in}}_\tau(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \wedge \sigma^{\mathsf{in}}_\tau(\mathsf{s\text{-}valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}})\right) \rightarrow \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \neq \mathsf{UnSet}$$

Hence we can simplify (4.13) by removing $\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \neq \mathsf{UnSet}$. This yields:

$$\begin{aligned} \mathsf{accept}^{\mathrm{ID}}_\tau &\leftrightarrow \bigvee_{\substack{\tau_1 = \_, \mathrm{TN}(j_0,0) \\ \tau_1 \prec \tau}} \begin{pmatrix} \pi_3(g(\phi^{\mathsf{in}}_\tau)) = \mathsf{Mac}^3_{\mathsf{k_m}}(\langle \mathsf{n}^{j_0}, \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}), \sigma^{\mathsf{in}}_\tau(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}})\rangle) \wedge \sigma^{\mathsf{in}}_\tau(\mathsf{s\text{-}valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \\ \wedge\, \mathsf{range}(\sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}), \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})) \wedge g(\phi^{\mathsf{in}}_{\tau_1}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \wedge \pi_1(g(\phi^{\mathsf{in}}_\tau)) = \mathsf{n}^{j_0} \\ \wedge\, \pi_2(g(\phi^{\mathsf{in}}_\tau)) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \oplus \mathsf{f_k}(\mathsf{n}^{j_0}) \wedge \sigma^{\mathsf{in}}_\tau(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \end{pmatrix} \\ &\leftrightarrow \bigvee_{\substack{\tau_1 = \_, \mathrm{TN}(j_0,0) \\ \tau_1 \prec \tau}} \mathsf{c\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau} \end{aligned}$$

Finally, we check that for every $\tau_1 = \_, \mathrm{TN}(j_0, 0)$ such that $\tau_1 \prec \tau$, we have $\mathsf{c\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau} \rightarrow \mathsf{accept}^{\mathrm{ID}}_{\tau_1}$. ■

*Proof of **(Equ5)**.* Using **(Acc4)** we know that:

$$\mathsf{accept}^{\mathrm{ID}}_\tau \leftrightarrow \bigvee_{\tau_i = \_, \mathrm{TU}_{\mathrm{ID}}(j_i,1) \prec \tau} \mathsf{accept}^{\mathrm{ID}}_\tau \wedge \mathsf{accept}_{\tau_i} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau_i})) = \mathsf{n}^j$$

Moreover, using **(Equ4)** we know that:

$$\mathsf{accept}^{\mathrm{ID}}_\tau \leftrightarrow \bigvee_{\substack{\tau_i = \_, \mathrm{TU}_{\mathrm{ID}}(j_i,1) \prec \tau \\ \tau_2 = \_, \mathrm{TN}(j_2,0) \prec \tau_i}} \mathsf{accept}^{\mathrm{ID}}_\tau \wedge \mathsf{c\text{-}tr}^{\mathsf{n}:\tau_2}_{\mathsf{u}:\tau_i} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau_i})) = \mathsf{n}^j$$

Let $\tau_2 = \_, \mathrm{TN}(j_2, 0) \prec \tau_i$. Then we know that $\mathsf{c\text{-}tr}^{\mathsf{n}:\tau_2}_{\mathsf{u}:\tau_i} \rightarrow \pi_1(g(\phi^{\mathsf{in}}_{\tau_i})) = \mathsf{n}^{j_2}$. Therefore using $=$-ind we know that if $j_2 \neq j$:

$$\left(\mathsf{c\text{-}tr}^{\mathsf{n}:\tau_2}_{\mathsf{u}:\tau_i} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau_i})) = \mathsf{n}^j\right) \rightarrow \left(\pi_1(g(\phi^{\mathsf{in}}_{\tau_i})) = \mathsf{n}^{j_2} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau_i})) = \mathsf{n}^j\right) \rightarrow \mathsf{false}$$

Hence:

$$\begin{aligned} \mathsf{accept}^{\mathrm{ID}}_\tau &\leftrightarrow \bigvee_{\substack{\tau_i = \_, \mathrm{TU}_{\mathrm{ID}}(j_i,1) \\ \tau_1 \prec \tau \tau_i}} \mathsf{accept}^{\mathrm{ID}}_\tau \wedge \mathsf{c\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_i} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau_i})) = \mathsf{n}^j \\ &\leftrightarrow \bigvee_{\substack{\tau_i = \_, \mathrm{TU}_{\mathrm{ID}}(j_i,1) \\ \tau_1 \prec \tau \tau_i}} \mathsf{accept}^{\mathrm{ID}}_\tau \wedge \mathsf{c\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_i} \qquad \text{(Since } \mathsf{c\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_i} \rightarrow \pi_1(g(\phi^{\mathsf{in}}_{\tau_i})) = \mathsf{n}^j\text{)} \end{aligned}$$

We inline the definition of $\mathsf{accept}^{\mathrm{ID}}_{\tau}$:

$$\leftrightarrow \bigvee_{\substack{\tau_i = \_, \mathrm{TU}_{\mathrm{ID}}(j_i,1) \\ \tau_1 \prec_\tau \tau_i}} g(\phi^{\mathsf{in}}_{\tau}) = \mathsf{Mac}^4_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\mathsf{n}^j) \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{b\text{-}auth}^j_{\mathrm{N}}) = \mathrm{ID} \wedge \mathsf{c\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_i}$$

Using **(Equ4)**, we know that for every $\tau_1 = \_, \mathrm{TN}(j_0, 0)$ such that $\tau_1 \prec \tau$, $\mathsf{c\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau} \rightarrow \mathsf{accept}^{\mathrm{ID}}_{\tau_1}$. Moreover, using **(A6)** we know that $\mathsf{accept}^{\mathrm{ID}}_{\tau_1} \rightarrow \sigma^{\mathsf{in}}_{\tau_1}(\mathsf{b\text{-}auth}^j_{\mathrm{N}}) = \mathrm{ID}$. Besides, $\sigma^{\mathsf{in}}_{\tau_1}(\mathsf{b\text{-}auth}^j_{\mathrm{N}}) = \mathrm{ID} \rightarrow \sigma^{\mathsf{in}}_{\tau}(\mathsf{b\text{-}auth}^j_{\mathrm{N}}) = \mathrm{ID}$. Hence $\mathsf{c\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau} \rightarrow \sigma^{\mathsf{in}}_{\tau}(\mathsf{b\text{-}auth}^j_{\mathrm{N}}) = \mathrm{ID}$. By consequence:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \leftrightarrow \bigvee_{\substack{\tau_i = \_, \mathrm{TU}_{\mathrm{ID}}(j_i,1) \\ \tau_1 \prec_\tau \tau_i}} g(\phi^{\mathsf{in}}_{\tau}) = \mathsf{Mac}^4_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\mathsf{n}^j) \wedge \mathsf{c\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_i} \qquad \blacksquare$$

4.10.5 $\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}$ Concealment

Lemma 4.12. *Let τ be a valid action trace on $\mathcal{S}_{id}$ and $\mathrm{ID}_x \in \mathcal{S}_{id}$. For every $\tau_a = \_, \mathrm{TN}(j_a, 1)$ or $\tau_a = \_, \mathrm{PN}(j_a, 1)$ such that $\tau_a \preceq \tau$, and for every $\tau_b = \mathrm{PU}_{\mathrm{ID}_x}(j_i, 1)$ or $\tau_b = \mathrm{TU}_{\mathrm{ID}_x}(j_i, 1)$ such that $\tau_b \prec \tau_a$, if:*

$$\{\tau_1 \mid \tau_b \prec_\tau \tau_1\} \cap \{\mathrm{PU}_{\mathrm{ID}_x}(j, \_), \mathrm{TU}_{\mathrm{ID}_x}(j, \_), \mathrm{FU}_{\mathrm{ID}_x}(j) \mid j \in \mathbb{N}\} \subseteq \{\mathrm{PU}_{\mathrm{ID}_x}(j_i, 2), \mathrm{FU}_{\mathrm{ID}_x}(j_i)\}$$

Then there exists a derivation of:

$$\mathsf{inc\text{-}accept}^{\mathrm{ID}_x}_{\tau_a} \wedge \sigma_{\tau_b}(\mathit{b\text{-}auth}^{\mathrm{ID}_x}_{\mathrm{U}}) = \mathit{n}^{j_a} \wedge \mathsf{accept}^{\mathrm{ID}_x}_{\tau_b} \rightarrow g(\phi^{\mathit{in}}_{\tau}) \not\equiv \mathrm{GUTI}^{j_a}$$

Proof ★(p. 118). Let β_τ be the term:

$$\beta_\tau \equiv \mathsf{inc\text{-}accept}^{\mathrm{ID}_x}_{\tau_a} \wedge \sigma_{\tau_b}(\mathsf{b\text{-}auth}^{\mathrm{ID}_x}_{\mathrm{U}}) = \mathsf{n}^{j_a} \wedge \mathsf{accept}^{\mathrm{ID}_x}_{\tau_b}$$

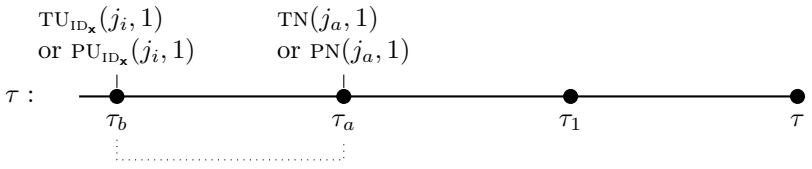
For every $\tau_a \preceq \tau_x \preceq \tau$, we let $\mathsf{leak}^{\mathsf{in}}_{\tau_x}$ be the vector containing the terms (in an arbitrary but fixed order):

- $\mathsf{leak}^{\mathsf{in}}_{\tau_0}$ if $\tau_x = \tau_0, \mathsf{ai}_0$ and $\tau_a \prec \tau_x$.
- The term β_τ.
- All the keys except $\mathsf{k}^{\mathrm{ID}_x}$, $\mathsf{k}^{\mathrm{ID}_x}_{\mathsf{m}}$ and the asymmetric secret key sk_{N}.
- All elements of $\sigma^{\mathsf{in}}_{\tau_x}$, except:
  - All the user ID_x values, i.e. for every x, $\sigma^{\mathsf{in}}_{\tau_x}(\mathrm{X}^{\mathrm{ID}_x}_{\mathrm{U}}) \notin \mathsf{leak}^{\mathsf{in}}_{\tau_x}$.
  - The network's GUTI value of user ID_x, i.e. $\sigma^{\mathsf{in}}_{\tau_x}(\mathrm{GUTI}^{\mathrm{ID}_x}_{\mathrm{N}}) \notin \mathsf{leak}^{\mathsf{in}}_{\tau_x}$.
- For every $\tau_a \preceq \tau_n \preceq \tau$ such that $\tau_n = \_, \mathrm{FN}(j)$, the term $\mathsf{Mac}^4_{\mathsf{k}^{\mathrm{ID}_x}_{\mathsf{m}}}(\mathsf{n}^j)$.
- For every $\tau_a \preceq \tau_n \preceq \tau$ such that $\tau_n = \_, \mathrm{PN}(j, 1)$ then $\mathsf{Mac}^4_{\mathsf{k}^{\mathrm{ID}_x}_{\mathsf{m}}}(\mathsf{n}^j)$, for every $\tau_2 = \_, \mathrm{PU}_{\mathrm{ID}_x}(j_2, 1) \preceq \tau_b$, the term $\mathsf{Mac}^2_{\mathsf{k}^{\mathrm{ID}_x}_{\mathsf{m}}}(\langle \mathsf{n}^j, \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))\rangle)$.

Let GUTI be a fresh name. We show by induction on τ_1 in $\tau_a \preceq \tau_1 \prec \tau$ that there are derivations of:

$$[\beta_\tau](\phi_{\tau_1}, \mathsf{leak}_{\tau_1}, \mathrm{GUTI}^{j_a}) \sim [\beta_\tau](\phi_{\tau_1}, \mathsf{leak}_{\tau_1}, \mathrm{GUTI}) \qquad \text{and} \qquad \beta_\tau \rightarrow \sigma_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}_x}_{\mathrm{N}}) = \mathrm{GUTI}^{j_a}$$

We depict the situation below:



Case $\tau_1 = \tau_a$ First, $\beta_\tau \rightarrow \mathsf{inc\text{-}accept}^{\mathrm{ID}_x}_{\tau_a}$, and $\mathsf{inc\text{-}accept}^{\mathrm{ID}_x}_{\tau_a} \rightarrow \sigma_{\tau_a}(\mathrm{GUTI}^{\mathrm{ID}_x}_{\mathrm{N}}) = \mathrm{GUTI}^{j_a}$. Therefore:

$$\beta_\tau \rightarrow \sigma_{\tau_a}(\mathrm{GUTI}^{\mathrm{ID}_x}_{\mathrm{N}}) = \mathrm{GUTI}^{j_a}$$

Then, we observe from the definition of leak_{τ_a} that $\mathrm{GUTI}^{j_a} \notin \mathsf{st}(\mathsf{leak}_{\tau_a})$ (since $\sigma_{\tau_a}(\mathrm{GUTI}^{\mathrm{ID}_x}_{\mathrm{N}})$ is *not* in leak_{τ_a}). Moreover GUTI^{j_a} does not appear in $\phi^{\mathsf{in}}_{\tau_a}$ and t_{τ_a}. Besides, GUTI is a fresh name. By consequence we can apply the Fresh axiom, and then conclude using Refl:

$$\frac{\dfrac{}{[\beta_\tau](\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}) \sim [\beta_\tau](\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1})}\,\mathsf{Refl}}{[\beta_\tau](\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI}^{j_a}) \sim [\beta_\tau](\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI})}\,\mathsf{Fresh}$$

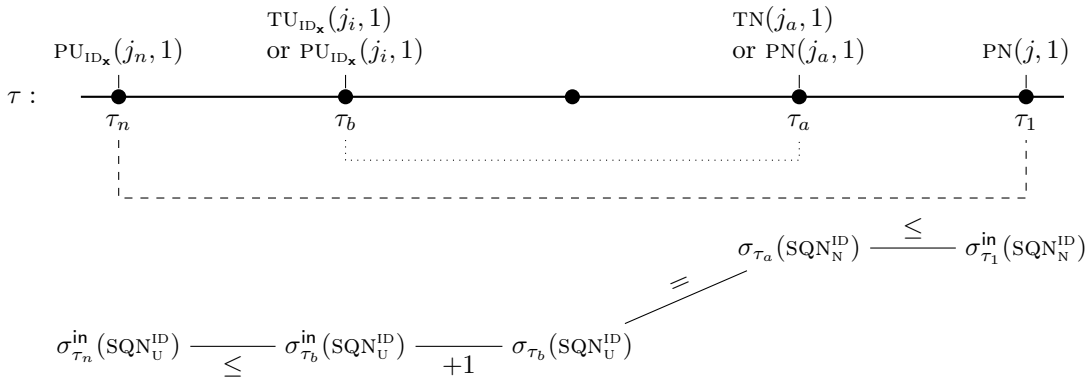


Figure 4.20: Graphical Representation Used in the Proof of Lemma 4.12

Case $\tau_a \prec \tau_1$ Let ai be such that $\tau_1 = \_, \mathsf{ai}$. Assume by induction that we have derivations of:

$$\big[\beta_\tau\big]\big(\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI}^{j_a}\big) \sim \big[\beta_\tau\big]\big(\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI}\big) \tag{4.14}$$

$$\beta_\tau \to \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}_x}_{\mathrm{N}}) = \mathrm{GUTI}^{j_a} \tag{4.15}$$

Part 1 First, we show that:

$$\beta_\tau \to \sigma_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}_x}_{\mathrm{N}}) = \mathrm{GUTI}^{j_a}$$

Since we know that (4.15) holds, we just need to look at the ai that update $\mathrm{GUTI}^{\mathrm{ID}_x}_{\mathrm{N}}$ to conclude:

- If $\mathsf{ai} = \mathrm{TN}(j,0)$. Using (4.14), we know that $\big[\beta_\tau\big] g(\phi^{\mathsf{in}}_{\tau_1}) \neq \mathrm{GUTI}^{j_a}$. Hence using (4.15):

  $$\beta_\tau \to \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}_x}_{\mathrm{N}}) \neq g(\phi^{\mathsf{in}}_{\tau_1})$$

  Which shows that $\beta_\tau \to \neg\mathsf{accept}^{\mathrm{ID}_x}_{\tau_1}$. This concludes this case.
- If $\mathsf{ai} = \mathrm{PN}(j,1)$. Since $\tau_a = \mathrm{TN}(j_a,1)$ or $\mathrm{PN}(j_a,1)$, we know by validity of τ that $j_a \neq j$. We give a graphical summary of this proof in Figure 4.20. Using **(Equ3)** we know that:

  $$\mathsf{accept}^{\mathrm{ID}_x}_{\tau_1} \to \bigvee_{\substack{\tau_n = \_, \mathrm{PU}_{\mathrm{ID}}(j_n,1) \\ \tau_n \prec \tau_1}} g(\phi^{\mathsf{in}}_{\tau_n}) = \mathsf{n}^j \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau_1})) = \{\langle \mathrm{ID}_x\,,\, \sigma^{\mathsf{in}}_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}_x}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j_n}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} \tag{4.16}$$

  Since $j_a \neq j$ we know that $\mathsf{n}^j \neq \mathsf{n}^{j_a}$. Moreover:

  $$\sigma_{\tau_b}(\mathsf{b\text{-}auth}^{\mathrm{ID}_x}_{\mathrm{U}}) = \mathsf{n}^{j_a} \wedge \mathsf{accept}^{\mathrm{ID}_x}_{\tau_b} \to g(\phi^{\mathsf{in}}_{\tau_b}) = \mathsf{n}^{j_a}$$

  Hence $\beta_\tau \to g(\phi^{\mathsf{in}}_{\tau_b}) \neq \mathsf{n}^j$. Moreover, for every τ' between τ_b and τ_1, we know that $\tau' \neq \mathrm{PU}_{\mathrm{ID}_x}(\_,1)$. Therefore we know that:

  $$\beta_\tau \wedge \mathsf{accept}^{\mathrm{ID}_x}_{\tau_1} \to \bigvee_{\substack{\tau_n = \_, \mathrm{PU}_{\mathrm{ID}}(j_n,1) \\ \tau_n \prec \tau_b}} g(\phi^{\mathsf{in}}_{\tau_n}) = \mathsf{n}^j \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau_1})) = \{\langle \mathrm{ID}_x\,,\, \sigma^{\mathsf{in}}_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}_x}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j_n}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}}$$

  Let $\tau_n = \_, \mathrm{PU}_{\mathrm{ID}}(j_n,1)$ such that $\tau_n \prec \tau_b$. We know that:

  $$\beta_\tau \to \sigma_{\tau_a}(\mathrm{SQN}^{\mathrm{ID}_x}_{\mathrm{N}}) = \sigma_{\tau_b}(\mathrm{SQN}^{\mathrm{ID}_x}_{\mathrm{U}}) = \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_b}(\mathrm{SQN}^{\mathrm{ID}_x}_{\mathrm{U}}))$$

  Since $\sigma_{\tau_a}(\mathrm{SQN}^{\mathrm{ID}_x}_{\mathrm{N}}) \leq \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}_x}_{\mathrm{N}})$ and $\sigma^{\mathsf{in}}_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}_x}_{\mathrm{U}}) \leq \sigma^{\mathsf{in}}_{\tau_b}(\mathrm{SQN}^{\mathrm{ID}_x}_{\mathrm{U}})$, we deduce that:

  $$\beta_\tau \wedge \mathsf{accept}^{\mathrm{ID}_x}_{\tau_1} \wedge g(\phi^{\mathsf{in}}_{\tau_n}) = \mathsf{n}^j \to \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}_x}_{\mathrm{N}}) > \sigma^{\mathsf{in}}_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}_x}_{\mathrm{U}})$$

  Moreover:

  $$\beta_\tau \wedge \mathsf{inc\text{-}accept}^{\mathrm{ID}_x}_{\tau_1} \wedge g(\phi^{\mathsf{in}}_{\tau_n}) = \mathsf{n}^j \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau_1})) = \{\langle \mathrm{ID}_x\,,\, \sigma^{\mathsf{in}}_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}_x}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j_n}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} \to \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}_x}_{\mathrm{N}}) \leq \sigma^{\mathsf{in}}_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}_x}_{\mathrm{U}})$$

Hence:

$$\beta_\tau \wedge \mathsf{accept}^{\mathrm{ID}_\mathsf{x}}_{\tau_1} \wedge g(\phi^{\mathsf{in}}_{\tau_n}) = \mathsf{n}^j \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau_1})) = \{\langle \mathrm{ID}_\mathsf{x}\,,\, \sigma^{\mathsf{in}}_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}_\mathsf{x}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j_n}_\mathsf{e}}_{\mathsf{pk}_{\mathrm{N}}} \rightarrow \neg\mathsf{inc\text{-}accept}^{\mathrm{ID}_\mathsf{x}}_{\tau_1}$$

Using (4.16), this shows that:

$$\beta_\tau \wedge \mathsf{accept}^{\mathrm{ID}_\mathsf{x}}_{\tau_1} \rightarrow \neg\mathsf{inc\text{-}accept}^{\mathrm{ID}_\mathsf{x}}_{\tau_1} \tag{4.17}$$

This concludes this case.

- If $\mathsf{ai} = \mathrm{TN}(j,1)$. Since $\tau_a = \mathrm{TN}(j_a,1)$ or $\mathrm{PN}(j_a,1)$, we know by validity of τ that $j_a \neq j$. From the induction hypothesis we know that $\beta_\tau \rightarrow \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}_\mathsf{x}}_{\mathrm{N}}) = \mathrm{GUTI}^{j_a}$. It is easy to check that:

$$\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}_\mathsf{x}}_{\mathrm{N}}) = \mathrm{GUTI}^{j_a} \rightarrow \sigma^{\mathsf{in}}_{\tau_1}(\mathsf{session}^{\mathrm{ID}_\mathsf{x}}_{\mathrm{N}}) = \mathsf{n}^{j_a}$$

Hence, since $j \neq j_a$:

$$\beta_\tau \;\rightarrow\; \sigma^{\mathsf{in}}_{\tau_1}(\mathsf{session}^{\mathrm{ID}_\mathsf{x}}_{\mathrm{N}}) = \mathsf{n}^{j_a} \;\rightarrow\; \sigma^{\mathsf{in}}_{\tau_1}(\mathsf{session}^{\mathrm{ID}_\mathsf{x}}_{\mathrm{N}}) \neq \mathsf{n}^{j} \rightarrow \neg\mathsf{inc\text{-}accept}^{\mathrm{ID}_\mathsf{x}}_{\tau_1} \;\rightarrow\; \sigma_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}_\mathsf{x}}_{\mathrm{N}}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}_\mathsf{x}}_{\mathrm{N}}) = \mathrm{GUTI}^{j_a}$$

Which concludes this case.

Part 2 We now show that:

$$\big[\beta_\tau\big]\big(\phi_{\tau_1}, \mathsf{leak}_{\tau_1}, \mathrm{GUTI}^{j_a}\big) \;\sim\; \big[\beta_\tau\big]\big(\phi_{\tau_1}, \mathsf{leak}_{\tau_1}, \mathrm{GUTI}\big)$$

We do a case disjunction on ai. We only details the case where ai is a symbolic action of user ID, with $\mathrm{ID} \neq \mathrm{ID}_\mathsf{x}$, and the case where $\mathsf{ai} = \mathrm{FN}(j_a)$. All the other cases are similar to these two cases, and their proof will only be sketched.

- If ai is a symbolic action of user ID, with $\mathrm{ID} \neq \mathrm{ID}_\mathsf{x}$, then for every $u \in \mathsf{leak}_{\tau_1} \backslash \mathsf{leak}^{\mathsf{in}}_{\tau_1}$ (resp. $u \equiv t_{\tau_1}$) we show that there exists a many-hole context C_u such that $u \equiv C_u[\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}]$ and C_u does not contain any nonce in $\mathcal{N}$.

  We only detail the case $\mathsf{ai} = \mathrm{FU}_{\mathrm{ID}}(j)$. First, observe that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau_1} \;\equiv\; \left(\begin{array}{ll} & \mathsf{eq}(\pi_2(g(\underline{\phi^{\mathsf{in}}_{\tau_1}})), \mathsf{Mac}^5_{\underline{\mathsf{k_m}}}(\langle \pi_1(g(\underline{\phi^{\mathsf{in}}_{\tau_1}})) \oplus \mathsf{f}^{\mathsf{r}}_{\underline{\mathsf{k}}}(\underline{\sigma^{\mathsf{in}}_{\tau_1}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}})}),\, \underline{\sigma^{\mathsf{in}}_{\tau_1}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}})}\rangle)) \\ \wedge & \neg\mathsf{eq}(\underline{\sigma^{\mathsf{in}}_{\tau_1}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}})}, \mathsf{fail}) \end{array}\right)$$

  All the underlined subterms are in $\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}$, therefore there exists C_{accept} such that $\mathsf{accept}^{\mathrm{ID}}_{\tau_1} \equiv C_{\mathsf{accept}}[\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}]$. Remark that $\mathsf{leak}_{\tau_1} \backslash \mathsf{leak}^{\mathsf{in}}_{\tau_1} = \{\sigma^{\mathsf{in}}_{\tau_1}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}), \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}})\}$. Moreover:

$$t_{\tau_1} \;\equiv\; \mathsf{if}\ \mathsf{accept}^{\mathrm{ID}}_{\tau_1}\ \mathsf{then\ ok\ else\ error} \qquad\qquad \sigma^{\mathsf{in}}_{\tau_1}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \;\equiv\; \mathsf{accept}^{\mathrm{ID}}_{\tau_1}$$

$$\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) \;\equiv\; \mathsf{if}\ \mathsf{accept}^{\mathrm{ID}}_{\tau_1}\ \mathsf{then}\ \pi_1(g(\underline{\phi^{\mathsf{in}}_{\tau_1}})) \oplus \mathsf{f}^{\mathsf{r}}_{\underline{\mathsf{k}}}(\underline{\sigma^{\mathsf{in}}_{\tau_1}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}})})\ \mathsf{else\ UnSet}$$

  Using the fact that all the underlined subterms are in $\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}$, and using C_{accept} it is easy to build the wanted contexts.

  We then conclude using the FA rule under context, the Dup rule and the induction hypothesis:

$$\dfrac{\dfrac{\big[\beta_\tau\big]\big(\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI}^{j_a}\big) \;\sim\; \big[\beta_\tau\big]\big(\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI}\big)}{\big[\beta_\tau\big]\big(\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI}^{j_a}, (C_u[\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}])_{u \in \{t_{\tau_1}, \mathsf{leak}_{\tau_1} \backslash \mathsf{leak}^{\mathsf{in}}_{\tau_1}\}}\big)}\ (\mathsf{FA_c} + \mathsf{Dup})^*}{\big[\beta_\tau\big]\big(\phi_{\tau_1}, \mathsf{leak}_{\tau_1}, \mathrm{GUTI}^{j_a}\big) \;\sim\; \big[\beta_\tau\big]\big(\phi_{\tau_1}, \mathsf{leak}_{\tau_1}, \mathrm{GUTI}\big)}\ R$$

- If $\mathsf{ai} = \mathrm{FN}(j_a)$. It is easy to check that:

$$\sigma^{\mathsf{in}}_{\tau_a}(\mathsf{e\text{-}auth}^{j_a}_{\mathrm{N}}) \neq \mathrm{ID}_\mathsf{x} \rightarrow \sigma^{\mathsf{in}}_{\tau_a}(\mathrm{GUTI}^{\mathrm{ID}_\mathsf{x}}_{\mathrm{N}}) \neq \mathrm{GUTI}^{j_a} \rightarrow \sigma^{\mathsf{in}}_{\tau}(\mathrm{GUTI}^{\mathrm{ID}_\mathsf{x}}_{\mathrm{N}}) \neq \mathrm{GUTI}^{j_a}$$

Therefore using the induction property (4.15) we deduce that $\beta_\tau \to \sigma^{\mathsf{in}}_{\tau_a}(\mathsf{e\text{-}auth}^{j_a}_{\mathrm{N}}) = \mathrm{ID}_{\mathsf{x}}$. Moreover by validity of τ, there are no session j_a network actions between τ_a and τ_1. It follows that $\sigma^{\mathsf{in}}_{\tau_a}(\mathsf{e\text{-}auth}^{j_a}_{\mathrm{N}}) = \mathrm{ID}_{\mathsf{x}} \to \sigma^{\mathsf{in}}_{\tau_1}(\mathsf{e\text{-}auth}^{j_a}_{\mathrm{N}}) = \mathrm{ID}_{\mathsf{x}}$. Hence:

$$[\beta_\tau]t_{\tau_1} = [\beta_\tau]\langle \mathrm{GUTI}^{j_a} \oplus \mathsf{f}^{\mathsf{r}}_{\mathsf{k}^{\mathrm{ID}_{\mathsf{x}}}}(\mathsf{n}^{j_a})\,,\, \mathsf{Mac}^5_{\mathsf{k}^{\mathrm{ID}_{\mathsf{x}}}_{\mathsf{m}}}(\langle \mathrm{GUTI}^{j_a}\,,\, \mathsf{n}^{j_a}\rangle)\rangle$$

Observe that:

$$\big[\beta_\tau\big]\big(\phi_{\tau_1}, \mathsf{leak}_{\tau_1}, \mathrm{GUTI}^{j_a}\big) = \big[\beta_\tau\big]\big(\phi^{\mathsf{in}}_{\tau_1}, \langle \mathrm{GUTI}^{j_a} \oplus \mathsf{f}^{\mathsf{r}}_{\mathsf{k}^{\mathrm{ID}_{\mathsf{x}}}}(\mathsf{n}^{j_a})\,,\, \mathsf{Mac}^5_{\mathsf{k}^{\mathrm{ID}_{\mathsf{x}}}_{\mathsf{m}}}(\langle \mathrm{GUTI}^{j_a}\,,\, \mathsf{n}^{j_a}\rangle)\rangle, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI}^{j_a}\big)$$

We are now going to apply the PRF-f axiom on the left to replace $\mathrm{GUTI}^{j_a} \oplus \mathsf{f}^{\mathsf{r}}_{\mathsf{k}^{\mathrm{ID}_{\mathsf{x}}}}(\mathsf{n}^{j_a})$ with $\mathrm{GUTI}^{j_a} \oplus \mathsf{n}_{\mathsf{f}}$ where n_{f} is a fresh nonce. For every $\tau_2 = \_, \mathrm{FU}_{\mathrm{ID}}(\_) \prec \tau_1$, we use **(Equ1)** to replace every occurrences of accept_{τ_2} in $\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \beta_\tau$ with:

$$\gamma_{\tau_2} \equiv \bigvee_{\substack{\tau_3 = \_, \mathrm{FN}(\_) \prec \tau_2 \\ \tau_3 \not\prec \tau_2\, {}^{\mathrm{NS}_{\mathrm{ID}}(\_)}}} \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_3}_{\mathsf{u}:\tau_2}$$

which yields the terms $\phi'^{\mathsf{in}}_{\tau_1}, \mathsf{leak}'^{\mathsf{in}}_{\tau_1}, \beta'_\tau$. We can check that:

$$\mathsf{set\text{-}prf}^{\mathsf{f}^{\mathsf{r}}_{\mathsf{k}^{\mathrm{ID}_{\mathsf{x}}}}}(\gamma_{\tau_2}) \subseteq \{\mathsf{n}^p \mid \exists \tau' = \_, \mathrm{FN}(p) \prec \tau_1\}$$

And that:

$$\mathsf{set\text{-}prf}^{\mathsf{f}^{\mathsf{r}}_{\mathsf{k}^{\mathrm{ID}_{\mathsf{x}}}}}(\phi'^{\mathsf{in}}_{\tau_1}, \mathsf{leak}'^{\mathsf{in}}_{\tau_1}) = \{\mathsf{n}^p \mid \exists \tau' = \_, \mathrm{FN}(p) \prec \tau_1\}$$

Therefore we can apply the PRF-f axiom as wanted: first we replace $\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \beta_\tau$ by $\phi'^{\mathsf{in}}_{\tau_1}, \mathsf{leak}'^{\mathsf{in}}_{\tau_1}, \beta'_\tau$ using rule R; then we apply the PRF-f axiom; and finally we rewrite all γ_{τ_2} back into $\mathsf{accept}^{\mathrm{ID}_{\mathsf{x}}}_{\tau_2}$. Finally, we use the $\oplus$-indep axiom to replace $\mathrm{GUTI}^{j_a} \oplus \mathsf{n}_{\mathsf{f}}$ with a fresh nonce n'_{f}. This yields:

$$\dfrac{\dfrac{\dfrac{\dfrac{\big[\beta_\tau\big]\big(\phi^{\mathsf{in}}_{\tau_1}, \langle \mathsf{n}'_{\mathsf{f}}\,,\, \mathsf{Mac}^5_{\mathsf{k}^{\mathrm{ID}_{\mathsf{x}}}_{\mathsf{m}}}(\langle \mathrm{GUTI}^{j_a}\,,\, \mathsf{n}^{j_a}\rangle)\rangle, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI}^{j_a}\big) \sim \big[\beta_\tau\big]\big(\phi_{\tau_1}, \mathsf{leak}_{\tau_1}, \mathrm{GUTI}\big)}{\big[\beta_\tau\big]\big(\phi^{\mathsf{in}}_{\tau_1}, \langle \mathrm{GUTI}^{j_a} \oplus \mathsf{n}_{\mathsf{f}}\,,\, \mathsf{Mac}^5_{\mathsf{k}^{\mathrm{ID}_{\mathsf{x}}}_{\mathsf{m}}}(\langle \mathrm{GUTI}^{j_a}\,,\, \mathsf{n}^{j_a}\rangle)\rangle, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI}^{j_a}\big) \sim \big[\beta_\tau\big]\big(\phi_{\tau_1}, \mathsf{leak}_{\tau_1}, \mathrm{GUTI}\big)}\,\oplus\text{-indep}}{\big[\beta_\tau\big]\big(\phi^{\mathsf{in}}_{\tau_1}, \langle \mathrm{GUTI}^{j_a} \oplus \mathsf{n}_{\mathsf{f}}\,,\, \mathsf{Mac}^5_{\mathsf{k}^{\mathrm{ID}_{\mathsf{x}}}_{\mathsf{m}}}(\langle \mathrm{GUTI}^{j_a}\,,\, \mathsf{n}^{j_a}\rangle)\rangle, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI}^{j_a}\big) \sim \big[\beta_\tau\big]\big(\phi_{\tau_1}, \mathsf{leak}_{\tau_1}, \mathrm{GUTI}\big)}\,R}{\big[\beta'_\tau\big]\big(\phi'^{\mathsf{in}}_{\tau_1}, \langle \mathrm{GUTI}^{j_a} \oplus \mathsf{f}^{\mathsf{r}}_{\mathsf{k}^{\mathrm{ID}_{\mathsf{x}}}}(\mathsf{n}^{j_a})\,,\, \mathsf{Mac}^5_{\mathsf{k}^{\mathrm{ID}_{\mathsf{x}}}_{\mathsf{m}}}(\langle \mathrm{GUTI}^{j_a}\,,\, \mathsf{n}^{j_a}\rangle)\rangle, \mathsf{leak}'^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI}^{j_a}\big) \sim \big[\beta_\tau\big]\big(\phi_{\tau_1}, \mathsf{leak}_{\tau_1}, \mathrm{GUTI}\big)}\,\text{PRF-f}}{\big[\beta_\tau\big]\big(\phi_{\tau_1}, \mathsf{leak}_{\tau_1}, \mathrm{GUTI}^{j_a}\big) \sim \big[\beta_\tau\big]\big(\phi_{\tau_1}, \mathsf{leak}_{\tau_1}, \mathrm{GUTI}\big)}\,R$$

We do a similar reasoning to replace $\mathsf{Mac}^5_{\mathsf{k}^{\mathrm{ID}_{\mathsf{x}}}_{\mathsf{m}}}(\langle \mathrm{GUTI}^{j_a}\,,\, \mathsf{n}^{j_a}\rangle)$ with a fresh nonce $\mathsf{n}''_{\mathsf{f}}$ using the PRF-MAC^5 axiom (we omit the details):

$$\dfrac{\big[\beta_\tau\big]\big(\phi^{\mathsf{in}}_{\tau_1}, \langle \mathsf{n}'_{\mathsf{f}}\,,\, \mathsf{n}''_{\mathsf{f}}\rangle, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI}^{j_a}\big) \sim \big[\beta_\tau\big]\big(\phi_{\tau_1}, \mathsf{leak}_{\tau_1}, \mathrm{GUTI}\big)}{\big[\beta_\tau\big]\big(\phi^{\mathsf{in}}_{\tau_1}, \big\langle \mathsf{n}'_{\mathsf{f}}\,,\, \mathsf{Mac}^5_{\mathsf{k}^{\mathrm{ID}_{\mathsf{x}}}_{\mathsf{m}}}(\langle \mathrm{GUTI}^{j_a}\,,\, \mathsf{n}^{j_a}\rangle)\big\rangle, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI}^{j_a}\big) \sim \big[\beta_\tau\big]\big(\phi_{\tau_1}, \mathsf{leak}_{\tau_1}, \mathrm{GUTI}\big)}\,(R + \text{PRF-MAC}^5)^*$$

We then do the same thing on the right side, and use the FA axiom under context

$$\dfrac{\dfrac{\big[\beta_\tau\big]\big(\phi^{\mathsf{in}}_{\tau_1}, \mathsf{n}'_{\mathsf{f}}, \mathsf{n}''_{\mathsf{f}}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI}^{j_a}\big) \sim \big[\beta_\tau\big]\big(\phi^{\mathsf{in}}_{\tau_1}, \mathsf{n}'_{\mathsf{f}}, \mathsf{n}''_{\mathsf{f}}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI}\big)}{\big[\beta_\tau\big]\big(\phi^{\mathsf{in}}_{\tau_1}, \langle \mathsf{n}'_{\mathsf{f}}\,,\, \mathsf{n}''_{\mathsf{f}}\rangle, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI}^{j_a}\big) \sim \big[\beta_\tau\big]\big(\phi^{\mathsf{in}}_{\tau_1}, \langle \mathsf{n}'_{\mathsf{f}}\,,\, \mathsf{n}''_{\mathsf{f}}\rangle, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI}\big)}\,\mathsf{FA_c}}{\big[\beta_\tau\big]\big(\phi^{\mathsf{in}}_{\tau_1}, \langle \mathsf{n}'_{\mathsf{f}}\,,\, \mathsf{n}''_{\mathsf{f}}\rangle, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI}^{j_a}\big) \sim \big[\beta_\tau\big]\big(\phi_{\tau_1}, \mathsf{leak}_{\tau_1}, \mathrm{GUTI}\big)}\,\mathsf{Ax}^*$$

Using the fact that $\beta_\tau \in \mathsf{leak}^{\mathsf{in}}_{\tau_1}$, we have:

$$\dfrac{\big[\beta_\tau\big]\big(\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI}^{j_a}\big), \mathsf{n}'_{\mathsf{f}}, \mathsf{n}''_{\mathsf{f}}, \sim \big[\beta_\tau\big]\big(\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI}\big), \mathsf{n}'_{\mathsf{f}}, \mathsf{n}''_{\mathsf{f}},}{\big[\beta_\tau\big]\big(\phi^{\mathsf{in}}_{\tau_1}, \mathsf{n}'_{\mathsf{f}}, \mathsf{n}''_{\mathsf{f}}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI}^{j_a}\big) \sim \big[\beta_\tau\big]\big(\phi^{\mathsf{in}}_{\tau_1}, \mathsf{n}'_{\mathsf{f}}, \mathsf{n}''_{\mathsf{f}}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI}\big)}\,\mathsf{Simp}$$

We then conclude using Fresh twice:

$$\dfrac{\big[\beta_\tau\big]\big(\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI}^{j_a}\big) \sim \big[\beta_\tau\big]\big(\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI}\big)}{\big[\beta_\tau\big]\big(\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI}^{j_a}\big), \mathsf{n}'_{\mathsf{f}}, \mathsf{n}''_{\mathsf{f}} \sim \big[\beta_\tau\big]\big(\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}, \mathrm{GUTI}\big), \mathsf{n}'_{\mathsf{f}}, \mathsf{n}''_{\mathsf{f}}}\,\mathsf{Fresh}^2$$

- We now sketch the proof of the induction property for the remaining cases:
  - If $\mathsf{ai} = \text{FN}(j)$ with $j \neq j_a$. First, we decompose t_{τ_1} into terms of $\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}$, except for the term:
    $$\left\langle \text{GUTI}^j \oplus \mathsf{f}^{\mathsf{r}}_{\mathsf{k}^{\text{ID}_\mathsf{x}}}(\mathsf{n}^j)\,,\, \mathsf{Mac}^5_{\mathsf{k}^{\text{ID}_\mathsf{x}}_\mathsf{m}}(\langle \text{GUTI}^j\,,\,\mathsf{n}^j\rangle)\right\rangle$$
    The rest of the proof goes as in case $\mathsf{ai} = \text{FN}(j_a)$. On both side, we do the following:
    - We apply the PRF-f axiom to replace $\text{GUTI}^j \oplus \mathsf{f}^{\mathsf{r}}_{\mathsf{k}^{\text{ID}_\mathsf{x}}}(\mathsf{n}^j)$ with $\text{GUTI}^j \oplus \mathsf{n}_\mathsf{f}$ where n_f is a fresh nonce.
    - We use the $\oplus$-ind axiom to replace $\text{GUTI}^j \oplus \mathsf{n}_\mathsf{f}$ with a fresh nonce n'_f
    - We apply the PRF-MAC^5 axiom to replace $\mathsf{Mac}^5_{\mathsf{k}^{\text{ID}_\mathsf{x}}_\mathsf{m}}(\langle \text{GUTI}^j\,,\,\mathsf{n}^j\rangle)$ with a fresh nonce n''_f.

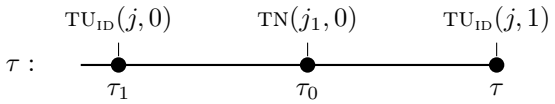
    Finally we use Fresh to get rid of the introduced nonces n'_f and n''_f.
  - If $\mathsf{ai} = \text{TN}(j,0)$. Using the induction hypothesis we know that $\beta_\tau \rightarrow \neg\mathsf{accept}^{\text{ID}_\mathsf{x}}_{\tau_1}$. We can therefore rewrite all occurrences of $\mathsf{accept}^{\text{ID}_\mathsf{x}}_{\tau_1}$ into false under the condition β_τ. This removes all occurrences of $\sigma^{\mathsf{in}}_{\tau_1}(\text{GUTI}^{\text{ID}_\mathsf{x}}_{\text{N}})$ in $\mathsf{leak}_{\tau_1}\backslash\mathsf{leak}^{\mathsf{in}}_{\tau_1}$ and t_{τ_1}. We can then decompose the resulting terms into terms of $\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}$.
  - If $\mathsf{ai} = \text{TN}(j,1)$. We can decompose $\mathsf{leak}_{\tau_1}\backslash\mathsf{leak}^{\mathsf{in}}_{\tau_1}$ and t_{τ_1} into terms of $\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}$ (we use the fact that $\mathsf{leak}^{\mathsf{in}}_{\tau_1}$ contains $\mathsf{Mac}^4_{\mathsf{k}^{\text{ID}_\mathsf{x}}_\mathsf{m}}(\mathsf{n}^j)$).
  - If $\mathsf{ai} = \text{PN}(j,0)$. This is trivial using Fresh.
  - If $\mathsf{ai} = \text{PN}(j,1)$. We use **(Equ3)** to rewrite all occurrences of $\mathsf{accept}^{\text{ID}_\mathsf{x}}_{\tau_1}$ in $\mathsf{leak}_{\tau_1}\backslash\mathsf{leak}^{\mathsf{in}}_{\tau_1}$ and t_{τ_1}:
    $$\mathsf{accept}^{\text{ID}_\mathsf{x}}_{\tau_1} \leftrightarrow \bigvee_{\substack{\tau_2 = \_,\text{PU}_{\text{ID}_\mathsf{x}}(j_2,1)\\ \tau_2 \prec \tau_1}} g(\phi^{\mathsf{in}}_{\tau_2}) = \mathsf{n}^j \wedge g(\phi^{\mathsf{in}}_{\tau_1}) = t_{\tau_2}$$
    We can then decompose the resulting terms into terms of $\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}$. This uses the fact that the terms:
    $$\left(\mathsf{Mac}^2_{\mathsf{k}^{\text{ID}_\mathsf{x}}_\mathsf{m}}(\langle \mathsf{n}^j\,,\,\mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_2}(\text{SQN}^{\text{ID}}_{\text{U}}))\rangle)\right)_{\tau_2 = \_,\text{PU}_{\text{ID}_\mathsf{x}}(j_2,1)\atop \tau_2 \prec \tau_1}$$
    are included in $\mathsf{leak}^{\mathsf{in}}_{\tau_1}$, since $\{\tau_2 = \_, \text{PU}_{\text{ID}_\mathsf{x}}(j_2,1) \mid \tau_2 \prec \tau_1\} = \{\tau_2 = \_, \text{PU}_{\text{ID}_\mathsf{x}}(j_2,1) \mid \tau_2 \prec \tau_b\}$.
  - If ai is a symbolic action of user ID, with $\text{ID} = \text{ID}_\mathsf{x}$, then either $\mathsf{ai} = \text{PU}_{\text{ID}_\mathsf{x}}(j_i,2)$ or $\mathsf{ai} = \text{FU}_{\text{ID}_\mathsf{x}}(j_i)$.
    - If $\mathsf{ai} = \text{PU}_{\text{ID}_\mathsf{x}}(j_i,2)$, then we show using **(Equ2)** that:
      $$\beta_\tau \rightarrow \left(\mathsf{accept}^{\text{ID}_\mathsf{x}}_{\tau_1} \leftrightarrow g(\phi^{\mathsf{in}}_{\tau_1}) = t_{\tau_a}\right)$$
      Therefore we can rewrite $\mathsf{accept}^{\text{ID}_\mathsf{x}}_{\tau_1}$ into $g(\phi^{\mathsf{in}}_{\tau_1}) = t_{\tau_a}$ under β_τ in t_{τ_1}. The resulting term can be easily decomposed into terms of $\phi^{\mathsf{in}}_{\tau_1}, \mathsf{leak}^{\mathsf{in}}_{\tau_1}$.
    - $\mathsf{ai} = \text{FU}_{\text{ID}_\mathsf{x}}(j_i)$. We do a similar reasoning, but using **(Equ1)** instead of **(Equ2)**. We omit the details. ■

4.10.6 Stronger Characterizations

Using the GUTI concealment lemma, we can show the following stronger version of **(Acc3)**:

Lemma 4.13. *For every valid action trace $\tau = \_, \mathsf{ai}$ on $\mathcal{S}_{\mathsf{id}}$ and identity $\text{ID} \in \mathcal{S}_{\mathsf{id}}$:*

- ***(StrAcc1)** If $\mathsf{ai} = \text{TU}_{\text{ID}}(j,1)$. Let $\tau_1 = \_, \text{TU}_{\text{ID}}(j,0)$ such that $\tau_1 \prec \tau$, and let $\mathsf{k} \equiv \mathsf{k}^{\text{ID}}$. Then:*



$$\mathsf{accept}^{\text{ID}}_{\tau} \rightarrow \bigvee_{\substack{\tau_0 = \_,\text{TN}(j_0,0)\\ \tau_1 \prec \tau \tau_0}} \left(\begin{array}{l} \mathsf{accept}^{\text{ID}}_{\tau_0} \wedge g(\phi^{\mathit{in}}_{\tau_0}) = \sigma^{\mathit{in}}_{\tau_1}(\text{GUTI}^{\text{ID}}_{\text{U}}) \wedge \pi_1(g(\phi^{\mathit{in}}_{\tau})) = \mathsf{n}^{j_0} \\ \wedge\ \pi_2(g(\phi^{\mathit{in}}_{\tau})) = \sigma^{\mathit{in}}_{\tau_0}(\text{SQN}^{\text{ID}}_{\text{N}}) \oplus \mathsf{f}_k(\mathsf{n}^{j_0}) \wedge \sigma^{\mathit{in}}_{\tau}(\text{GUTI}^{\text{ID}}_{\text{U}}) = \sigma^{\mathit{in}}_{\tau_0}(\text{GUTI}^{\text{ID}}_{\text{N}}) \end{array}\right)$$

Proof ⋆(p. 120). First, by applying **(Acc3)** we get that:

$$\mathsf{accept}^{\text{ID}}_{\tau} \rightarrow \bigvee_{\substack{\tau_0 = \_, \text{TN}(j_0,0) \\ \tau_0 \prec \tau}} \begin{pmatrix} \mathsf{accept}^{\text{ID}}_{\tau_0} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau})) = \mathsf{n}^{j_0} \wedge \pi_2(g(\phi^{\mathsf{in}}_{\tau})) = \sigma^{\mathsf{in}}_{\tau_0}(\text{SQN}^{\text{ID}}_{\text{N}}) \oplus \mathsf{f}_{\mathsf{k}}(\mathsf{n}^{j_0}) \\ \wedge\, \sigma^{\mathsf{in}}_{\tau}(\text{GUTI}^{\text{ID}}_{\text{U}}) = \sigma^{\mathsf{in}}_{\tau_0}(\text{GUTI}^{\text{ID}}_{\text{N}}) \end{pmatrix} \tag{4.18}$$

We have $\mathsf{accept}^{\text{ID}}_{\tau} \rightarrow \sigma^{\mathsf{in}}_{\tau}(\mathsf{s\text{-}valid\text{-}guti}^{\text{ID}}_{\text{U}})$, and $\sigma^{\mathsf{in}}_{\tau}(\mathsf{s\text{-}valid\text{-}guti}^{\text{ID}}_{\text{U}}) \rightarrow \sigma^{\mathsf{in}}_{\tau_1}(\mathsf{valid\text{-}guti}^{\text{ID}}_{\text{U}})$. Let $\tau_0 = \_, \text{TN}(j_0, 0)$, we know that $\mathsf{accept}^{\text{ID}}_{\tau_0} \rightarrow \sigma^{\mathsf{in}}_{\tau_0}(\text{GUTI}^{\text{ID}}_{\text{N}}) \neq \mathsf{UnSet}$. Therefore:

$$\mathsf{accept}^{\text{ID}}_{\tau} \rightarrow \bigvee_{\substack{\tau_0 = \_, \text{TN}(j_0,0) \\ \tau_0 \prec \tau}} \begin{pmatrix} \mathsf{accept}^{\text{ID}}_{\tau_0} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau})) = \mathsf{n}^{j_0} \wedge \pi_2(g(\phi^{\mathsf{in}}_{\tau})) = \sigma^{\mathsf{in}}_{\tau_0}(\text{SQN}^{\text{ID}}_{\text{N}}) \oplus \mathsf{f}_{\mathsf{k}}(\mathsf{n}^{j_0}) \\ \wedge\, \sigma^{\mathsf{in}}_{\tau}(\text{GUTI}^{\text{ID}}_{\text{U}}) = \sigma^{\mathsf{in}}_{\tau_0}(\text{GUTI}^{\text{ID}}_{\text{N}}) \wedge \sigma^{\mathsf{in}}_{\tau}(\text{GUTI}^{\text{ID}}_{\text{U}}) \neq \mathsf{UnSet} \wedge \sigma^{\mathsf{in}}_{\tau_1}(\mathsf{valid\text{-}guti}^{\text{ID}}_{\text{U}}) \end{pmatrix}$$

We want to get a contradiction if $\tau_0 \prec \tau_1$. Let $\tau_0 = \_, \text{TN}(j_0, 0) \prec \tau$, and assume that $\tau_0 \prec \tau_1$. If there does not exists any τ_2 such that $\tau_2 = \_, \text{FU}_{\text{ID}}(j_i) \prec \tau_1$, then it is easy to show that $\sigma^{\mathsf{in}}_{\tau}(\text{GUTI}^{\text{ID}}_{\text{U}}) = \mathsf{UnSet}$. In that case, from the equation above we get that $\neg\mathsf{accept}^{\text{ID}}_{\tau}$, which concludes this case.

Therefore, let τ_2 be maximal w.r.t. $\prec$ such that $\tau_2 = \_, \text{FU}_{\text{ID}}(j_i) \prec \tau_1$. We have $\tau_2 \not\prec_{\tau} \text{FU}_{\text{ID}}(\_)$. Assume that there exists a user ID action between τ_2 and τ_1. It is easy to show by induction over τ' in $\tau_2 \prec \tau' \preceq \tau_1$ that, since there are no $\text{FU}_{\text{ID}}(\_)$ action between τ_2 and τ_1, we have $\neg\sigma^{\mathsf{in}}_{\tau_1}(\mathsf{valid\text{-}guti}^{\text{ID}}_{\text{U}})$. This implies $\neg\mathsf{accept}^{\text{ID}}_{\tau}$, which concludes this case.

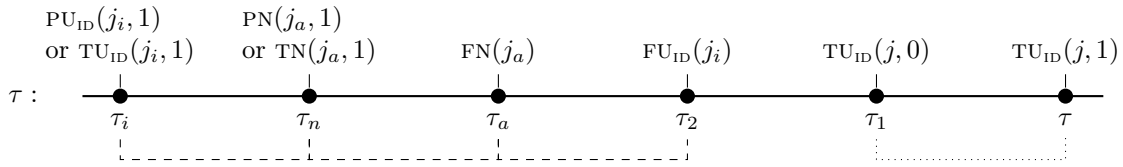
Therefore we can safely assume that there are no user ID actions between τ_2 and τ_1. We deduce that $\sigma^{\mathsf{in}}_{\tau_1}(\mathsf{valid\text{-}guti}^{\text{ID}}_{\text{U}}) \rightarrow \mathsf{accept}^{\text{ID}}_{\tau_2}$. Hence $\mathsf{accept}^{\text{ID}}_{\tau} \rightarrow \mathsf{accept}^{\text{ID}}_{\tau_2}$. By applying **(Equ1)** to τ_2, we know that:

$$\mathsf{accept}^{\text{ID}}_{\tau} \rightarrow \bigvee_{\substack{\tau_a = \_, \text{FN}(j_a) \prec \tau_2 \\ \tau_a \not\prec_{\tau} \text{NS}_{\text{ID}}(\_)}} \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_2} \tag{4.19}$$

We recall that:

$$\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_2} \equiv \begin{pmatrix} \mathsf{inj\text{-}auth}_{\tau_2}(\text{ID}, j_a) \wedge \sigma^{\mathsf{in}}_{\tau_2}(\mathsf{e\text{-}auth}^{j_a}_{\text{N}}) \neq \mathsf{UnknownId} \\ \wedge\, \pi_1(g(\phi^{\mathsf{in}}_{\tau_2})) = \text{GUTI}^{j_a} \oplus \mathsf{f}^{\mathsf{r}}_{\mathsf{k}}(\mathsf{n}^{j_a}) \wedge\ \pi_2(g(\phi^{\mathsf{in}}_{\tau_2})) = \mathsf{Mac}^5_{\mathsf{k_m}}(\langle \text{GUTI}^{j_a}, \mathsf{n}^{j_a} \rangle) \end{pmatrix}$$

Let $\tau_a = \_, \text{FN}(j_a) \prec \tau_2$ such that $\tau_a \not\prec_{\tau} \text{NS}_{\text{ID}}(\_)$. We know that there exists $\tau_n = \_, \text{PN}(j_a, 1)$ or $\tau_n = \_, \text{TN}(j_a, 1)$ such that $\tau_n \prec \tau_a$, and that $\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_2} \rightarrow \mathsf{accept}^{\text{ID}}_{\tau_n}$. Let $\tau_i = \_, \text{PU}_{\text{ID}}(j_i, 1)$ or $\_, \text{TU}_{\text{ID}}(j_i, 1)$ such that $\tau_i \prec \tau_2$. If $\tau_n \prec \tau_i$, we show using **(Acc1)** if $\tau_n = \_, \text{PN}(j_a, 1)$ or **(Acc4)** if $\tau_n = \_, \text{PN}(j_a, 1)$ that we have $\neg\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_2}$. Therefore, we assume that $\tau_i \prec \tau_n$. We depict the situation below:



We check that $\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_2} \rightarrow \sigma_{\tau_2}(\text{GUTI}^{\text{ID}}_{\text{U}}) = \text{GUTI}^{j_a}$. Moreover, since there are no user ID actions between τ_2 and τ_1 or between τ_1 and τ, $\sigma_{\tau_2}(\text{GUTI}^{\text{ID}}_{\text{U}}) = \sigma^{\mathsf{in}}_{\tau}(\text{GUTI}^{\text{ID}}_{\text{U}})$. From (4.18), we know that $\mathsf{accept}^{\text{ID}}_{\tau} \rightarrow \sigma^{\mathsf{in}}_{\tau}(\text{GUTI}^{\text{ID}}_{\text{U}}) = \sigma^{\mathsf{in}}_{\tau_0}(\text{GUTI}^{\text{ID}}_{\text{N}})$. It follows that:

$$\mathsf{accept}^{\text{ID}}_{\tau} \wedge \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_2} \rightarrow \sigma^{\mathsf{in}}_{\tau_0}(\text{GUTI}^{\text{ID}}_{\text{N}}) = \text{GUTI}^{j_a} \tag{4.20}$$

If $\tau_0 \prec \tau_n$, then it is easy to check that $\sigma^{\mathsf{in}}_{\tau_0}(\text{GUTI}^{\text{ID}}_{\text{N}}) \neq \text{GUTI}^{j_a}$. Therefore we have $\neg(\mathsf{accept}^{\text{ID}}_{\tau} \wedge \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_2})$.

Now, we assume that $\tau_n \prec \tau_0$. Recall that we assumed $\tau_0 \prec \tau_1$. Our goal is to apply the GUTI concealment lemma (Lemma 4.12) to τ_0 get a contradiction. We can check that the following hypothesis of Lemma 4.12 is true:

$$\{\tau' \mid \tau_i \prec_{\tau_0} \tau_b\} \cap \{\text{PU}_{\text{ID}}(j, \_), \text{TU}_{\text{ID}}(j, \_), \text{FU}_{\text{ID}}(j) \mid j \in \mathbb{N}\} \subseteq \{\text{PU}_{\text{ID}}(j_i, 2), \text{FU}_{\text{ID}}(j_i)\}$$

We deduce that:

$$\mathsf{inc\text{-}accept}^{\text{ID}}_{\tau_n} \wedge \sigma_{\tau_i}(\mathsf{b\text{-}auth}^{\text{ID}}_{\text{U}}) = \mathsf{n}^{j_a} \wedge \mathsf{accept}^{\text{ID}\times}_{\tau_i} \rightarrow g(\phi^{\mathsf{in}}_{\tau_0}) \neq \text{GUTI}^{j_a} \tag{4.21}$$

We know that:

$$\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_2} \rightarrow \mathsf{accept}^{\mathrm{ID}}_{\tau_i} \wedge \sigma_{\tau_i}(\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^{j_a} \tag{4.22}$$

Moreover, $\neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_n} \rightarrow \sigma_{\tau_n}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \neq \mathrm{GUTI}^{j_a}$. It is then straightforward to check that $\neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_n} \rightarrow \sigma_{\tau_0}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \neq \mathrm{GUTI}^{j_a}$. Therefore, using (4.20) we get that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \wedge \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_2} \wedge \neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_n} \rightarrow \left(\sigma^{\mathsf{in}}_{\tau_0}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) = \mathrm{GUTI}^{j_a} \wedge \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \neq \mathrm{GUTI}^{j_a}\right) \rightarrow \mathsf{false}$$

Hence $\mathsf{accept}^{\mathrm{ID}}_{\tau} \wedge \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_2} \rightarrow \mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_n}$. Therefore using (4.21) and (4.22), we get:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \wedge \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_2} \rightarrow g(\phi^{\mathsf{in}}_{\tau_0}) \neq \mathrm{GUTI}^{j_a} \tag{4.23}$$

We have $\mathsf{accept}^{\mathrm{ID}}_{\tau_0} \rightarrow g(\phi^{\mathsf{in}}_{\tau_0}) = \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}})$. We get from this, (4.20) and (4.23) that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \wedge \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_2} \wedge \mathsf{accept}^{\mathrm{ID}}_{\tau_0} \rightarrow \mathsf{false}$$

This holds for every $\tau_a = \_, \mathrm{FN}(j_a) \prec \tau_2$. We deduce from (4.19) that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \wedge \mathsf{accept}^{\mathrm{ID}}_{\tau_0} \rightarrow \mathsf{false}$$

Since we have this for every $\tau_0 \prec \tau_1$, we can rewrite (4.18) to get:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \rightarrow \bigvee_{\substack{\tau_0 = \_, \mathrm{TN}(j_0,0) \\ \tau_1 \prec \tau_0 \prec \tau}} \begin{pmatrix} \mathsf{accept}^{\mathrm{ID}}_{\tau_0} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau})) = \mathsf{n}^{j_0} \wedge \pi_2(g(\phi^{\mathsf{in}}_{\tau})) = \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \oplus \mathsf{f}_{\mathsf{k}}(\mathsf{n}^{j_0}) \\ \wedge\, \sigma^{\mathsf{in}}_{\tau}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \end{pmatrix} \tag{4.24}$$

To conclude, we observe that $\mathsf{accept}^{\mathrm{ID}}_{\tau} \wedge \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_2} \rightarrow \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \mathrm{GUTI}^{j_a}$. We recall that $\mathsf{accept}^{\mathrm{ID}}_{\tau_0} \rightarrow g(\phi^{\mathsf{in}}_{\tau_0}) = \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}})$. We conclude using (4.20) that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \wedge \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_2} \rightarrow \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = g(\phi^{\mathsf{in}}_{\tau_0})$$

Since this holds for every $\tau_a = \_, \mathrm{FN}(j_a) \prec \tau_2$, we deduce from (4.19) that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \wedge \mathsf{accept}^{\mathrm{ID}}_{\tau_0} \rightarrow \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = g(\phi^{\mathsf{in}}_{\tau_0})$$

Hence using (4.24) we get:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \rightarrow \bigvee_{\substack{\tau_0 = \_, \mathrm{TN}(j_0,0) \\ \tau_1 \prec \tau_0 \prec \tau}} \begin{pmatrix} \mathsf{accept}^{\mathrm{ID}}_{\tau_0} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau})) = \mathsf{n}^{j_0} \wedge \pi_2(g(\phi^{\mathsf{in}}_{\tau})) = \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \oplus \mathsf{f}_{\mathsf{k}}(\mathsf{n}^{j_0}) \\ \wedge\, \sigma^{\mathsf{in}}_{\tau}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \wedge \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = g(\phi^{\mathsf{in}}_{\tau_0}) \end{pmatrix}$$

■

We now prove the following strong acceptance characterization properties:

Lemma 4.14. *For every valid action trace $\tau = \_, \mathsf{ai}$ on $\mathcal{S}_{\mathsf{id}}$ and identity $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$:*

- ***(StrEqu1)*** *If $\mathsf{ai} = \mathrm{FU}_{\mathrm{ID}}(j)$. Let $\tau_2 = \_, \mathrm{TU}_{\mathrm{ID}}(j,0)$ or $\_, \mathrm{PU}_{\mathrm{ID}}(j,1)$ such that $\tau_2 \prec \tau$, then:*

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \leftrightarrow \bigvee_{\tau_2 \prec_\tau \tau_1 = \_, \mathrm{FN}(j_0)} \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau}$$

- ***(StrEqu2)*** *If $\mathsf{ai} = \mathrm{TU}_{\mathrm{ID}}(j,1)$. Let $\tau_2 = \_, \mathrm{TU}_{\mathrm{ID}}(j,0)$ such that $\tau_2 \prec \tau$. Then for every τ_1 such that $\tau_1 = \_, \mathrm{TN}(j_1,0)$ and $\tau_2 \prec_\tau \tau_1$, we let:*

$$\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \equiv \begin{pmatrix} \pi_1(g(\phi^{\mathsf{in}}_{\tau})) = \mathsf{n}^{j_1} \;\wedge\; \pi_2(g(\phi^{\mathsf{in}}_{\tau})) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \oplus \mathsf{f}_{\mathsf{k}^{\mathrm{ID}}}(\mathsf{n}^{j_1}) \\ \wedge\; \pi_3(g(\phi^{\mathsf{in}}_{\tau})) = \mathsf{Mac}^3_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle \mathsf{n}^{j_1}, \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}), \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) \rangle) \\ \wedge\; g(\phi^{\mathsf{in}}_{\tau_1}) = \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) \;\wedge\; \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \;\wedge\; \sigma^{\mathsf{in}}_{\tau_2}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \\ \wedge\; \mathsf{range}(\sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}), \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})) \end{pmatrix}$$

Then:

$$\left(\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \rightarrow \mathsf{accept}^{\mathrm{ID}}_{\tau} \wedge \mathsf{accept}^{\mathrm{ID}}_{\tau_1}\right)_{\substack{\tau_1 = \_, \mathrm{TN}(j_1,0) \\ \tau_2 \prec_\tau \tau_1}} \qquad \mathsf{accept}^{\mathrm{ID}}_{\tau} \leftrightarrow \bigvee_{\substack{\tau_1 = \_, \mathrm{TN}(j_1,0) \\ \tau_2 \prec_\tau \tau_1}} \mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau}$$

- ***(StrEqu3)*** *If* $\mathsf{ai} = \mathrm{TN}(j,1)$. *Let* $\tau_1 = \_, \mathrm{TN}(j,0)$ *such that* $\tau_1 \prec \tau$. *Let* $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$ *and* τ_i, τ_2 *be such that* $\tau_i = \_, \mathrm{TU}_{\mathrm{ID}}(j_i,1)$, $\tau_2 = \_, \mathrm{TU}_{\mathrm{ID}}(j_i,0)$ *and* $\tau_2 \prec_\tau \tau_1 \prec_\tau \tau_i$. *Let:*

$$\mathsf{full\text{-}tr}^{\mathsf{n}:\tau_1,\tau}_{\mathsf{u}:\tau_2,\tau_i} \equiv \mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau_i} \wedge g(\phi^{\mathsf{in}}_\tau) = \mathsf{Mac}^4_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{im}}}(\mathsf{n}^j)$$

Then:

$$\left(\mathsf{full\text{-}tr}^{\mathsf{n}:\tau_1,\tau}_{\mathsf{u}:\tau_2,\tau_i} \rightarrow \mathsf{accept}^{\mathrm{ID}}_\tau \wedge \mathsf{accept}^{\mathrm{ID}}_{\tau_i} \wedge \mathsf{accept}^{\mathrm{ID}}_{\tau_1}\right)_{\substack{\tau_2=\_,\mathrm{TU}_{\mathrm{ID}}(j_i,0)\\ \tau_i=\_,\mathrm{TU}_{\mathrm{ID}}(j_i,1)\\ \tau_2\prec_\tau\tau_1\prec_\tau\tau_i}} \qquad \mathsf{accept}^{\mathrm{ID}}_\tau \leftrightarrow \bigvee_{\substack{\tau_2=\_,\mathrm{TU}_{\mathrm{ID}}(j_i,0)\\ \tau_i=\_,\mathrm{TU}_{\mathrm{ID}}(j_i,1)\\ \tau_2\prec_\tau\tau_1\prec_\tau\tau_i}} \mathsf{full\text{-}tr}^{\mathsf{n}:\tau_1,\tau}_{\mathsf{u}:\tau_2,\tau_i}$$

- ***(StrEqu4)*** *If* $\mathsf{ai} = \mathrm{PU}_{\mathrm{ID}}(j,2)$ *then for every* $\tau_1 = \_, \mathrm{PN}(j_1,1)$ *such that* $\tau_2 \prec_\tau \tau_1$, *we have:*

$$\neg\sigma^{\mathit{in}}_\tau(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \rightarrow \mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_1} \wedge \sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) - \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = 0$$

Moreover:

$$\neg\sigma^{\mathit{in}}_\tau(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{accept}^{\mathrm{ID}}_\tau \rightarrow \sigma_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) - \sigma_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = 0$$

4.10.7 ⋆ (p. 127) Proof of Lemma 4.14

Proof of ***(StrEqu1)***. First, we apply **(Equ1)**:

$$\mathsf{accept}^{\mathrm{ID}}_\tau \leftrightarrow \bigvee_{\substack{\tau_1=\_,\mathrm{FN}(j_0)\prec\tau\\ \tau_1\not\prec_\tau \mathrm{NS}_{\mathrm{ID}}(\_)}} \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau}$$

Let $\tau_1 = \_, \mathrm{FN}(j_0) \prec \tau$. Remark that if $\tau_2 \prec \tau_1$ then $\tau_1 \not\prec_\tau \mathrm{NS}_{\mathrm{ID}}(\_)$. Hence to conclude we just need to show that if $\tau_1 \prec \tau_2$ then $\neg\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau}$.

Let $\tau_i = \_, \mathrm{PU}_{\mathrm{ID}}(j,2)$ or $\_, \mathrm{TU}_{\mathrm{ID}}(j,1)$ such that $\tau_i \prec \tau$. We do a case disjunction on τ_i:

- If $\tau_i = \_, \mathrm{PU}_{\mathrm{ID}}(j,2)$. We know that $\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau} \rightarrow \mathsf{accept}^{\mathrm{ID}}_{\tau_i}$, hence by applying **(Acc2)** to τ_i:

$$\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau} \rightarrow \bigvee_{\substack{\tau_x=\_,\mathrm{PN}(j_x,1)\\ \tau_2\prec\tau_x\prec\tau_i}} \mathsf{accept}^{\mathrm{ID}}_{\tau_x} \wedge g(\phi^{\mathsf{in}}_{\tau_2}) = \mathsf{n}^{j_x} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau_x})) = \{\langle \mathrm{ID}, \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^j_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}}$$

  We know that $\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau} \rightarrow g(\phi^{\mathsf{in}}_{\tau_2}) = \mathsf{n}^{j_0}$. We deduce that the main term of the disjunction above is false whenever $j_x \neq j_0$. Hence we have $\neg\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau}$ if there does not exist any τ_0 such that $\tau_2 \prec \tau_0 \prec \tau_i$ and $\tau_0 = \_, \mathrm{PN}(j_0,1)$.

  If $\tau_1 \prec \tau_2$ then we know that for every τ_0, if $\tau_0 = \_, \mathrm{PN}(j_0,1) \prec \tau$ then $\tau_0 \prec \tau_1$, and by transitivity $\tau_0 \prec \tau_2$. Hence there does not exist any τ_0 such that $\tau_2 \prec \tau_0 \prec \tau_i$ and $\tau_0 = \_, \mathrm{PN}(j_0,1)$. We deduce that if $\tau_1 \prec \tau_2$ then $\neg\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau}$ holds, which is what we wanted.

- If $\tau_i = \_, \mathrm{TU}_{\mathrm{ID}}(j,1)$. We know that $\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau} \rightarrow \mathsf{accept}^{\mathrm{ID}}_{\tau_i}$, hence by applying **(StrAcc1)** to τ_i:

$$\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau} \rightarrow \bigvee_{\substack{\tau_x=\_,\mathrm{TN}(j_x,0)\\ \tau_2\prec\tau_x\prec\tau_i}} \begin{pmatrix} \mathsf{accept}^{\mathrm{ID}}_{\tau_x} \wedge g(\phi^{\mathsf{in}}_{\tau_x}) = \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau_i})) = \mathsf{n}^{j_x} \\ \wedge\ \pi_2(g(\phi^{\mathsf{in}}_{\tau_i})) = \sigma^{\mathsf{in}}_{\tau_x}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \oplus \mathsf{f}_{\mathsf{k}}(\mathsf{n}^{j_x}) \wedge \sigma^{\mathsf{in}}_{\tau_i}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_x}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \end{pmatrix}$$

  Similarly to what we did for $\tau_i = \_, \mathrm{PU}_{\mathrm{ID}}(j_i,2)$, the main term above if false if $j_x \neq j_0$. Hence we have $\neg\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau}$ if there does not exist any τ_0 such that $\tau_2 \prec \tau_0 \prec \tau_i$ and $\tau_0 = \_, \mathrm{TN}(j_0,0)$. Since this is the case whenever $\tau_1 \prec \tau_2$, we deduce that if $\tau_1 \prec \tau_2$ then $\neg\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau}$ holds. ■

Proof of ***(StrEqu2)***. We repeating the proof of **(Equ4)**, but using **(StrAcc1)** instead of **(Acc3)**. All the reasonings we did apply, only the set of τ_1 the disjunction quantifies upon changes. We quantify over τ_1 in $\{\tau_1 \mid \tau_1 = \_, \mathrm{TN}(j_0,0) \wedge \tau_2 \prec_\tau \tau_1\}$ instead of $\{\tau_1 \mid \tau_1 = \_, \mathrm{TN}(j_0,0) \wedge \tau_1 \prec \tau\}$. We get that:

$$\mathsf{accept}^{\mathrm{ID}}_\tau \leftrightarrow \bigvee_{\substack{\tau_1=\_,\mathrm{TN}(j_0,0)\\ \tau_2\prec_\tau\tau_1}} \begin{pmatrix} \pi_3(g(\phi^{\mathsf{in}}_\tau)) = \mathsf{Mac}^3_{\mathsf{k}_{\mathsf{m}}}(\langle \mathsf{n}^{j_0}, \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}), \sigma^{\mathsf{in}}_\tau(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}})\rangle) \wedge \sigma^{\mathsf{in}}_\tau(\mathsf{s\text{-}valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \\ \wedge\ \mathsf{range}(\sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}), \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})) \wedge g(\phi^{\mathsf{in}}_{\tau_1}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \wedge \pi_1(g(\phi^{\mathsf{in}}_\tau)) = \mathsf{n}^{j_0} \\ \wedge\ \pi_2(g(\phi^{\mathsf{in}}_\tau)) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \oplus \mathsf{f}_{\mathsf{k}}(\mathsf{n}^{j_0}) \wedge \sigma^{\mathsf{in}}_\tau(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \end{pmatrix}$$

Since no user ID action occurs between τ_2 and τ, we know that:

$$\sigma^{\mathsf{in}}_{\tau}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) \qquad \sigma^{\mathsf{in}}_{\tau}(\mathsf{s\text{-}valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \leftrightarrow \sigma^{\mathsf{in}}_{\tau_2}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}})$$

Using this, we can rewrite the characterization of $\mathsf{accept}^{\mathrm{ID}}_{\tau}$ as follows (we underline the subterms where rewriting occurred):

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \leftrightarrow \bigvee_{\substack{\tau_1 = \_, \mathrm{TN}(j_0,0) \\ \tau_2 \prec_\tau \tau_1}} \begin{pmatrix} \pi_3(g(\phi^{\mathsf{in}}_{\tau})) = \mathsf{Mac}^3_{\mathsf{k_m}}(\langle \mathsf{n}^{j_0}, \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}), \underline{\sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}})}\rangle) \wedge \underline{\sigma^{\mathsf{in}}_{\tau_2}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}})} \\ \wedge\, \mathsf{range}(\sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}), \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})) \wedge g(\phi^{\mathsf{in}}_{\tau_1}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau})) = \mathsf{n}^{j_0} \\ \wedge\, \pi_2(g(\phi^{\mathsf{in}}_{\tau})) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \oplus \mathsf{f}_{\mathsf{k}}(\mathsf{n}^{j_0}) \wedge \underline{\sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}})} = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \end{pmatrix}$$

We rewrite $\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}})$ into $\sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}})$:

$$\leftrightarrow \bigvee_{\substack{\tau_1 = \_, \mathrm{TN}(j_0,0) \\ \tau_2 \prec_\tau \tau_1}} \begin{pmatrix} \pi_3(g(\phi^{\mathsf{in}}_{\tau})) = \mathsf{Mac}^3_{\mathsf{k_m}}(\langle \mathsf{n}^{j_0}, \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}), \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}})\rangle) \wedge \sigma^{\mathsf{in}}_{\tau_2}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \\ \wedge\, \mathsf{range}(\sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}), \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})) \wedge g(\phi^{\mathsf{in}}_{\tau_1}) = \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau})) = \mathsf{n}^{j_0} \\ \wedge\, \pi_2(g(\phi^{\mathsf{in}}_{\tau})) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \oplus \mathsf{f}_{\mathsf{k}}(\mathsf{n}^{j_0}) \wedge \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \end{pmatrix}$$

Finally we re-order the conjuncts:

$$\leftrightarrow \bigvee_{\substack{\tau_1 = \_, \mathrm{TN}(j_0,0) \\ \tau_2 \prec_\tau \tau_1}} \begin{pmatrix} \pi_1(g(\phi^{\mathsf{in}}_{\tau})) = \mathsf{n}^{j_1} \;\wedge\; \pi_2(g(\phi^{\mathsf{in}}_{\tau})) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \oplus \mathsf{f}_{\mathsf{k}^{\mathrm{ID}}}(\mathsf{n}^{j_1}) \\ \wedge\; \pi_3(g(\phi^{\mathsf{in}}_{\tau})) = \mathsf{Mac}^3_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle \mathsf{n}^{j_1}, \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}), \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}})\rangle) \\ \wedge\; g(\phi^{\mathsf{in}}_{\tau_1}) = \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) \;\wedge\; \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \;\wedge\; \sigma^{\mathsf{in}}_{\tau_2}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \\ \wedge\; \mathsf{range}(\sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}), \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})) \end{pmatrix}$$

$$\leftrightarrow \bigvee_{\substack{\tau_1 = \_, \mathrm{TN}(j_0,0) \\ \tau_2 \prec_\tau \tau_1}} \mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau}$$

Finally, for every $\tau_1 = \_, \mathrm{TN}(j_1, 0)\tau_2 \prec_\tau \tau_1$ we can check that:

$$\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \;\rightarrow\; \mathsf{accept}^{\mathrm{ID}}_{\tau} \wedge \mathsf{accept}^{\mathrm{ID}}_{\tau_1}$$ ■

*Proof of **(StrEqu3)**.* The proof that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \leftrightarrow \bigvee_{\substack{\tau_2 = \_, \mathrm{TU}_{\mathrm{ID}}(j_i,0) \\ \tau_i = \_, \mathrm{TU}_{\mathrm{ID}}(j_i,1) \\ \tau_2 \prec_\tau \tau_1 \prec_\tau \tau_i}} \mathsf{full\text{-}tr}^{\mathsf{n}:\tau_1,\tau}_{\mathsf{u}:\tau_2,\tau_i}$$

is exactly the same than the proof of **(Equ5)**, but using **(StrEqu2)** instead of **(Equ4)**.

Finally, it is straightforward to check that for every $\tau_2 = \_, \mathrm{TU}_{\mathrm{ID}}(j_i, 0)$, $\tau_i = \_, \mathrm{TU}_{\mathrm{ID}}(j_i, 1)$ such that $\tau_2 \prec_\tau \tau_1 \prec_\tau \tau_i$ we have:

$$\mathsf{full\text{-}tr}^{\mathsf{n}:\tau_1,\tau}_{\mathsf{u}:\tau_2,\tau_i} \;\rightarrow\; \mathsf{accept}^{\mathrm{ID}}_{\tau} \wedge \mathsf{accept}^{\mathrm{ID}}_{\tau_i} \wedge \mathsf{accept}^{\mathrm{ID}}_{\tau_1}$$ ■

*Proof of **(StrEqu4)**.* Let $\tau_2 = \_\mathrm{PU}_{\mathrm{ID}}(j, 1)$ such that $\tau_2 \prec \tau$. Using **(Equ2)**, we know that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \leftrightarrow \bigvee_{\substack{\tau_1 = \_, \mathrm{PN}(j_1,1) \\ \tau_2 \prec_\tau \tau_1}} \mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau}$$

Therefore to prove **(StrEqu4)** it is sufficient to show that for every τ_1 such that $\tau_1 = \_, \mathrm{PN}(j_1, 1)$ and $\tau_2 \prec_\tau \tau_1$ we have:

$$\neg\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \;\rightarrow\; \mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_1} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) - \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = 0 \wedge \sigma_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) - \sigma_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = 0$$

Hence let τ_1 with $\tau_1 = \_, \mathrm{PN}(j_1, 1)$ and $\tau_2 \prec_\tau \tau_1$.

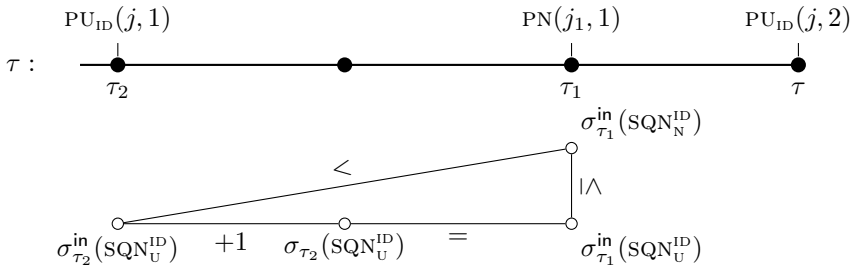


Figure 4.21: First Graphical Representation Used in the Proof of Lemma 4.14

Part 1 First, we are going to show that:

$$\neg\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \rightarrow \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = \sigma_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) \tag{4.25}$$

We know that $\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_1} \rightarrow \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = \sigma_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$, which is what we wanted. Hence it only remains to show (4.25) when $\neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_1}$. Using **(B5)** we know that $\sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \leq \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$. By validity of τ there are no user action between τ_2 and τ, hence $\sigma_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$. Observe that:

$$\mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \wedge \neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_1} \rightarrow \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) > \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) \qquad \sigma_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) + 1$$

We summarize this graphically in Figure 4.21. We deduce that:

$$\begin{aligned}\neg\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \wedge \neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_1} &\rightarrow \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) < \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \leq \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) + 1\\ &\rightarrow \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) + 1\\ &\rightarrow \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = \sigma_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\end{aligned} \tag{4.26}$$

Which is what we wanted to show.

Part 2 We now show that:

$$\neg\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \rightarrow \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) > \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \tag{4.27}$$

First, notice that:

$$\begin{aligned}\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_1} &\rightarrow \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) + 1\\ &\rightarrow \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) > \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})\\ &\rightarrow \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) > \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \qquad \text{(By } \mathbf{(B1)}\text{)}\end{aligned}$$

Therefore we only need to prove:

$$\neg\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \wedge \neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_1} \rightarrow \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) > \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$$

Which is straightforward:

$$\begin{aligned}\neg\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \wedge \neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_1} &\rightarrow \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) + 1 && \text{(By (4.26))}\\ &\rightarrow \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) > \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\\ &\rightarrow \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) > \sigma_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) && \text{(By } \mathbf{(B5)}\text{)}\end{aligned}$$

Which concludes the proof of (4.27).

Part 3 We give the proof of:

$$\neg\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \rightarrow \sigma_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \wedge \sigma_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \tag{4.28}$$

By validity of τ we know that $\sigma_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$, therefore using (4.25) we know that:

$$\neg\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \rightarrow \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = \sigma_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$$

To conclude, we need to show that $\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}$ was kept unchanged since τ_1, i.e. that $\neg\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau}$ implies that $\sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = \sigma_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$. This requires that no SUPI or GUTI network session incremented $\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}$. Therefore we need to show the two following properties:

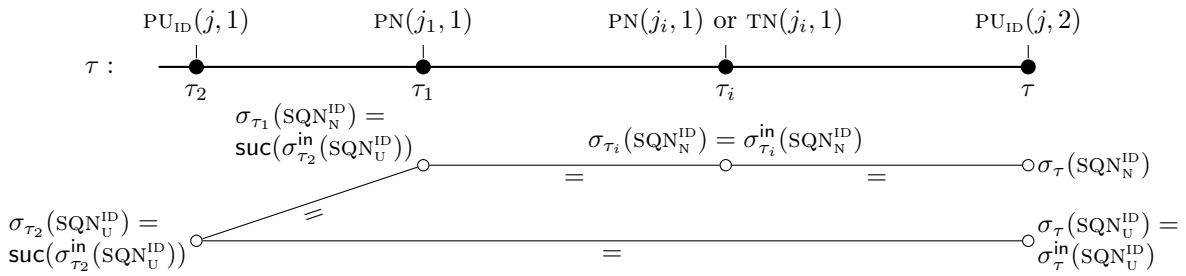


Figure 4.22: Second Graphical Representation Used in the Proof of Lemma 4.14

- **SUPI:** For every $\tau_1 \prec_\tau \tau_i$ such that $\tau_i = \_, \mathrm{PN}(j_i, 1)$:

$$\neg\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \rightarrow \neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_i} \tag{4.29}$$

- **GUTI:** For every $\tau_1 \prec_\tau \tau_i$ such that $\tau_i = \_, \mathrm{TN}(j_i, 1)$:

$$\neg\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \rightarrow \neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_i} \tag{4.30}$$

Assuming the two properties above, showing that (4.28) holds is easy. First, using (4.29) and (4.30) we know that:

$$\neg\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \rightarrow \sigma_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$$

We know that $\sigma_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$. We deduce that $\sigma_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = \sigma_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$, which concludes this case. We summarize this graphically in Figure 4.22.

Part 4 (Proof of (4.29)) Let $\tau_1 \prec_\tau \tau_i$ such that $\tau_i = \_, \mathrm{PN}(j_i, 1)$. Using **(Acc1)** we know that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau_i} \rightarrow \bigvee_{\tau'=\_,\mathrm{PU}_{\mathrm{ID}}(j',1)\prec_\tau\tau_i} \pi_1(g(\phi^{\mathsf{in}}_{\tau_i})) = \{\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_{\tau'}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j'}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} \wedge g(\phi^{\mathsf{in}}_{\tau'}) = \mathsf{n}^{j_i}$$

We know that $\mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \rightarrow g(\phi^{\mathsf{in}}_{\tau_2}) = \mathsf{n}^{j_1} \neq \mathsf{n}^{j_i}$. Moreover from the validity of τ we know that for every τ'' such that:

$$\tau_2 = \_, \mathrm{PU}_{\mathrm{ID}}(j,1) \prec_\tau \tau'' = \_, \mathsf{ai}'' \prec_\tau \tau = \_, \mathrm{PU}_{\mathrm{ID}}(j,2)$$

We have $\mathsf{ai}'' \neq \mathrm{PU}_{\mathrm{ID}}(\_,\_)$. Hence:

$$\mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \wedge \mathsf{accept}^{\mathrm{ID}}_{\tau_i} \rightarrow \bigvee_{\tau'=\_,\mathrm{PU}_{\mathrm{ID}}(j',1)\prec_\tau\tau_2} \pi_1(g(\phi^{\mathsf{in}}_{\tau_i})) = \{\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_{\tau'}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j'}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} \wedge g(\phi^{\mathsf{in}}_{\tau'}) = \mathsf{n}^{j_i}$$

Which implies that:

$$\mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \wedge \mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_i} \rightarrow \bigvee_{\tau'=\_,\mathrm{PU}_{\mathrm{ID}}(j',1)\prec_\tau\tau_2} \sigma_{\tau_i}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau'}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))$$

We recall (4.25):

$$\neg\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \rightarrow \sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = \sigma_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$$

Let $\tau' = \_, \mathrm{PU}_{\mathrm{ID}}(j',1) \prec_\tau \tau_2$. We know using **(B1)** that:

$$\sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \leq \sigma_{\tau_i}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \qquad\qquad \sigma_{\tau'}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) \leq \sigma_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$$

Moreover using **(A2)** we know that $\sigma_{\tau'}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) \neq \sigma_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$, hence $\sigma_{\tau'}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) < \sigma_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$. We summarize what we know graphically in Figure 4.23. Therefore:

$$\neg\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \wedge \mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_i}$$

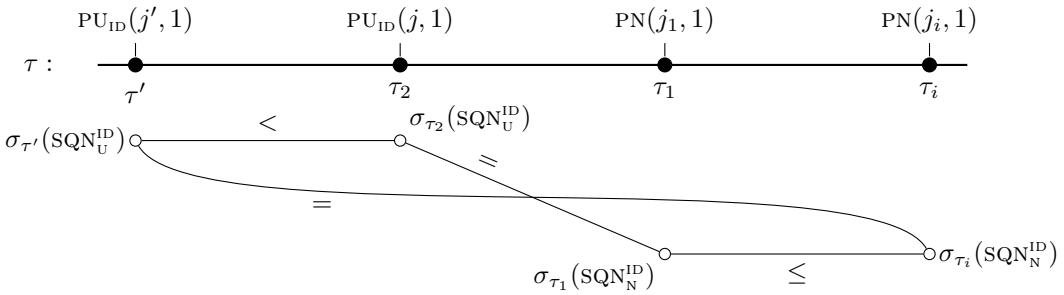


Figure 4.23: Third Graphical Representation Used in the Proof of Lemma 4.14

$$
\begin{aligned}
&\rightarrow \bigvee_{\tau'=\_,\mathrm{PU}_{\mathrm{ID}}(j',1)\prec_\tau\tau_2} \begin{pmatrix} \sigma_{\tau'}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) < \sigma_{\tau_2}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) \wedge \sigma_{\tau_2}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) = \sigma_{\tau_1}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) \\ \wedge \sigma_{\tau_1}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) \leq \sigma_{\tau_i}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) \wedge \sigma_{\tau_i}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) = \sigma_{\tau'}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) \end{pmatrix} \\
&\rightarrow \bigvee_{\tau'=\_,\mathrm{PU}_{\mathrm{ID}}(j',1)\prec_\tau\tau_2} \sigma_{\tau'}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) < \sigma_{\tau'}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) \\
&\rightarrow \mathsf{false}
\end{aligned}
$$

Which concludes this proof.

Part 5 (Proof of (4.30)) Let $\tau_1 \prec_\tau \tau_i$ such that $\tau_i = \_, \mathrm{TN}(j_i,1)$. Using Lemma 4.7, we know that:

$$\mathsf{accept}_{\tau_i}^{\mathrm{ID}} \rightarrow \sigma_{\tau_i}^{\mathsf{in}}(\mathsf{e\text{-}auth}_{\mathrm{N}}^{j}) = \mathrm{ID} \rightarrow \bigvee_{\substack{\tau'=\_,\mathrm{TU}_{\mathrm{ID}}(\_,1) \\ \tau'\prec_\tau\tau_i}} \sigma_{\tau'}(\mathsf{b\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}) = \mathsf{n}^{j_i}$$

Since $\mathsf{supi\text{-}tr}_{\mathsf{u}:\tau_2,\tau}^{\mathsf{n}:\tau_1} \rightarrow g(\phi_{\tau_2}^{\mathsf{in}}) = \mathsf{n}^{j_1}$, we know that $\mathsf{supi\text{-}tr}_{\mathsf{u}:\tau_2,\tau}^{\mathsf{n}:\tau_1} \rightarrow \sigma_{\tau_2}(\mathsf{b\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}) = \mathsf{n}^{j_1}$. As we know that $\mathsf{n}^{j_1} \neq \mathsf{n}^{j_i}$, we deduce that $\mathsf{supi\text{-}tr}_{\mathsf{u}:\tau_2,\tau}^{\mathsf{n}:\tau_1} \rightarrow \sigma_{\tau_2}(\mathsf{b\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}) \neq \mathsf{n}^{j_i}$. Moreover using the validity of τ we know that $\sigma_{\tau_i}(\mathsf{b\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}) = \sigma_{\tau_2}(\mathsf{b\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}})$. Therefore:

$$\mathsf{supi\text{-}tr}_{\mathsf{u}:\tau_2,\tau}^{\mathsf{n}:\tau_1} \wedge \mathsf{accept}_{\tau_i}^{\mathrm{ID}} \rightarrow \bigvee_{\substack{\tau'=\_,\mathrm{TU}_{\mathrm{ID}}(\_,1) \\ \tau'\prec_\tau\tau_2}} \sigma_{\tau'}(\mathsf{b\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}) = \mathsf{n}^{j_i}$$

Let $\tau' = \_, \mathrm{TU}_{\mathrm{ID}}(\_,1)$ with $\tau' \prec_\tau \tau_2$. We know that $\sigma_{\tau'}(\mathsf{b\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}) = \mathsf{n}^{j_i}$ implies that $\sigma_{\tau'}(\mathsf{b\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}) \neq \mathsf{fail}$, and therefore $\mathsf{accept}_{\tau'}^{\mathrm{ID}}$ holds:

$$\sigma_{\tau'}(\mathsf{b\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}) = \mathsf{n}^{j_i} \rightarrow \sigma_{\tau'}(\mathsf{b\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}) \neq \mathsf{fail} \rightarrow \mathsf{accept}_{\tau'}^{\mathrm{ID}}$$

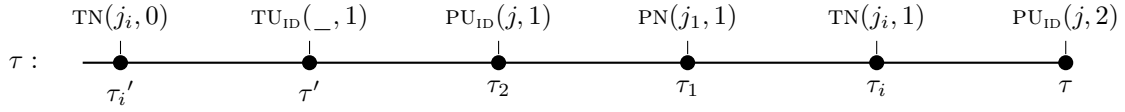
By applying **(Acc3)** we know that:

$$\mathsf{accept}_{\tau'}^{\mathrm{ID}} \rightarrow \bigvee_{\tau_i'=\_,\mathrm{TN}(j_i',0)\prec_\tau\tau'} \pi_1(g(\phi_{\tau'}^{\mathsf{in}})) = \mathsf{n}^{j_i'}$$

Since $[\mathsf{accept}_{\tau'}^{\mathrm{ID}}]\sigma_{\tau'}(\mathsf{b\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}) = [\mathsf{accept}_{\tau'}^{\mathrm{ID}}]\pi_1(g(\phi_{\tau'}^{\mathsf{in}}))$ we deduce:

$$\sigma_{\tau'}(\mathsf{b\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}) = \mathsf{n}^{j_i} \rightarrow \mathsf{false} \qquad \text{if } \tau' \prec_\tau \mathrm{TN}(j_i,0)$$

Hence if $\tau' \prec_\tau \mathrm{TN}(j_i,0)$ we know that $\neg\left(\mathsf{supi\text{-}tr}_{\mathsf{u}:\tau_2,\tau}^{\mathsf{n}:\tau_1} \wedge \mathsf{accept}_{\tau_i}^{\mathrm{ID}}\right)$, which is what we wanted to show. Therefore let $\tau_i' = \_, \mathrm{TN}(j_i,0)$, and assume $\tau_i' \prec_\tau \tau'$. We summarize graphically this below:



We recall (4.27):

$$\neg\sigma_\tau^{\mathsf{in}}(\mathsf{sync}_{\mathrm{U}}^{\mathrm{ID}}) \wedge \mathsf{supi\text{-}tr}_{\mathsf{u}:\tau_2,\tau}^{\mathsf{n}:\tau_1} \rightarrow \sigma_{\tau_2}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) < \sigma_{\tau_1}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}})$$

Hence, using **(B4)** we know that:

$$\neg\sigma_\tau^{\mathsf{in}}(\mathsf{sync}_{\mathrm{U}}^{\mathrm{ID}}) \wedge \mathsf{supi\text{-}tr}_{\mathsf{u}:\tau_2,\tau}^{\mathsf{n}:\tau_1} \rightarrow \bigvee_{\substack{\tau_2 \preceq \tau_x \preceq \tau_1 \\ \tau_x = \_,\mathrm{TN}(j_x,0) \text{ or } \_,\mathrm{TN}(j_x,1) \text{ or } \_,\mathrm{PN}(j_x,1)}} \sigma_{\tau_1}(\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}}) = \mathsf{n}^{j_x}$$

Since $\mathrm{TN}(j_i,0) \prec_\tau \tau_2$ and $\tau_1 \prec_\tau \mathrm{TN}(j_i,1)$:

$$\neg\sigma_\tau^{\mathsf{in}}(\mathsf{sync}_{\mathrm{U}}^{\mathrm{ID}}) \wedge \mathsf{supi\text{-}tr}_{\mathsf{u}:\tau_2,\tau}^{\mathsf{n}:\tau_1} \rightarrow \sigma_{\tau_1}(\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}}) \neq \mathsf{n}^{j_i}$$

For every $\tau_1 \preceq \tau''$ we have:

$$\sigma_{\tau''}(\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}}) = \begin{cases} \mathsf{if}\ \mathsf{inc\text{-}accept}_{\tau''}^{\mathrm{ID}}\ \mathsf{then}\ \mathsf{n}^{j''}\ \mathsf{else}\ \sigma_{\tau''}^{\mathsf{in}}(\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}}) & \text{if } \tau'' = \_,\mathrm{PN}(j'',1) \\ \mathsf{if}\ \mathsf{accept}_{\tau''}^{\mathrm{ID}}\ \mathsf{then}\ \mathsf{n}^{j''}\ \mathsf{else}\ \sigma_{\tau''}^{\mathsf{in}}(\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}}) & \text{if } \tau'' = \_,\mathrm{TN}(j'',0) \\ \sigma_{\tau''}^{\mathsf{in}}(\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}}) & \text{otherwise} \end{cases}$$

Since $\tau' \not\prec_\tau \mathrm{TN}(j_i,0)$, we know that after having set $\sigma_{\tau''}(\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}})$ to n^{j_1} at τ_1, it can never be set to n^{j_i}. Formally, we show by induction that:

$$\sigma_{\tau_1}(\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}}) \neq \mathsf{n}^{j_i} \rightarrow \sigma_{\tau''}(\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}}) \neq \mathsf{n}^{j_i}$$

We conclude by observing that $\sigma_{\tau_i}^{\mathsf{in}}(\mathsf{session}_{\mathrm{N}}^{\mathrm{ID}}) \neq \mathsf{n}^{j_i} \rightarrow \neg\mathsf{inc\text{-}accept}_{\tau_i}^{\mathrm{ID}}$.

Part 6 To conclude the proof of **(StrEqu4)**, it only remains to show that:

$$\neg\sigma_\tau^{\mathsf{in}}(\mathsf{sync}_{\mathrm{U}}^{\mathrm{ID}}) \wedge \mathsf{supi\text{-}tr}_{\mathsf{u}:\tau_2,\tau}^{\mathsf{n}:\tau_1} \rightarrow \mathsf{inc\text{-}accept}_{\tau_1}^{\mathrm{ID}} \tag{4.31}$$

Since $\mathsf{supi\text{-}tr}_{\mathsf{u}:\tau_2,\tau}^{\mathsf{n}:\tau_1} \rightarrow \mathsf{accept}_{\tau_1}^{\mathrm{ID}}$, and since:

$$\mathsf{accept}_{\tau_1}^{\mathrm{ID}} \wedge \neg\mathsf{inc\text{-}accept}_{\tau_1}^{\mathrm{ID}} \leftrightarrow \sigma_{\tau_1}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) > \sigma_{\tau_2}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})$$

To show that (4.31) holds, it is sufficient to show that:

$$\neg\sigma_\tau^{\mathsf{in}}(\mathsf{sync}_{\mathrm{U}}^{\mathrm{ID}}) \wedge \mathsf{supi\text{-}tr}_{\mathsf{u}:\tau_2,\tau}^{\mathsf{n}:\tau_1} \rightarrow \sigma_{\tau_1}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) \leq \sigma_{\tau_2}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})$$

We generalize this, and show by induction that for every τ_n such that $\tau_2 \preceq \tau_n \prec_\tau \tau_1$, we have:

$$\neg\sigma_\tau^{\mathsf{in}}(\mathsf{sync}_{\mathrm{U}}^{\mathrm{ID}}) \wedge \mathsf{supi\text{-}tr}_{\mathsf{u}:\tau_2,\tau}^{\mathsf{n}:\tau_1} \rightarrow \sigma_{\tau_n}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) \leq \sigma_{\tau_2}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})$$

If $\tau_n = \tau_2$, this is immediate using **(B5)** and the fact that $\sigma_{\tau_n}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) = \sigma_{\tau_n}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}})$. Therefore let $\tau_n >_\tau \tau_2$, and assume by induction that:

$$\neg\sigma_\tau^{\mathsf{in}}(\mathsf{sync}_{\mathrm{U}}^{\mathrm{ID}}) \wedge \mathsf{supi\text{-}tr}_{\mathsf{u}:\tau_2,\tau}^{\mathsf{n}:\tau_1} \rightarrow \sigma_{\tau_n}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) \leq \sigma_{\tau_2}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})$$

We then have three cases:

- If $\tau_n \neq \_,\mathrm{PN}(\_,1)$ and $\tau_n \neq \_,\mathrm{TN}(\_,1)$, we know that $\sigma_{\tau_n}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) = \sigma_{\tau_n}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}})$, and we conclude directly using the induction hypothesis.
- If $\tau_n = \_,\mathrm{PN}(j_n,1)$. Using **(Equ3)** we know that:

$$\sigma_{\tau_n}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) \neq \sigma_{\tau_n}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) \rightarrow \mathsf{accept}_{\tau_n}^{\mathrm{ID}}$$
$$\rightarrow \bigvee_{\substack{\tau_x = \_,\mathrm{PU}_{\mathrm{ID}}(j_x,1) \\ \tau_x \prec_\tau \tau_n}} \underbrace{\begin{pmatrix} g(\phi_{\tau_x}^{\mathsf{in}}) = \mathsf{n}^{j_n} \wedge \pi_1(g(\phi_{\tau_n}^{\mathsf{in}})) = \{\langle \mathrm{ID}, \sigma_{\tau_x}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})\rangle\}_{\mathsf{pk}_{\mathrm{N}}}^{\mathsf{n_e}^{j_n}} \\ \wedge\ \pi_2(g(\phi_{\tau_n}^{\mathsf{in}})) = \mathsf{Mac}_{\mathsf{k_m}}^{1}(\langle\{\langle \mathrm{ID}, \sigma_{\tau_x}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})\rangle\}_{\mathsf{pk}_{\mathrm{N}}}^{\mathsf{n_e}^{j_n}}, g(\phi_{\tau_x}^{\mathsf{in}})\rangle) \end{pmatrix}}_{\theta_{\tau_x}}$$

Since $\tau_n \prec_\tau \tau_1$, we know that $j_n \neq j_1$. Moreover, $\mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \rightarrow g(\phi^{\mathsf{in}}_{\tau_2}) = \mathsf{n}^{j_1}$. By consequence:

$$\mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \rightarrow g(\phi^{\mathsf{in}}_{\tau_2}) \neq \mathsf{n}^{j_n}$$

Which shows that $\neg(\mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \wedge \theta_{\tau_2})$. Hence:

$$\mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \wedge \sigma_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \neq \sigma^{\mathsf{in}}_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \rightarrow \bigvee_{\substack{\tau_x = \_,\mathrm{PU}_{\mathrm{ID}}(j_x,1)\\ \tau_x \prec_\tau \tau_2}} \theta_{\tau_x}$$

Observe that for every $\tau_x = \_, \mathrm{PU}_{\mathrm{ID}}(j_x, 1)$ such that $\tau_x \prec_\tau \tau_2$:

$$\theta_{\tau_x} \rightarrow \sigma_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = \mathsf{if}\ \sigma^{\mathsf{in}}_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \leq \sigma^{\mathsf{in}}_{\tau_x}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\ \mathsf{then}\ \sigma^{\mathsf{in}}_{\tau_x}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\ \mathsf{else}\ \sigma^{\mathsf{in}}_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$$

Using **(B1)**, we know that $\sigma^{\mathsf{in}}_{\tau_x}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) \leq \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$. Therefore we have the inequality:

$$\theta_{\tau_x} \rightarrow \sigma_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \leq \mathsf{if}\ \sigma^{\mathsf{in}}_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \leq \sigma^{\mathsf{in}}_{\tau_x}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\ \mathsf{then}\ \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\ \mathsf{else}\ \sigma^{\mathsf{in}}_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$$

And using the induction hypothesis, we get that:

$$\neg\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \wedge \theta_{\tau_x} \rightarrow \sigma_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \leq \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$$

Hence:

$$\neg\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \wedge \sigma_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \neq \sigma^{\mathsf{in}}_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \rightarrow \sigma_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \leq \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$$

From which we deduce, using the induction hypothesis, that:

$$\neg\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \rightarrow \sigma_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \leq \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$$

- If $\tau_n = \_, \mathrm{TN}(j_n, 1)$. Using **(StrEqu2)**, we know that:

$$\sigma_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \neq \sigma^{\mathsf{in}}_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \rightarrow \mathsf{accept}^{\mathrm{ID}}_{\tau_n} \rightarrow \bigvee_{\substack{\tau_x{}' = \_,\mathrm{TU}_{\mathrm{ID}}(j_x,0)\\ \tau_n{}' = \_,\mathrm{TN}(j_n,0)\\ \tau_x = \_,\mathrm{TU}_{\mathrm{ID}}(j_x,1)\\ \tau_x{}' \prec_\tau \tau_n{}' \prec_\tau \tau_x \prec_\tau \tau_n}} \mathsf{full\text{-}tr}^{\mathsf{n}:\tau_n{}',\tau_n}_{\mathsf{u}:\tau_x{}',\tau_x}$$

Let $\tau_x = \_, \mathrm{TU}_{\mathrm{ID}}(j_x, 1)$, $\tau_n{}' = \_, \mathrm{TN}(j_n, 0)$, $\tau_x{}' = \_, \mathrm{TU}_{\mathrm{ID}}(j_x, 0)$ s.t. $\tau_x{}' \prec_\tau \tau_n{}' \prec_\tau \tau_x \prec_\tau \tau_n$. Then:

$$\mathsf{full\text{-}tr}^{\mathsf{n}:\tau_n{}',\tau_n}_{\mathsf{u}:\tau_x{}',\tau_x} \wedge \mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_n} \rightarrow \bigwedge_{\tau_n{}' \prec_\tau \tau_i \prec_\tau \tau_n} \neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_i} \rightarrow \sigma_{\tau_n{}'}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = \sigma^{\mathsf{in}}_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$$

Moreover, since:

$$\mathsf{full\text{-}tr}^{\mathsf{n}:\tau_n{}',\tau_n}_{\mathsf{u}:\tau_x{}',\tau_x} \wedge \mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_n} \rightarrow \sigma^{\mathsf{in}}_{\tau_x}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma_{\tau_n{}'}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$$

We deduce that:

$$\mathsf{full\text{-}tr}^{\mathsf{n}:\tau_n{}',\tau_n}_{\mathsf{u}:\tau_x{}',\tau_x} \rightarrow \sigma_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = \mathsf{if}\ \mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_n}\ \mathsf{then}\ \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_x}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))\ \mathsf{else}\ \sigma^{\mathsf{in}}_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$$

By validity of τ, we know that $j_x \neq j$ and that $\tau_x \prec_\tau \tau_2$. Therefore using **(B1)** we know that $\sigma_{\tau_x}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) \leq \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$. Moreover $\sigma_{\tau_x}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_x}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))$. Hence:

$$\mathsf{full\text{-}tr}^{\mathsf{n}:\tau_n{}',\tau_n}_{\mathsf{u}:\tau_x{}',\tau_x} \rightarrow \sigma_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \leq \mathsf{if}\ \mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_n}\ \mathsf{then}\ \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\ \mathsf{else}\ \sigma^{\mathsf{in}}_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$$

And using the induction hypothesis, we get that:

$$\neg\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \wedge \mathsf{full\text{-}tr}^{\mathsf{n}:\tau_n{}',\tau_n}_{\mathsf{u}:\tau_x{}',\tau_x} \rightarrow \sigma_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \leq \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$$

Hence:

$$\neg\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \wedge \sigma_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \neq \sigma^{\mathsf{in}}_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \rightarrow \sigma_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \leq \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$$

From which we deduce, using the induction hypothesis, that:

$$\neg\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \rightarrow \sigma_{\tau_n}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \leq \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$$

■

4.11 Unlinkability

In this section, we prove the σ_{ul}-unlinkability of the AKA$^+$ protocol. To do this, we need, for every valid basic action trace τ, to show that there exists a derivation of $\phi_\tau \sim \phi_{\underline{\tau}}$. We show this by induction on τ.

4.11.1 Resistance Against De-Synchronization Attacks

To show that the GUTI protocol is σ_{ul}-unlinkable, we need the protocol the be secure against de-synchronization attacks: for every agent ID, the adversary should not be able to keep ID synchronized in the left protocol, while de-synchronizing $\nu_\tau(\text{ID})$ in the right protocol.

Therefore, we need the range check on the sequence number to hold on the left if and only if the range check holds on the right. More precisely, for every left identity ID and matching right identity $\nu_\tau(\text{ID})$, the result of the range checks should be indistinguishable:

$$\mathsf{range}(\sigma_\tau(\text{SQN}_\text{U}^\text{ID}), \sigma_\tau(\text{SQN}_\text{N}^\text{ID})) \quad \sim \quad \mathsf{range}(\sigma_{\underline{\tau}}(\text{SQN}_\text{U}^{\nu_\tau(\text{ID})}), \sigma_{\underline{\tau}}(\text{SQN}_\text{N}^{\nu_\tau(\text{ID})})) \tag{4.32}$$

Unfortunately, this property is not a invariant of the AKA$^+$ protocol, for two reasons:

- First, knowing that the range checks are indistinguishable after a symbolic execution τ is not enough to show that they are indistinguishable after $\tau_1 = \tau, \mathsf{ai}$ (for some ai). For example, take a model where $\mathsf{range}(u, v)$ is implemented as a check that the difference between u and v lies in some interval:

  $$[\![\mathsf{range}(u, v)]\!] \text{ if and only if } [\![u]\!] - [\![v]\!] \in \{0, \dots, D\}$$

  for some constant $D > 0$, and where suc is an increment by one. Then, a priori, we may have:

  $$[\![\sigma_\tau(\text{SQN}_\text{U}^\text{ID})]\!] - [\![\sigma_\tau(\text{SQN}_\text{N}^\text{ID})]\!] = 0 \in \{0, \dots, D\}$$
  $$[\![\sigma_{\underline{\tau}}(\text{SQN}_\text{U}^{\nu_\tau(\text{ID})})]\!] - [\![\sigma_{\underline{\tau}}(\text{SQN}_\text{N}^{\nu_\tau(\text{ID})})]\!] = D \in \{0, \dots, D\}$$

  While (4.32) holds for τ, it does not hold for $\tau_1 = \tau, \text{PU}_\text{ID}(j, 1)$. Indeed, after executing $\text{PU}_\text{ID}(j, 1)$:

  $$[\![\sigma_{\tau_1}(\text{SQN}_\text{U}^\text{ID})]\!] - [\![\sigma_{\tau_1}(\text{SQN}_\text{N}^\text{ID})]\!] = 1 \in \{0, \dots, D\}$$
  $$[\![\sigma_{\underline{\tau_1}}(\text{SQN}_\text{U}^{\nu_{\tau_1}(\text{ID})})]\!] - [\![\sigma_{\underline{\tau_1}}(\text{SQN}_\text{N}^{\nu_{\tau_1}(\text{ID})})]\!] = D + 1 \notin \{0, \dots, D\}$$

  To avoid this, we require that $\mathsf{range}(\_, \_)$ and $\mathsf{suc}(\_)$ are implemented as, respectively, an equality check and an integer by-one increment. Moreover, we strengthen the induction property to show that the difference between the sequence numbers are indistinguishable, i.e.:

  $$\sigma_\tau(\text{SQN}_\text{U}^\text{ID}) - \sigma_\tau(\text{SQN}_\text{N}^\text{ID}) \quad \sim \quad \sigma_{\underline{\tau}}(\text{SQN}_\text{U}^{\nu_\tau(\text{ID})}) - \sigma_{\underline{\tau}}(\text{SQN}_\text{N}^{\nu_\tau(\text{ID})}) \tag{4.33}$$

- Second, the property in (4.33) does not always hold: after a $\text{NS}_\text{ID}(\_)$ action, the agent ID and the network may be synchronized on the left (if, e.g., the SUPI protocol has just been successfully executed), but $\nu_\tau(\text{ID})$ is not synchronized with the network.

  Even though the property does not hold, there is no σ_{ul}-unlinkability attack. Indeed a desynchronization attack would need the GUTI protocol to succeed on the left and fail on the right. But the GUTI protocol requires that a fresh GUTI has been established between ID (resp. $\nu_\tau(\text{ID})$) and the network. This can only be achieved through a honest execution of the SUPI protocol. As such a execution will re-synchronize the agent and the network sequence numbers *on both side*, there is no attack.

  To model this, we extended, in Section 4.6.4, the state with a new boolean variable, $\mathsf{sync}_\text{U}^\text{ID}$, that records whether there was a successful execution of the SUPI protocol with agent ID since the last reset $\text{NS}_\text{ID}(\_)$. This variable is only here for proof purposes, and is never used in the actual protocol. We can then state the synchronization invariant:

  $$\underbrace{\mathsf{if}\ \sigma_\tau(\mathsf{sync}_\text{U}^\text{ID})\ \mathsf{then}\ \sigma_\tau(\text{SQN}_\text{U}^\text{ID}) - \sigma_\tau(\text{SQN}_\text{N}^\text{ID})\ \mathsf{else\ error}}_{\mathsf{sync\text{-}diff}_\tau^\text{ID}} \sim \underbrace{\mathsf{if}\ \sigma_{\underline{\tau}}(\mathsf{sync}_\text{U}^{\nu_\tau(\text{ID})})\ \mathsf{then}\ \sigma_{\underline{\tau}}(\text{SQN}_\text{U}^{\nu_\tau(\text{ID})}) - \sigma_{\underline{\tau}}(\text{SQN}_\text{N}^{\nu_\tau(\text{ID})})\ \mathsf{else\ error}}_{\mathsf{sync\text{-}diff}_{\underline{\tau}}^{\nu_\tau(\text{ID})}}$$

4.11.2 The Case Term Construction

We give some definitions that are useful to handle sequences of if_then_else_ in terms.

Definition 4.23. Let $L = (i_1, \ldots, i_l)$ be a list of indices, and $(b_i)_{i\in L}$, $(t_i)_{i\in L}$ two list of terms. Then:

$$\underset{i\in L}{\mathsf{case}}((b_i)_{i\in L} : (m_i)_{i\in L}) \equiv \begin{cases} \mathsf{if}\ b_{i_1}\ \mathsf{then}\ m_{i_1}\ \mathsf{else}\ \underset{i\in L_0}{\mathsf{case}}((b_i)_{i\in L_0} : (m_i)_{i\in L_0}) & \text{when } L \neq \emptyset \text{ and } L_0 = (i_2, \ldots, i_l) \\ \mathsf{defaut} & \text{otherwise} \end{cases}$$

We often abuse notation, and write $\underset{i\in L}{\mathsf{case}}(b_i : m_i)$ instead of $\underset{i\in L}{\mathsf{case}}((b_i)_{i\in L} : (m_i)_{i\in L})$.

Proposition 4.18. *Let $L = (i_1, \ldots, i_l)$ be a list of indices, and $(b_i)_{i\in L}$, $(t_i)_{i\in L}$ two list of terms. If $(b_i)_{i\in L}$ is a CS partition, then for any permutation π of $\{1, \ldots, l\}$, if we let $L_\pi = (i_{\pi(1)}, \ldots, i_{\pi l})$ then:*

$$\underset{i\in L}{\mathsf{case}}(b_i : m_i) \;=\; \underset{i\in L_\pi}{\mathsf{case}}(b_i : m_i)$$

In that case, we write $\underset{i\in\{i_1,\ldots,i_l\}}{\mathsf{case}}(b_i : m_i)$ (i.e. we use a set notation instead of list notation).

Proof. The proof is straightforward by induction over $|L|$. ■

If $(b_i)_{i\in L}$ is such that $(\bigvee_{i\in L} b_i) = \mathsf{true}$ then the case where all tests fail and we return defaut never happens. This motivates the introduction of a second definition.

Definition 4.24. Let $L = (i_1, \ldots, i_l)$ be a list of indices with $l \geq 1$, and $(b_i)_{i\in L}$, $(t_i)_{i\in L}$ two list of terms. Then:

$$\underset{i\in L}{\mathsf{s\text{-}case}}((b_i)_{i\in L} : (m_i)_{i\in L}) \;\equiv\; \begin{cases} \mathsf{if}\ b_{i_1}\ \mathsf{then}\ m_{i_1}\ \mathsf{else}\ \underset{i\in L_0}{\mathsf{case}}((b_i)_{i\in L_0} : (m_i)_{i\in L_0}) & \text{if } L_0 = (i_2, \ldots, i_l) \text{ and } l > 1 \\ m_1 & \text{if } l = 1 \end{cases}$$

Proposition 4.19. *For every list of terms $(b_i)_{i\in L}$ and $(t_i)_{i\in L}$, if $(\bigvee_{i\in L} b_i) = \mathsf{true}$ then:*

$$\underset{i\in L}{\mathsf{case}}(b_i : m_i) = \underset{i\in L}{\mathsf{s\text{-}case}}(b_i : m_i)$$

Proof. We omit the proof. ■

4.11.3 Strengthened Induction Hypothesis

We want to prove that for every valid action trace τ, we have a derivation of:

$$\phi_\tau^{\mathrm{AKA}_N^+} \sim \phi_{\underline{\tau}}^{\mathrm{AKA}_{\underline{N}}^+}$$

for some $\underline{N} = C.N$ large enough (more precisely, C must be larger than $|\tau|$). Instead of proving the formula above, we prove that we have a derivation of the stronger formula:

$$\phi_\tau^{\mathrm{AKA}_N^+}, \mathsf{l\text{-}reveal}_\tau^C \sim \phi_{\underline{\tau}}^{\mathrm{AKA}_{\underline{N}}^+}, \mathsf{r\text{-}reveal}_\tau^C$$

where $\mathsf{l\text{-}reveal}_\tau^C$ and $\mathsf{r\text{-}reveal}_\tau^C$ are terms used in the proof by induction on τ. Basically, we anticipate and include in $\mathsf{l\text{-}reveal}_\tau^C$ and $\mathsf{r\text{-}reveal}_\tau^C$ elements that we will need later in the proof. Morally, they contain terms representing information that can be safely leaked to the adversary, either because he already knows it, or because he can learn this information later in the protocol execution.

Definition 4.25. Let $\tau = \tau_0, \mathsf{ai}$ be a valid basic action trace on $\mathcal{S}_{\mathsf{id}}$ and C an integer. Then reveal_τ^C is a list of elements of the form $u \sim v$ containing exactly the elements:

1. All the elements from $\mathsf{reveal}_{\tau_0}^C$.

2. For every identity ID, let:

$$\mathsf{m\text{-}suci}_{\tau}^{\mathrm{ID}} \equiv [\sigma_\tau(\mathsf{valid\text{-}guti}_{\mathrm{U}}^{\mathrm{ID}})]\sigma_\tau(\mathrm{GUTI}_{\mathrm{U}}^{\mathrm{ID}})$$

Then, for every $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$, reveal_τ^C contains the following elements:

$$\sigma_\tau(\mathsf{valid\text{-}guti}_{\mathrm{U}}^{\mathrm{ID}}) \sim \sigma_{\underline{\tau}}(\mathsf{valid\text{-}guti}_{\mathrm{U}}^{\nu_\tau(\mathrm{ID})}) \qquad \mathsf{m\text{-}suci}_{\tau}^{\mathrm{ID}} \sim \mathsf{m\text{-}suci}_{\underline{\tau}}^{\nu_\tau(\mathrm{ID})} \qquad \sigma_\tau(\mathsf{sync}_{\mathrm{U}}^{\mathrm{ID}}) \sim \sigma_{\underline{\tau}}(\mathsf{sync}_{\mathrm{U}}^{\nu_\tau(\mathrm{ID})})$$

$$\mathsf{sync\text{-}diff}_{\tau}^{\mathrm{ID}} \sim \mathsf{sync\text{-}diff}_{\underline{\tau}}^{\nu_\tau(\mathrm{ID})}$$

3. If $\mathsf{ai} \neq \mathrm{NS}\_(\_)$ then for every identity $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$:

$$\sigma_\tau(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) - \sigma_\tau^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) \sim \sigma_{\underline{\tau}}(\mathrm{SQN}_{\mathrm{U}}^{\nu_\tau(\mathrm{ID})}) - \sigma_{\underline{\tau}}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\nu_\tau(\mathrm{ID})})$$

4. If $\mathsf{ai} = \mathrm{TU}_{\mathrm{ID}}(j, 0)$, then:

$$\sigma_\tau(\mathsf{s\text{-}valid\text{-}guti}_{\mathrm{U}}^{\mathrm{ID}}) \sim \sigma_{\underline{\tau}}(\mathsf{s\text{-}valid\text{-}guti}_{\mathrm{U}}^{\nu_\tau(\mathrm{ID})})$$

5. If $\mathsf{ai} = \mathrm{PU}_{\mathrm{ID}}(j, 1)$, then:

$$\{\langle \mathrm{ID}\,,\, \sigma_\tau^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})\rangle\}_{\mathsf{pk}_{\mathrm{N}}}^{\mathsf{n}_{\mathsf{e}}^j} \sim \{\langle \nu_\tau(\mathrm{ID})\,,\, \sigma_\tau^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\nu_\tau(\mathrm{ID})})\rangle\}_{\mathsf{pk}_{\mathrm{N}}}^{\mathsf{n}_{\mathsf{e}}^j}$$

$$\mathsf{Mac}_{\mathsf{k_m}^{\mathrm{ID}}}^1(\langle\{\langle \mathrm{ID}\,,\, \sigma_\tau^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})\rangle\}_{\mathsf{pk}_{\mathrm{N}}}^{\mathsf{n}_{\mathsf{e}}^j}\,,\, g(\phi_\tau^{\mathsf{in}})\rangle) \sim \mathsf{Mac}_{\mathsf{k_m}^{\nu_\tau(\mathrm{ID})}}^1(\langle\{\langle \nu_\tau(\mathrm{ID})\,,\, \sigma_{\underline{\tau}}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\nu_\tau(\mathrm{ID})})\rangle\}_{\mathsf{pk}_{\mathrm{N}}}^{\mathsf{n}_{\mathsf{e}}^j}\,,\, g(\phi_{\underline{\tau}}^{\mathsf{in}})\rangle)$$

6. If $\mathsf{ai} = \mathrm{PU}_{\mathrm{ID}}(\_, 2)$, $\mathrm{TU}_{\mathrm{ID}}(\_, 1)$ or $\mathrm{FU}_{\mathrm{ID}}(\_)$:

$$\sigma_\tau(\mathsf{e\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}) \sim \sigma_{\underline{\tau}}(\mathsf{e\text{-}auth}_{\mathrm{U}}^{\nu_\tau(\mathrm{ID})})$$

7. If $\mathrm{TU}_{\mathrm{ID}}(j, 1)$ then for every $\tau_1 = \_, \mathrm{TN}(j_0, 0)$ such that $\mathrm{TU}_{\mathrm{ID}}(j, 0) \prec_\tau \tau_1$:

$$\mathsf{Mac}_{\mathsf{k_m}^{\mathrm{ID}}}^4(\mathsf{n}^{j_0}) \sim \mathsf{Mac}_{\mathsf{k_m}^{\nu_\tau(\mathrm{ID})}}^4(\mathsf{n}^{j_0})$$

8. If $\mathsf{ai} = \mathrm{PN}(j, 1)$ then for every $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$, for every $\tau_1 = \_, \mathrm{PU}_{\mathrm{ID}}(j_1, 1) \prec \tau$ such that $\tau_1 \not\prec_\tau \mathrm{NS}_{\mathrm{ID}}(\_)$:

$$\mathsf{Mac}_{\mathsf{k_m}^{\mathrm{ID}}}^2(\langle \mathsf{n}^j\,,\, \mathsf{suc}(\sigma_{\tau_1}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}))\rangle) \sim \mathsf{Mac}_{\mathsf{k_m}^{\nu_\tau(\mathrm{ID})}}^2(\langle \mathsf{n}^j\,,\, \mathsf{suc}(\sigma_{\underline{\tau_1}}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\nu_\tau(\mathrm{ID})}))\rangle)$$

9. If $\mathsf{ai} = \mathrm{PN}(j, 1)$ or $\mathsf{ai} = \mathrm{TN}(j, 1)$, for every identity $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$, we let:

$$\begin{aligned}
\mathsf{net\text{-}e\text{-}auth}_\tau(\mathrm{ID}, j) &\equiv \mathsf{eq}(\sigma_\tau(\mathsf{e\text{-}auth}_{\mathrm{N}}^j), \mathrm{ID}) \\
\underline{\mathsf{net\text{-}e\text{-}auth}}_{\underline{\tau}}(\mathrm{ID}, j) &\equiv \bigvee_{\underline{\mathrm{ID}} \in \mathsf{copies\text{-}id}_C(\mathrm{ID})} \mathsf{eq}(\sigma_{\underline{\tau}}(\mathsf{e\text{-}auth}_{\mathrm{N}}^j), \underline{\mathrm{ID}})
\end{aligned}$$

Then we ask that:

$$\mathsf{net\text{-}e\text{-}auth}_\tau(\mathrm{ID}, j) \sim \underline{\mathsf{net\text{-}e\text{-}auth}}_{\underline{\tau}}(\mathrm{ID}, j)$$

10. If $\mathsf{ai} = \mathrm{FN}(j)$ for every identity $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$ we let $\{\underline{\mathrm{ID}}_1, \ldots, \underline{\mathrm{ID}}_{l_{\mathrm{ID}}}\} = \mathsf{copies\text{-}id}_C(\mathrm{ID})$. We define:

$$\begin{aligned}
\mathsf{t\text{-}suci\text{-}}\oplus_\tau(\mathrm{ID}, j) &\equiv \mathrm{GUTI}^j \oplus \mathsf{f}_{\mathsf{k}^{\mathrm{ID}}}^{\mathsf{r}}(\mathsf{n}^j) \\
\underline{\mathsf{t\text{-}suci\text{-}}\oplus}_{\underline{\tau}}(\mathrm{ID}, j) &\equiv \underset{1 \leq i \leq l_{\mathrm{ID}}}{\mathsf{s\text{-}case}}(\mathsf{eq}(\sigma_{\underline{\tau}}(\mathsf{e\text{-}auth}_{\mathrm{N}}^j), \underline{\mathrm{ID}}_i) : \mathrm{GUTI}^j \oplus \mathsf{f}_{\mathsf{k}^{\underline{\mathrm{ID}}_i}}^{\mathsf{r}}(\mathsf{n}^j)) \\
\mathsf{t\text{-}mac}_\tau(\mathrm{ID}, j) &\equiv \mathsf{Mac}_{\mathsf{k_m}^{\mathrm{ID}}}^5(\langle \mathrm{GUTI}^j\,,\, \mathsf{n}^j\rangle) \\
\underline{\mathsf{t\text{-}mac}}_{\underline{\tau}}(\mathrm{ID}, j) &\equiv \underset{1 \leq i \leq l_{\mathrm{ID}}}{\mathsf{s\text{-}case}}(\mathsf{eq}(\sigma_{\underline{\tau}}(\mathsf{e\text{-}auth}_{\mathrm{N}}^j), \underline{\mathrm{ID}}_i) : \mathsf{Mac}_{\mathsf{k_m}^{\underline{\mathrm{ID}}_i}}^5(\langle \mathrm{GUTI}^j\,,\, \mathsf{n}^j\rangle))
\end{aligned}$$

Then we ask that:

$$\begin{aligned}
\mathrm{GUTI}^j &\sim \mathrm{GUTI}^j \\
[\mathsf{net\text{-}e\text{-}auth}_\tau(\mathrm{ID}, j)]\,(\mathsf{t\text{-}suci\text{-}}\oplus_\tau(\mathrm{ID}, j)) &\sim [\underline{\mathsf{net\text{-}e\text{-}auth}}_{\underline{\tau}}(\mathrm{ID}, j)]\left(\underline{\mathsf{t\text{-}suci\text{-}}\oplus}_{\underline{\tau}}(\mathrm{ID}, j)\right) \\
[\mathsf{net\text{-}e\text{-}auth}_\tau(\mathrm{ID}, j)]\,(\mathsf{t\text{-}mac}_\tau(\mathrm{ID}, j)) &\sim [\underline{\mathsf{net\text{-}e\text{-}auth}}_{\underline{\tau}}(\mathrm{ID}, j)]\left(\underline{\mathsf{t\text{-}mac}}_{\underline{\tau}}(\mathrm{ID}, j)\right)
\end{aligned}$$

Let $(u_i \sim v_i)_{i\in I}$ be such that $\mathsf{reveal}^C_\tau = (u_i \sim v_i)_{i\in I}$. Then we let $\mathsf{l\text{-}reveal}^C_\tau = (u_i)_{i\in I}$ be the list of left elements of reveal^C_τ, and $\mathsf{r\text{-}reveal}^C_\tau = (v_i)_{i\in I}$ list of left elements of reveal^C_τ (in the same order).

Lemma 4.15. *Let N be a number of identities, τ a valid basic action trace on N identities, C a number of copies larger than $|\tau|$ and $\underline{N} = C.N$. Then there exists a derivation of:*

$$\phi_\tau^{AKA^+_N}, \mathsf{l\text{-}reveal}^C_\tau \sim \phi_{\underline{\tau}}^{AKA^+_{\underline{N}}}, \mathsf{r\text{-}reveal}^C_\tau$$

Proof. The proof is given in Section 4.12. ∎

Using this lemma, we can prove Theorem 4.1, which we recall below:

Theorem. *The AKA^+ protocol is σ_{ul}-unlinkable for an arbitrary number of agents and sessions when the asymmetric encryption $\{\_\}_\_$ is* IND-CCA$_1$ *secure and f and f' (resp. Mac^1–Mac^5) satisfy jointly the* PRF *assumption.*

Proof. Using Proposition 4.3, we only need to show that for every $\tau \in \mathsf{support}(\mathcal{R}_{\mathsf{ul}})$, there is a derivation of:

$$\phi_\tau^{\mathrm{AKA}^+_N} \sim \phi_{\underline{\tau}}^{\mathrm{AKA}^+_{\underline{N}}} \tag{4.34}$$

Moreover, using Proposition 4.1, we know that for every $\tau \in \mathsf{support}(\mathcal{R}_{\mathsf{ul}})$, τ is a valid action trace. Moreover, τ uses only the identities $\{\mathrm{ID}_1, \ldots, \mathrm{ID}_N\}$, and is by consequence a basic action trace. Therefore, it is sufficient to prove that there exists a derivation of the formula in (4.34) for every valid basic action trace τ. We conclude using the Restr rule and Lemma 4.15:

$$\frac{\phi_\tau^{\mathrm{AKA}^+_N}, \mathsf{l\text{-}reveal}^C_\tau \sim \phi_{\underline{\tau}}^{\mathrm{AKA}^+_{\underline{N}}}, \mathsf{r\text{-}reveal}^C_\tau}{\phi_\tau^{\mathrm{AKA}^+_N} \sim \phi_{\underline{\tau}}^{\mathrm{AKA}^+_{\underline{N}}}}\ \mathsf{Restr}$$

∎

4.12 ⋆ (p. 159) Proof of Lemma 4.15

The proof is by induction over τ. For $\tau = \epsilon$, we just need to check that the elements of item 2 of Definition 4.25 are indistinguishable, which is obvious from the definition of σ_ϵ in Definition 4.4.

We now show the inductive case: let $\tau = \tau_0, \mathsf{ai}$ be a valid basic action trace on $\mathcal{S}_{\mathsf{id}}$, and let $C \geq |\tau|$. From now on, the number of copies C is implicit, and we omit it (except when necessary). We want to build of derivation of:

$$\phi_\tau, \mathsf{l\text{-}reveal}_\tau \sim \phi_{\underline{\tau}}, \mathsf{r\text{-}reveal}_\tau$$

By induction, we assume that there exists a derivation of:

$$\phi_\tau^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi_{\underline{\tau}}^{\mathsf{in}}, \mathsf{r\text{-}reveal}_{\tau_0}$$

The proof is a case disjunction on the value of ai. Before starting, we assume that the following proposition is true (we postpone its proof to the end of this chapter, in Section 4.13).

Proposition 4.20. *For every basic valid action trace $\tau = \_, \mathsf{ai}$ on $\mathcal{S}_{\mathsf{id}}$*

- ***(Der1)** For every identity $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$, for every τ_1 such that $\tau_1 \prec \tau$ and $\tau_1 \not\prec_\tau \mathrm{NS}_{\mathrm{ID}}(\_)$, there exist derivations using only Simp of:*

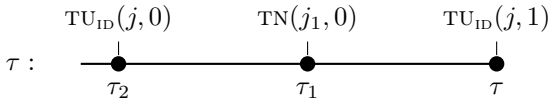
$$\frac{\phi_\tau^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi_{\underline{\tau}}^{\mathsf{in}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\begin{array}{l}\mathsf{l\text{-}reveal}_{\tau_0}, \sigma_\tau^{\mathsf{in}}(\mathrm{sync}_{\mathrm{U}}^{\mathrm{ID}}) \wedge \sigma_\tau^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) < \sigma_{\tau_1}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) \\ \sim\ \mathsf{r\text{-}reveal}_{\tau_0}, \sigma_{\underline{\tau}}^{\mathsf{in}}(\mathrm{sync}_{\mathrm{U}}^{\nu_\tau(\mathrm{ID})}) \wedge \sigma_{\underline{\tau}}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{N}}^{\nu_\tau(\mathrm{ID})}) < \sigma_{\underline{\tau_1}}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\nu_\tau(\mathrm{ID})})\end{array}}\ \mathsf{Simp}$$

$$\frac{\phi_\tau^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi_{\underline{\tau}}^{\mathsf{in}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\begin{array}{l}\mathsf{l\text{-}reveal}_{\tau_0}, \sigma_{\tau_1}^{\mathsf{in}}(\mathrm{sync}_{\mathrm{U}}^{\mathrm{ID}}) \wedge \sigma_{\tau_1}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) < \sigma_\tau^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) \\ \sim\ \mathsf{r\text{-}reveal}_{\tau_0}, \sigma_{\underline{\tau_1}}^{\mathsf{in}}(\mathrm{sync}_{\mathrm{U}}^{\nu_\tau(\mathrm{ID})}) \wedge \sigma_{\underline{\tau_1}}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{N}}^{\nu_\tau(\mathrm{ID})}) < \sigma_{\underline{\tau}}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\nu_\tau(\mathrm{ID})})\end{array}}\ \mathsf{Simp}$$

- ***(Der2)*** *If* $\mathsf{ai} = \mathrm{FU}_{\mathrm{ID}}(j)$. *For every* $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$, *for every* $\tau_1 = \_, \mathrm{FN}(j_0) \prec \tau$ *such that* $\tau_1 \not\prec_\tau \mathrm{NS}_{\mathrm{ID}}(\_)$:
  - *We have* $\underline{\tau_1} = \_, \mathrm{FN}(j_0)$, $\underline{\tau} = \_, \mathrm{FU}_{\nu_\tau(\mathrm{ID})}(j)$, $\underline{\tau_1} \prec_{\underline{\tau}} \underline{\tau}$ *and* $\underline{\tau_1} \not\prec_{\underline{\tau}} \mathrm{NS}_{\nu_\tau(\mathrm{ID})}(\_)$. *Therefore,* $\mathsf{fu\text{-}tr}_{\mathsf{u}:\underline{\tau}}^{\mathsf{n}:\underline{\tau_1}}$ *is well-defined.*
  - *There is a derivation of:*

$$\frac{\phi_\tau^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi_{\underline{\tau}}^{\mathsf{in}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi_\tau^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{fu\text{-}tr}_{\mathsf{u}:\tau}^{\mathsf{n}:\tau_1} \sim \phi_{\underline{\tau}}^{\mathsf{in}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{fu\text{-}tr}_{\mathsf{u}:\underline{\tau}}^{\mathsf{n}:\underline{\tau_1}}}\ \mathit{Simp}$$

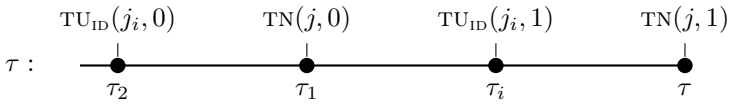
- ***(Der3)*** *If* $\mathsf{ai} = \mathrm{TU}_{\mathrm{ID}}(j,1)$. *For every* $\tau_1 = \_, \mathrm{TN}(j_1,0)$, $\tau_2 = \_, \mathrm{TU}_{\mathrm{ID}}(j,0)$ *such that* $\tau_2 \prec_\tau \tau_1$:



  - *We have* $\underline{\tau_2} = \_, \mathrm{TU}_{\nu_\tau(\mathrm{ID})}(j,0)$, $\underline{\tau_1} = \_, \mathrm{TU}_{\nu_\tau(\mathrm{ID})}(j,1)$ *and* $\underline{\tau_2} \prec_{\underline{\tau}} \underline{\tau_1} \prec_{\underline{\tau}} \underline{\tau}$. *Therefore,* $\mathsf{part\text{-}tr}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}}^{\mathsf{n}:\underline{\tau_1}}$ *is well-defined.*
  - *There is a derivation of:*

$$\frac{\phi_\tau^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi_{\underline{\tau}}^{\mathsf{in}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi_\tau^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{part\text{-}tr}_{\mathsf{u}:\tau_2,\tau}^{\mathsf{n}:\tau_1} \sim \phi_{\underline{\tau}}^{\mathsf{in}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{part\text{-}tr}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}}^{\mathsf{n}:\underline{\tau_1}}}\ \mathit{Simp}$$

- ***(Der4)*** *If* $\mathsf{ai} = \mathrm{TN}(j,1)$. *For every* $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$, $\tau_i = \_, \mathrm{TU}_{\mathrm{ID}}(j_i,1)$, $\tau_1 = \_, \mathrm{TN}(j,0)$, $\tau_2 = \_, \mathrm{TU}_{\mathrm{ID}}(j_i,0)$ *such that* $\tau_2 \prec_\tau \tau_1 \prec_\tau \tau_i$:



  - *We have* $\underline{\tau_2} = \_, \mathrm{TU}_{\nu_{\tau_1}(\mathrm{ID})}(j_i,0)$, $\underline{\tau_i} = \_, \mathrm{TU}_{\nu_{\tau_1}(\mathrm{ID})}(j_i,1)$ *and* $\underline{\tau_2} \prec_{\underline{\tau}} \underline{\tau_1} \prec_{\underline{\tau}} \underline{\tau_i} \prec_{\underline{\tau}} \underline{\tau}$. *Therefore,* $\mathsf{full\text{-}tr}_{\mathsf{u}:\underline{\tau_2},\underline{\tau_i}}^{\mathsf{n}:\underline{\tau_1},\underline{\tau}}$ *is well-defined.*
  - *There is a derivation of:*

$$\frac{\phi_\tau^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi_{\underline{\tau}}^{\mathsf{in}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi_\tau^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{full\text{-}tr}_{\mathsf{u}:\tau_2,\tau_i}^{\mathsf{n}:\tau_1,\tau} \sim \phi_{\underline{\tau}}^{\mathsf{in}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{full\text{-}tr}_{\mathsf{u}:\underline{\tau_2},\underline{\tau_i}}^{\mathsf{n}:\underline{\tau_1},\underline{\tau}}}\ \mathit{Simp}$$

Proof. The proof is given in Section 4.13 ■

We now proceed with the proof of Lemma 4.15. Let $\underline{\mathsf{ai}}$ be such that $\underline{\tau} = \_, \underline{\mathsf{ai}}$.

4.12.1 Case $\mathsf{ai} = \mathrm{NS}_{\mathrm{ID}}(j)$

We know that $\underline{\mathsf{ai}} = \mathrm{NS}_{\nu_{\underline{\tau}}(\mathrm{ID})}(j)$ and $\nu_\tau(\mathrm{ID}) = \mathsf{fresh\text{-}id}(\nu_{\tau_0}(\mathrm{ID}))$. Moreover, $\phi_\tau \equiv \phi_\tau^{\mathsf{in}}$ and $\phi_{\underline{\tau}} \equiv \phi_{\underline{\tau}}^{\mathsf{in}}$. Hence $\mathsf{l\text{-}reveal}_\tau$ and $\mathsf{l\text{-}reveal}_{\tau_0}$ coincide everywhere except on:

$$\sigma_\tau(\mathsf{valid\text{-}guti}_{\mathrm{U}}^{\mathrm{ID}}) \sim \sigma_{\underline{\tau}}(\mathsf{valid\text{-}guti}_{\mathrm{U}}^{\nu_\tau(\mathrm{ID})}) \qquad \mathsf{sync\text{-}diff}_\tau^{\mathrm{ID}} \sim \mathsf{sync\text{-}diff}_{\underline{\tau}}^{\nu_\tau(\mathrm{ID})} \qquad \mathsf{m\text{-}suci}_{\ \tau}^{\ \mathrm{ID}} \sim \mathsf{m\text{-}suci}_{\ \underline{\tau}}^{\ \nu_\tau(\mathrm{ID})}$$

We conclude with the following derivation:

$$\frac{\dfrac{\phi_\tau^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi_{\underline{\tau}}^{\mathsf{in}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi_\tau^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{false}, \mathsf{defaut}, \mathsf{false} \sim \phi_{\underline{\tau}}^{\mathsf{in}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{false}, \mathsf{defaut}, \mathsf{false}}\ \mathsf{Simp}}{\begin{array}{c}\phi_\tau^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0}, \sigma_\tau(\mathsf{valid\text{-}guti}_{\mathrm{U}}^{\mathrm{ID}}), \mathsf{m\text{-}suci}_{\ \tau}^{\ \mathrm{ID}}, \mathsf{sync\text{-}diff}_\tau^{\mathrm{ID}} \\ \sim\ \phi_{\underline{\tau}}^{\mathsf{in}}, \mathsf{r\text{-}reveal}_{\tau_0}, \sigma_{\underline{\tau}}(\mathsf{valid\text{-}guti}_{\mathrm{U}}^{\nu_\tau(\mathrm{ID})}), \mathsf{m\text{-}suci}_{\ \underline{\tau}}^{\ \nu_\tau(\mathrm{ID})}, \mathsf{sync\text{-}diff}_{\underline{\tau}}^{\nu_\tau(\mathrm{ID})}\end{array}}\ R$$

4.12.2 Case $\mathsf{ai} = \mathrm{PN}(j, 0)$

We know that $\underline{\mathsf{ai}} = \mathrm{PN}(j, 0)$. Here $\mathsf{l\text{-}reveal}_\tau$ and $\mathsf{l\text{-}reveal}_{\tau_0}$ coincides completely. Using invariant **(A1)** we know that $\mathsf{n}^j \notin \mathsf{st}(\phi^{\mathsf{in}}_\tau)$, and $\mathsf{n}^j \notin \mathsf{st}(\phi_{\underline{\tau_0}})$. Therefore we conclude this case using the axiom Fresh:

$$\frac{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{n}^j \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{n}^j}\ \mathsf{Fresh}$$

4.12.3 Case $\mathsf{ai} = \mathrm{PU}_{\mathrm{ID}}(j, 1)$

We know that $\underline{\mathsf{ai}} = \mathrm{PU}_{\nu_\tau(\mathrm{ID})}(j, 1)$. Here $\mathsf{l\text{-}reveal}_\tau$ and $\mathsf{l\text{-}reveal}_{\tau_0}$ coincides everywhere except on the pairs:

$$\sigma_\tau(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \sim \sigma_{\underline{\tau}}(\mathsf{valid\text{-}guti}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}}) \qquad \mathsf{m\text{-}suci}^{\mathrm{ID}}_{\tau} \sim \mathsf{m\text{-}suci}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}} \qquad \mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} \sim \mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}}$$

$$\sigma_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) - \sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) \sim \sigma_{\underline{\tau}}(\mathrm{SQN}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}}) - \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}})$$

$$\{\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^j_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} \sim \{\langle \nu_\tau(\mathrm{ID})\,,\, \sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}})\rangle\}^{\mathsf{n}^j_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}}$$

$$\mathsf{Mac}^1_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle\{\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^j_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}}\,,\, g(\phi^{\mathsf{in}}_\tau)\rangle) \sim \mathsf{Mac}^1_{\mathsf{k}^{\nu_\tau(\mathrm{ID})}_{\mathsf{m}}}(\langle\{\langle \nu_\tau(\mathrm{ID})\,,\, \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}})\rangle\}^{\mathsf{n}^j_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}}\,,\, g(\phi^{\mathsf{in}}_{\underline{\tau}})\rangle)$$

Part 1 We know that $\sigma_\tau(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \equiv \sigma_{\underline{\tau}}(\mathsf{valid\text{-}guti}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}}) \equiv \mathsf{false}$. We deduce that $\mathsf{m\text{-}suci}^{\mathrm{ID}}_{\tau} = \mathsf{m\text{-}suci}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}} = \mathsf{defaut}$. It follows that:

$$\frac{\dfrac{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{false}, \mathsf{defaut} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{false}, \mathsf{defaut}}\ \mathsf{FA}^*}{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \sigma_\tau(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}), \mathsf{m\text{-}suci}^{\mathrm{ID}}_{\tau} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \sigma_{\underline{\tau}}(\mathsf{valid\text{-}guti}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}}), \mathsf{m\text{-}suci}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}}}\ R \tag{4.35}$$

Part 2 We have:

$$\begin{aligned}
\sigma_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) - \sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) &= \mathsf{suc}(\sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})) - \sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) &&= \mathbf{1}\\
\sigma_{\underline{\tau}}(\mathrm{SQN}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}}) - \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}}) &= \mathsf{suc}(\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}})) - \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}}) &&= \mathbf{1}
\end{aligned}$$

And:

$$\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} = [\sigma_\tau(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})]\,(\sigma_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) - \sigma_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})) = \left[\sigma^{\mathsf{in}}_\tau(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})\right]\left(\mathsf{suc}(\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau_0})\right)$$

Similarly, $\mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}} = \left[\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}})\right]\left(\mathsf{suc}(\mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau_0}})\right)$. Hence:

$$\frac{\dfrac{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_\tau(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}), \mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}}), \mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau_0}}}\ \mathsf{Dup}^*}{\begin{array}{c}\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau}, \quad \sigma_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) - \sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\\ \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}}, \sigma_{\underline{\tau}}(\mathrm{SQN}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}}) - \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}})\end{array}}\ \mathsf{Simp} \tag{4.36}$$

Part 3 Let $s_l \equiv \mathsf{len}(\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle)$. Using the CCA_1 axiom we directly have that:

$$\frac{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, s_l \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, s_l \qquad \dfrac{\overline{\mathsf{len}(\mathrm{ID}) = \mathsf{len}(\nu_\tau(\mathrm{ID}))} \qquad \mathsf{len}(\sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})) = \mathsf{len}(\sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}}))}{\mathsf{len}(\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle) = \mathsf{len}(\langle \nu_\tau(\mathrm{ID})\,,\, \sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}})\rangle)}\ \mathrm{CCA}_1}{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \{\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^j_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \{\langle \nu_\tau(\mathrm{ID})\,,\, \sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}})\rangle\}^{\mathsf{n}^j_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}}} \tag{4.37}$$

Moreover, using Proposition 4.11, we know that:

$$\mathsf{len}(\sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})) = \mathsf{len}(\sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}})) = \mathsf{len}(\mathsf{sqn\text{-}init}^{\mathrm{ID}}_{\mathrm{U}})$$

We deduce that $s_l = \mathsf{len}(\langle \mathrm{ID}, \mathsf{sqn\text{-}init}_{\mathrm{U}}^{\mathrm{ID}} \rangle)$, therefore:

$$\frac{\phi_\tau^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi_{\underline{\tau}}^{\mathsf{in}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi_\tau^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0}, s_l \sim \phi_{\underline{\tau}}^{\mathsf{in}}, \mathsf{r\text{-}reveal}_{\tau_0}, s_l} \qquad \text{and} \qquad \frac{}{\mathsf{len}(\sigma_\tau^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})) = \mathsf{len}(\sigma_\tau^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\nu_\tau(\mathrm{ID})}))}$$

This completes the derivation in (4.37).

Part 4 To conclude, it only remains to deal with the Mac^1 terms. We start by computing $\mathsf{set\text{-}mac}^1_{\mathsf{k}_{\mathsf{m}}^{\mathrm{ID}}}$:

$$\begin{aligned}\mathsf{set\text{-}mac}^1_{\mathsf{k}_{\mathsf{m}}^{\mathrm{ID}}}(\phi_\tau^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0}) \;=\; & \left\{ \langle \{\langle \mathrm{ID}, \sigma_{\tau_1}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})\rangle\}^{\mathsf{n}_{\mathsf{e}}^{j_1}}_{\mathsf{pk}_{\mathrm{N}}}, g(\phi_{\tau_1}^{\mathsf{in}})\rangle \mid \tau_1 = \_, \mathrm{PU}_{\mathrm{ID}}(j_1,1) \prec \tau \right\} \\ \cup\; & \left\{ \langle \pi_1(g(\phi_{\tau_1}^{\mathsf{in}})), \mathsf{n}^{j_1}\rangle \mid \tau_1 = \_, \mathrm{PN}(j_1,1) \prec \tau \right\}\end{aligned}$$

We want to get rid of the second set above: using **(Equ3)**, we know that for every $\tau_1 = \_, \mathrm{PN}(j_1,1) \prec \tau$:

$$\mathsf{accept}_\tau^{\mathrm{ID}} \leftrightarrow \bigvee_{\substack{\tau_2 = \_, \mathrm{PU}_{\mathrm{ID}}(j_2,1) \\ \tau_2 \prec_\tau \tau_1}} \begin{pmatrix} g(\phi_{\tau_2}^{\mathsf{in}}) = \mathsf{n}^j \wedge \pi_1(g(\phi_{\tau_1}^{\mathsf{in}})) = \{\langle \mathrm{ID}, \sigma_{\tau_2}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})\rangle\}^{\mathsf{n}_{\mathsf{e}}^{j_2}}_{\mathsf{pk}_{\mathrm{N}}} \\ \wedge\; \pi_2(g(\phi_{\tau_1}^{\mathsf{in}})) = \mathsf{Mac}^1_{\mathsf{k}_{\mathsf{m}}^{\mathrm{ID}}}(\langle \{\langle \mathrm{ID}, \sigma_{\tau_2}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})\rangle\}^{\mathsf{n}_{\mathsf{e}}^{j_2}}_{\mathsf{pk}_{\mathrm{N}}}, g(\phi_{\tau_2}^{\mathsf{in}})\rangle) \end{pmatrix} \tag{4.38}$$

We let Ψ' be the vector of terms $\phi_\tau^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0}$ where we replaced every occurrence of $\mathsf{accept}_{\tau_1}^{\mathrm{ID}}$ (where $\tau_1 = \_, \mathrm{PN}(j_1,1) \prec \tau$) by the equivalent term from (4.38). We can check that we have:

$$\mathsf{set\text{-}mac}^1_{\mathsf{k}_{\mathsf{m}}^{\mathrm{ID}}}(\Psi') \;=\; \left\{ \langle \{\langle \mathrm{ID}, \sigma_{\tau_1}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})\rangle\}^{\mathsf{n}_{\mathsf{e}}^{j_1}}_{\mathsf{pk}_{\mathrm{N}}}, g(\phi_{\tau_1}^{\mathsf{in}})\rangle \mid \tau_1 = \_, \mathrm{PU}_{\mathrm{ID}}(j_1,1) \prec \tau \right\}$$

For every $\tau_1 = \_, \mathrm{PU}_{\mathrm{ID}}(j_1,1) \prec \tau$, using Proposition 4.11 we know that:

$$\mathsf{len}(\langle \mathrm{ID}, \sigma_\tau^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})\rangle) = \mathsf{len}(\langle \mathrm{ID}, \sigma_{\tau_1}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})\rangle)$$

Moreover, using the axioms in $\mathsf{Ax}_{\mathsf{len}}$ we know that $\mathsf{len}(\langle \mathrm{ID}, \sigma_\tau^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})\rangle) \neq 0$. Therefore, using Proposition 4.10 we get that we have:

$$\{\langle \mathrm{ID}, \sigma_\tau^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})\rangle\}^{\mathsf{n}_{\mathsf{e}}^{j}}_{\mathsf{pk}_{\mathrm{N}}} \neq \{\langle \mathrm{ID}, \sigma_{\tau_1}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})\rangle\}^{\mathsf{n}_{\mathsf{e}}^{j_1}}_{\mathsf{pk}_{\mathrm{N}}}$$

Hence by left injectivity of $\langle \cdot, \_ \rangle$:

$$\langle \{\langle \mathrm{ID}, \sigma_\tau^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})\rangle\}^{\mathsf{n}_{\mathsf{e}}^{j}}_{\mathsf{pk}_{\mathrm{N}}}, g(\phi_\tau^{\mathsf{in}})\rangle \neq \langle \{\langle \mathrm{ID}, \sigma_{\tau_1}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})\rangle\}^{\mathsf{n}_{\mathsf{e}}^{j_1}}_{\mathsf{pk}_{\mathrm{N}}}, g(\phi_{\tau_1}^{\mathsf{in}})\rangle$$

It follows that we can apply the PRF-MAC^1 axiom to replace the following term by a fresh nonce n:

$$\mathsf{Mac}^1_{\mathsf{k}_{\mathsf{m}}^{\mathrm{ID}}}(\langle \{\langle \mathrm{ID}, \sigma_\tau^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})\rangle\}^{\mathsf{n}_{\mathsf{e}}^{j}}_{\mathsf{pk}_{\mathrm{N}}}, g(\phi_\tau^{\mathsf{in}})\rangle)$$

We then rewrite every occurrence of the right-hand side of (4.38) into $\mathsf{accept}_{\tau_1}^{\mathrm{ID}}$:

$$\frac{\begin{array}{c}\phi_\tau^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{n} \;\sim\; \phi_{\underline{\tau}}^{\mathsf{in}}, \mathsf{r\text{-}reveal}_{\tau_0} \mathsf{Mac}^1_{\mathsf{k}_{\mathsf{m}}^{\nu_\tau(\mathrm{ID})}}(\langle \{\langle \nu_\tau(\mathrm{ID}), \sigma_{\underline{\tau}}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\nu_\tau(\mathrm{ID})})\rangle\}^{\mathsf{n}_{\mathsf{e}}^{j}}_{\mathsf{pk}_{\mathrm{N}}}, g(\phi_{\underline{\tau}}^{\mathsf{in}})\rangle) \\ \phi_\tau^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{Mac}^1_{\mathsf{k}_{\mathsf{m}}^{\mathrm{ID}}}(\langle \{\langle \mathrm{ID}, \sigma_\tau^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})\rangle\}^{\mathsf{n}_{\mathsf{e}}^{j}}_{\mathsf{pk}_{\mathrm{N}}}, g(\phi_\tau^{\mathsf{in}})\rangle)\end{array}}{\begin{array}{c}\phi_\tau^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{Mac}^1_{\mathsf{k}_{\mathsf{m}}^{\mathrm{ID}}}(\langle \{\langle \mathrm{ID}, \sigma_\tau^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})\rangle\}^{\mathsf{n}_{\mathsf{e}}^{j}}_{\mathsf{pk}_{\mathrm{N}}}, g(\phi_\tau^{\mathsf{in}})\rangle) \\ \sim\; \phi_{\underline{\tau}}^{\mathsf{in}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{Mac}^1_{\mathsf{k}_{\mathsf{m}}^{\nu_\tau(\mathrm{ID})}}(\langle \{\langle \nu_\tau(\mathrm{ID}), \sigma_{\underline{\tau}}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\nu_\tau(\mathrm{ID})})\rangle\}^{\mathsf{n}_{\mathsf{e}}^{j}}_{\mathsf{pk}_{\mathrm{N}}}, g(\phi_{\underline{\tau}}^{\mathsf{in}})\rangle)\end{array}}\ \text{PRF-MAC}^1$$

We then do the same on the right side (we omit the details), and conclude using Fresh:

$$\frac{\dfrac{\phi_\tau^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0} \;\sim\; \phi_{\underline{\tau}}^{\mathsf{in}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi_\tau^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{n} \;\sim\; \phi_{\underline{\tau}}^{\mathsf{in}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{n}}\ \mathsf{Fresh}}{\phi_\tau^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{n} \;\sim\; \phi_{\underline{\tau}}^{\mathsf{in}}, \mathsf{r\text{-}reveal}_{\tau_0} \mathsf{Mac}^1_{\mathsf{k}_{\mathsf{m}}^{\nu_\tau(\mathrm{ID})}}(\langle \{\langle \nu_\tau(\mathrm{ID}), \sigma_{\underline{\tau}}^{\mathsf{in}}(\mathrm{SQN}_{\mathrm{U}}^{\nu_\tau(\mathrm{ID})})\rangle\}^{\mathsf{n}_{\mathsf{e}}^{j}}_{\mathsf{pk}_{\mathrm{N}}}, g(\phi_{\underline{\tau}}^{\mathsf{in}})\rangle)}\ \text{PRF-MAC}^1$$

We conclude the proof by combining the derivation above with the derivations in (4.35), (4.36) and (4.37), and by using the induction hypothesis.

4.12.4 Case $\mathsf{ai} = \mathrm{PN}(j,1)$

We know that $\underline{\mathsf{ai}} = \mathrm{PN}(j,1)$. For every $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$, let M_{ID} be the set:

$$M_{\mathrm{ID}} \;=\; \{\tau_2 \mid \tau_2 = \_, \mathrm{PU}_{\mathrm{ID}}(j_1,1) \prec \tau \wedge \tau_2 \not\prec_\tau \mathrm{NS}_{\mathrm{ID}}(\_)\}$$

Here $\mathsf{l\text{-}reveal}_\tau$ and $\mathsf{l\text{-}reveal}_{\tau_0}$ coincides everywhere except on the following pairs:

$$\left(\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} \;\sim\; \mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}}\right)_{\mathrm{ID}\in\mathcal{S}_{\mathsf{id}}} \qquad \left(\mathsf{net\text{-}e\text{-}auth}_\tau(\mathrm{ID},j) \;\sim\; \underline{\mathsf{net\text{-}e\text{-}auth}_{\underline{\tau}}}(\mathrm{ID},j)\right)_{\mathrm{ID}\in\mathcal{S}_{\mathsf{id}}}$$

$$\left(\mathsf{Mac}^2_{\mathsf{k_m}^{\mathrm{ID}}}(\langle \mathsf{n}^j\,,\,\mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))\rangle) \;\sim\; \mathsf{Mac}^2_{\mathsf{k_m}^{\nu_\tau(\mathrm{ID})}}(\langle \mathsf{n}^j\,,\,\mathsf{suc}(\sigma^{\mathsf{in}}_{\underline{\tau_2}}(\mathrm{SQN}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}}))\rangle)\right)_{\tau_2\in M_{\mathrm{ID}},\mathrm{ID}\in\mathcal{S}_{\mathsf{id}}}$$

Part 1 Let $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$, we consider all the new sessions started with identity ID in τ:

$$\{\mathrm{NS}_{\mathrm{ID}}(0),\dots,\mathrm{NS}_{\mathrm{ID}}(l_{\mathrm{ID}})\} = \{\mathrm{NS}_{\mathrm{ID}}(i) \mid \mathrm{NS}_{\mathrm{ID}}(i) \in \tau\}$$

This induce a partition of symbolic actions in τ for identity ID. Indeed, let k be such that $\mathrm{ID} = \mathsf{A}_{k,0}$, and for every $-1 \le i \le l_{\mathrm{ID}}$, let $\underline{\mathrm{ID}}_i = \mathsf{A}_{k,i+1}$. Then we define, for every $-1 \le i \le l_{\mathrm{ID}}$:

$$T^i_{\mathrm{ID}} \;=\; \left\{ \tau_1 \mid \tau_1 = \_, \mathrm{PU}_{\mathrm{ID}}(j_1,1) \wedge \begin{cases} \mathrm{NS}_{\mathrm{ID}}(i) \prec_\tau \tau_1 \prec_\tau \mathrm{NS}_{\mathrm{ID}}(i+1) & \text{if } 0 \le i < l_{\mathrm{ID}} \\ \tau_1 \prec_\tau \mathrm{NS}_{\mathrm{ID}}(0) & \text{if } i = -1 \\ \mathrm{NS}_{\mathrm{ID}}(l_{\mathrm{ID}}) \prec_\tau \tau_1 \prec \tau & \text{if } i = l_{\mathrm{ID}} \end{cases} \right\}$$

And $T_{\mathrm{ID}} = \{\tau_1 \mid \tau_1 = \_, \mathrm{PU}_{\mathrm{ID}}(j_1,1) \wedge \tau_1 \prec \tau\}$. We have $T_{\mathrm{ID}} = \biguplus_{-1\le i\le l_{\mathrm{ID}}} T^i_{\mathrm{ID}}$, and for every $-1 \le i \le l_{\mathrm{ID}}$:

$$\forall \tau_1 \in T^i_{\mathrm{ID}},\, \nu_{\tau_1}(\mathrm{ID}) = \underline{\mathrm{ID}}_i \quad \text{and} \quad T^i_{\mathrm{ID}} \;=\; \left\{\tau_1 \mid \underline{\tau_1} = \_, \mathrm{PU}_{\underline{\mathrm{ID}}_i}(j_1,1) \wedge \underline{\tau_1} \prec \underline{\tau_1}\right\}$$

Part 2 Using **(Equ3)** we know that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \;\leftrightarrow\; \bigvee_{\tau_1=\_,\mathrm{PU}_{\mathrm{ID}}(j_1,1)\in T_{\mathrm{ID}}} \underbrace{\begin{pmatrix} g(\phi^{\mathsf{in}}_{\tau_1}) = \mathsf{n}^j \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau})) = \{\langle \mathrm{ID}\,,\,\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j_1}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} \\ \wedge\, \pi_2(g(\phi^{\mathsf{in}}_{\tau})) = \mathsf{Mac}^1_{\mathsf{k_m}^{\mathrm{ID}}}(\langle\{\langle \mathrm{ID}\,,\,\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j_1}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}}\,,\, g(\phi^{\mathsf{in}}_{\tau_1})\rangle) \end{pmatrix}}_{b^{\mathrm{ID}}_{\tau_1}} \tag{4.39}$$

For all $\tau_1 \in T_{\mathrm{ID}}$, we let $b^{\mathrm{ID}}_{\tau_1}$ be the main term of the disjunction above.

Similarly, using **(Equ3)** on $\underline{\tau}$, we have that for every $-1 \le i \le l_{\mathrm{ID}}$:

$$\mathsf{accept}^{\underline{\mathrm{ID}}_i}_{\underline{\tau}} \;\leftrightarrow\; \bigvee_{\tau_1=\_,\mathrm{PU}_{\mathrm{ID}}(j_1,1)\in T^i_{\mathrm{ID}}} \underbrace{\begin{pmatrix} g(\phi^{\mathsf{in}}_{\underline{\tau_1}}) = \mathsf{n}^j \wedge \pi_1(g(\phi^{\mathsf{in}}_{\underline{\tau}})) = \{\langle \underline{\mathrm{ID}}_i\,,\,\sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}^{\underline{\mathrm{ID}}_i}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j_1}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} \\ \wedge\, \pi_2(g(\phi^{\mathsf{in}}_{\underline{\tau}})) = \mathsf{Mac}^1_{\mathsf{k_m}^{\underline{\mathrm{ID}}_i}}(\langle\{\langle \underline{\mathrm{ID}}_i\,,\,\sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}^{\underline{\mathrm{ID}}_i}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j_1}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}}\,,\, g(\phi^{\mathsf{in}}_{\underline{\tau_1}})\rangle) \end{pmatrix}}_{b^{\underline{\mathrm{ID}}_i}_{\underline{\tau_1}}} \tag{4.40}$$

Moreover, if we let $\{\underline{\mathrm{ID}}_{l_{\mathrm{ID}}+1},\dots,\underline{\mathrm{ID}}_m\}$ be such that:

$$\mathsf{copies\text{-}id}_C(\mathrm{ID}) = \{\underline{\mathrm{ID}}_0,\dots,\underline{\mathrm{ID}}_{l_{\mathrm{ID}}}\} \uplus \{\underline{\mathrm{ID}}_{l_{\mathrm{ID}}+1},\dots,\underline{\mathrm{ID}}_m\}$$

Then, for all $i > l_{\mathrm{ID}}$, we have $\mathsf{accept}^{\underline{\mathrm{ID}}_i}_{\underline{\tau}} \leftrightarrow \mathsf{false}$. Therefore, using **(A5)**, we can show that:

$$\underline{\mathsf{net\text{-}e\text{-}auth}}^{\mathrm{ID}}_{\underline{\tau}} \;\leftrightarrow\; \bigvee_{-1\le i\le l} \mathsf{accept}^{\underline{\mathrm{ID}}_i}_{\underline{\tau}} \tag{4.41}$$

Part 3 For every $\tau_1,\tau_2 \in T_{\mathrm{ID}}$ such that $\tau_1 \neq \tau_2$, $\tau_1 = \_, \mathrm{PU}_{\mathrm{ID}}(j_1,1)$ and $\tau_2 = \_, \mathrm{PU}_{\mathrm{ID}}(j_2,1)$, using Proposition 4.10 and 4.11 we can show that:

$$b^{\mathrm{ID}}_{\tau_1} \wedge b^{\mathrm{ID}}_{\tau_2} \;\to\; \{\langle \mathrm{ID}\,,\,\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j_1}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} = \{\langle \mathrm{ID}\,,\,\sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j_2}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} \;\to\; \mathsf{false}$$

Similarly, for every $\tau_1,\tau_2 \in T^{\underline{\mathrm{ID}}_i}_{\mathrm{ID}}$ such that $\tau_1 \neq \tau_2$:

$$b^{\underline{\mathrm{ID}}_i}_{\underline{\tau_1}} \wedge b^{\underline{\mathrm{ID}}_i}_{\underline{\tau_2}} \;\to\; \mathsf{false}$$

Moreover, since for all identities $\mathrm{ID}_1 \neq \mathrm{ID}_2$, we have $\mathsf{eq}(\mathrm{ID}_1, \mathrm{ID}_2) = \mathsf{false}$ we know that:

$$\neg\big(\mathsf{accept}_{\tau}^{\mathrm{ID}_1} \wedge \mathsf{accept}_{\tau}^{\mathrm{ID}_2}\big) \qquad\qquad \neg\big(\mathsf{accept}_{\underline{\tau}}^{\mathrm{ID}_1} \wedge \mathsf{accept}_{\underline{\tau}}^{\mathrm{ID}_2}\big)$$

We deduce that:

$$\Big(\big((b_{\tau_1}^{\mathrm{ID}})_{\tau_1 \in T_{\mathrm{ID}}}\big)_{\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}}, \underbrace{\textstyle\bigwedge_{\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}} \neg \mathsf{accept}_{\tau}^{\mathrm{ID}}}_{b_{\mathsf{unk}}}\Big) \quad \text{and} \quad \Big(\big((\underline{b_{\tau_1}^{\mathrm{ID}_i}})_{\tau_1 \in T_{\mathrm{ID}}^i \wedge -1 \leq i \leq l_{\mathrm{ID}}}\big)_{\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}}, \underbrace{\textstyle\bigwedge_{\mathrm{ID} \in \mathsf{copies\text{-}id}_C(\mathcal{S}_{\mathsf{id}})} \neg \mathsf{accept}_{\tau}^{\mathrm{ID}}}_{\underline{b_{\mathsf{unk}}}}\Big)$$

are CS partitions. Besides, for all $\tau_1 \in T_{\mathrm{ID}}$ we have:

$$\big[b_{\tau_1}^{\mathrm{ID}}\big]\big(t_{\tau} \;=\; \mathsf{Mac}^2_{\mathsf{k}_{\mathsf{m}}^{\mathrm{ID}}}(\langle \mathsf{n}^j\,,\, \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}))\rangle)\big) \qquad \text{and} \qquad [b_{\mathsf{unk}}]\,(t_{\tau} = \mathsf{UnknownId})$$

From Proposition 4.18 we deduce:

$$\begin{aligned} t_{\tau} \;=\; \mathsf{if}\ \neg b_{\mathsf{unk}}\ \mathsf{then}\ \underset{\substack{\tau_1 \in T_{\mathrm{ID}} \\ \mathrm{ID} \in \mathcal{S}_{\mathsf{id}}}}{\mathsf{case}}\ (b_{\tau_1}^{\mathrm{ID}} : \mathsf{Mac}^2_{\mathsf{k}_{\mathsf{m}}^{\mathrm{ID}}}(\langle \mathsf{n}^j\,,\, \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}))\rangle)) \\ \mathsf{else}\ \mathsf{UnknownId} \end{aligned} \tag{4.42}$$

Similarly, for every $-1 \leq i \leq l_{\mathrm{ID}}$, for every $\tau_1 \in T_i^{\mathrm{ID}}$:

$$\big[\underline{b_{\tau_1}^{\mathrm{ID}_i}}\big]\big(t_{\underline{\tau}} \;=\; \mathsf{Mac}^2_{\mathsf{k}_{\mathsf{m}}^{\mathrm{ID}_i}}(\langle \mathsf{n}^j\,,\, \mathsf{suc}(\sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}_i}))\rangle)\big) \qquad \text{and} \qquad \big[\underline{b_{\mathsf{unk}}}\big]\,\big(t_{\underline{\tau}} = \mathsf{UnknownId}\big)$$

Again, from Proposition 4.18 we deduce:

$$\begin{aligned} t_{\underline{\tau}} \;=\; \mathsf{if}\ \neg \underline{b_{\mathsf{unk}}}\ \mathsf{then}\ \underset{\substack{\tau_1 \in T_{\mathrm{ID}}^i \\ -1 \leq i \leq l_{\mathrm{ID}} \\ \mathrm{ID} \in \mathcal{S}_{\mathsf{id}}}}{\mathsf{case}}\ (\underline{b_{\tau_1}^{\mathrm{ID}_i}} : \mathsf{Mac}^2_{\mathsf{k}_{\mathsf{m}}^{\mathrm{ID}_i}}(\langle \mathsf{n}^j\,,\, \mathsf{suc}(\sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}_i}))\rangle)) \\ \mathsf{else}\ \mathsf{UnknownId} \end{aligned}$$

Since $T_{\mathrm{ID}} \;=\; \biguplus_{-1 \leq i \leq l_{\mathrm{ID}}} T_{\mathrm{ID}}^i$, and since $\forall \tau_1 \in T_{\mathrm{ID}}^i$, $\underline{\mathrm{ID}_i} = \nu_{\tau_1}(\mathrm{ID})$, we know that:

$$\begin{aligned} t_{\underline{\tau}} \;=\; \mathsf{if}\ \neg \underline{b_{\mathsf{unk}}}\ \mathsf{then}\ \underset{\substack{\tau_1 \in T_{\mathrm{ID}} \\ \mathrm{ID} \in \mathcal{S}_{\mathsf{id}}}}{\mathsf{case}}\ (\underline{b_{\tau_1}}^{\nu_{\tau_1}(\mathrm{ID})} : \mathsf{Mac}^2_{\mathsf{k}_{\mathsf{m}}^{\nu_{\tau_1}(\mathrm{ID})}}(\langle \mathsf{n}^j\,,\, \mathsf{suc}(\sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}_{\mathrm{U}}^{\nu_{\tau_1}(\mathrm{ID})}))\rangle)) \\ \mathsf{else}\ \mathsf{UnknownId} \end{aligned} \tag{4.43}$$

Part 4 We are going to show that for every $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$, $-1 \leq i \leq l_{\mathrm{ID}}$, and $\tau_1 = \mathrm{PU}_{\mathrm{ID}}(j_1, 1) \in T_{\mathrm{ID}}^i$:

$$\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, b_{\tau_1}^{\mathrm{ID}} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \underline{b_{\tau_1}^{\mathrm{ID}_i}} \tag{4.44}$$

For this, we rewrite $b_{\tau_1}^{\mathrm{ID}}$ and $\underline{b_{\tau_1}^{\mathrm{ID}_i}}$ using, respectively, (4.39) and (4.40). First, remark that the following pairs of terms are in reveal_{τ_0}:

$$(\mathsf{n}^j, \mathsf{n}^j) \qquad \Big(\big\{\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})\rangle\big\}^{\mathsf{n}_{\mathsf{e}}^{j_1}}_{\mathsf{pk}_{\mathrm{N}}}, \big\{\langle \nu_{\tau_1}(\mathrm{ID})\,,\, \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}_{\mathrm{U}}^{\nu_{\tau_1}(\mathrm{ID})})\rangle\big\}^{\mathsf{n}_{\mathsf{e}}^{j_1}}_{\mathsf{pk}_{\mathrm{N}}}\Big)$$

$$\Big(\mathsf{Mac}^1_{\mathsf{k}_{\mathsf{m}}^{\mathrm{ID}}}(\langle\big\{\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})\rangle\big\}^{\mathsf{n}_{\mathsf{e}}^{j_1}}_{\mathsf{pk}_{\mathrm{N}}}, g(\phi^{\mathsf{in}}_{\tau_1})\rangle), \mathsf{Mac}^1_{\mathsf{k}_{\mathsf{m}}^{\nu_{\tau_1}(\mathrm{ID})}}(\langle\big\{\langle \nu_{\tau_1}(\mathrm{ID})\,,\, \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}_{\mathrm{U}}^{\nu_{\tau_1}(\mathrm{ID})})\rangle\big\}^{\mathsf{n}_{\mathsf{e}}^{j_1}}_{\mathsf{pk}_{\mathrm{N}}}, g(\phi^{\mathsf{in}}_{\underline{\tau_1}})\rangle)\Big)$$

Therefore:

$$\frac{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, b_{\tau_1}^{\mathrm{ID}} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \underline{b_{\tau_1}^{\mathrm{ID}_i}}}\ \mathsf{Simp} \tag{4.45}$$

This concludes the proof of (4.44). Combining this with (4.39), (4.40) and (4.41), we have:

$$\frac{\dfrac{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \big(b_{\tau_1}^{\mathrm{ID}}\big)_{\tau_1 \in T_{\mathrm{ID}}^i, -1 \leq i \leq l_{\mathrm{ID}}} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \big(\underline{b_{\tau_1}^{\mathrm{ID}_i}}\big)_{\tau_1 \in T_{\mathrm{ID}}^i, -1 \leq i \leq l_{\mathrm{ID}}}}\ \mathsf{Simp}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{net\text{-}e\text{-}auth}_{\tau}^{\mathrm{ID}} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \underline{\mathsf{net\text{-}e\text{-}auth}_{\underline{\tau}}^{\mathrm{ID}}}}\ \mathsf{Simp} \tag{4.46}$$

And:

$$\dfrac{\dfrac{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \left(b^{\mathrm{ID}}_{\tau_1}\right)_{\tau_1 \in T^i_{\mathrm{ID}}, -1 \leq i \leq l_{\mathrm{ID}}} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \left(b^{\mathrm{ID}_i}_{\underline{\tau_1}}\right)_{\tau_1 \in T^i_{\mathrm{ID}}, -1 \leq i \leq l_{\mathrm{ID}}}}\ \mathsf{Simp}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, b_{\mathsf{unk}} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, b_{\underline{\mathsf{unk}}}}\ \mathsf{Simp} \tag{4.47}$$

We can now prove that $t_\tau \sim t_{\underline{\tau}}$. First we rewrite t_τ and $t_{\underline{\tau}}$ using, respectively, (4.42) and (4.43). Then we split the proof with FA, and combine it with (4.45) and (4.47). This yields:

$$\dfrac{\begin{array}{l}\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \left(\mathsf{Mac}^2_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle \mathsf{n}^j\,,\, \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))\rangle)\right)_{\tau_1 \in T_{\mathrm{ID}}, \mathrm{ID} \in \mathcal{S}_{\mathsf{id}}} \\ \sim\ \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \left(\mathsf{Mac}^2_{\mathsf{k}^{\nu_{\tau_1}(\mathrm{ID})}_{\mathsf{m}}}(\langle \mathsf{n}^j\,,\, \mathsf{suc}(\sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}^{\nu_{\tau_1}(\mathrm{ID})}_{\mathrm{U}}))\rangle)\right)_{\tau_1 \in T_{\mathrm{ID}}, \mathrm{ID} \in \mathcal{S}_{\mathsf{id}}}\end{array}}{\begin{array}{l}\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, b_{\mathsf{unk}}, \left(b^{\mathrm{ID}}_{\tau_1}, \mathsf{Mac}^2_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle \mathsf{n}^j\,,\, \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))\rangle)\right)_{\tau_1 \in T_{\mathrm{ID}}, \mathrm{ID} \in \mathcal{S}_{\mathsf{id}}} \\ \sim\ \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, b_{\underline{\mathsf{unk}}}, \left(b^{\nu_{\tau_1}(\mathrm{ID})}_{\underline{\tau_1}}, \mathsf{Mac}^2_{\mathsf{k}^{\nu_{\tau_1}(\mathrm{ID})}_{\mathsf{m}}}(\langle \mathsf{n}^j\,,\, \mathsf{suc}(\sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}^{\nu_{\tau_1}(\mathrm{ID})}_{\mathrm{U}}))\rangle)\right)_{\tau_1 \in T_{\mathrm{ID}}, \mathrm{ID} \in \mathcal{S}_{\mathsf{id}}}\end{array}}\ \mathsf{Simp} \tag{4.48}$$

Notice that for every $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$, $M_{\mathrm{ID}} = T^{l_{\mathrm{ID}}}_{\mathrm{ID}}$. Therefore the Mac part in $\mathsf{reveal}_\tau \backslash \mathsf{reveal}_{\tau_0}$ appears in the derivation above, i.e.:

$$\begin{array}{rl} & \left(\mathsf{Mac}^2_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle \mathsf{n}^j\,,\, \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))\rangle), \mathsf{Mac}^2_{\mathsf{k}^{\nu_\tau(\mathrm{ID})}_{\mathsf{m}}}(\langle \mathsf{n}^j\,,\, \mathsf{suc}(\sigma^{\mathsf{in}}_{\underline{\tau_2}}(\mathrm{SQN}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}}))\rangle)\right)_{\tau_2 \in M_{\mathrm{ID}}, \mathrm{ID} \in \mathcal{S}_{\mathsf{id}}} \\ \subseteq & \left(\mathsf{Mac}^2_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle \mathsf{n}^j\,,\, \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))\rangle), \mathsf{Mac}^2_{\mathsf{k}^{\nu_{\tau_1}(\mathrm{ID})}_{\mathsf{m}}}(\langle \mathsf{n}^j\,,\, \mathsf{suc}(\sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}^{\nu_{\tau_1}(\mathrm{ID})}_{\mathrm{U}}))\rangle)\right)_{\tau_1 \in T_{\mathrm{ID}}, \mathrm{ID} \in \mathcal{S}_{\mathsf{id}}} \end{array} \tag{4.49}$$

Part 5 Let $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$. Our goal is to apply the PRF-MAC^2 hypothesis to $\mathsf{Mac}^2_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle \mathsf{n}^j\,,\, \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))\rangle)$ simultaneously for every $\tau_1 \in T_{\mathrm{ID}}$ in:

$$\Psi \ \equiv\ \phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \left(\mathsf{Mac}^2_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle \mathsf{n}^j\,,\, \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))\rangle)\right)_{\tau_1 \in T_{\mathrm{ID}}, \mathrm{ID} \in \mathcal{S}_{\mathsf{id}}}$$

Using **(Equ2)** we know that for every $\mathrm{NS}_{\mathrm{ID}}(l_{\mathrm{ID}}) \prec_\tau \tau_i = \_, \mathrm{PU}_{\mathrm{ID}}(j_i, 2)$:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau_i} \ \leftrightarrow \bigvee_{\substack{\tau_1 = \_, \mathrm{PN}(j_1, 1) \\ \tau_2 = \_, \mathrm{PU}_{\mathrm{ID}}(j_i, 1) \\ \tau_2 \prec_\tau \tau_1 \prec_\tau \tau}} g(\phi^{\mathsf{in}}_{\tau}) = \mathsf{Mac}^2_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle \mathsf{n}^{j_1}\,,\, \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))\rangle) \ \wedge\ g(\phi^{\mathsf{in}}_{\tau_2}) = \mathsf{n}^{j_1} \tag{4.50}$$

Let Ψ' be the formula obtained from Ψ by rewriting every $\mathsf{accept}^{\mathrm{ID}}_{\tau_i}$ s.t. $\mathrm{NS}_{\mathrm{ID}}(l_{\mathrm{ID}}) \prec_\tau \tau_i = \_, \mathrm{PU}_{\mathrm{ID}}(j_i, 2)$ using the equation above. Then we can check that for every $\tau_1 \in T_{\mathrm{ID}}$, there is only one occurrence of $\mathsf{Mac}^2_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle \mathsf{n}^j\,,\, \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))\rangle)$ in Ψ'. Moreover:

$$\begin{array}{rl} \mathsf{set\text{-}mac}^2_{\mathrm{ID}}(\Psi') \backslash \{\langle \mathsf{n}^j\,,\, \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))\rangle\} & = \\ & \{\langle \mathsf{n}^j\,,\, \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))\rangle \mid \tau_2 \in T_{\mathrm{ID}} \wedge \tau_1 \neq \tau_2\} \\ & \cup\ \{\langle \mathsf{n}^{j_0}\,,\, \mathsf{suc}(\pi_2(\mathsf{dec}(\pi_1(g(\phi^{\mathsf{in}}_{\tau_i})), \mathsf{sk}_{\mathrm{N}})))\rangle \mid \tau_i = \_, \mathrm{PN}(j_0, 1) \prec \tau\} \end{array}$$

To apply the PRF-MAC^2 axioms, it is sufficient to show that for every element u in the set above, we have $(\langle \mathsf{n}^j\,,\, \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))\rangle \neq u$:

- Using **(A2)** we know that for every $\tau_1, \tau_2 \in T_{\mathrm{ID}}$, if $\tau_1 \neq \tau_2$ then $\sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})) \neq \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))$. Hence:

$$\langle \mathsf{n}^j\,,\, \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))\rangle \ \neq\ \langle \mathsf{n}^j\,,\, \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))\rangle$$

- for every $\tau_i = \_, \mathrm{PN}(j_0, 1) \prec \tau$, we have $j_0 < j$, hence $\mathsf{n}^{j_0} \neq \mathsf{n}^j$ and by consequence:

$$\langle \mathsf{n}^j\,,\, \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))\rangle \ \neq\ \langle \mathsf{n}^{j_0}\,,\, \mathsf{suc}(\pi_2(\mathsf{dec}(\pi_1(g(\phi^{\mathsf{in}}_{\tau_i})), \mathsf{sk}_{\mathrm{N}})))\rangle$$

We can conclude: we rewrite Ψ into Ψ'; we apply PRF-MAC^2 for every $\tau_1 \in T_{\mathrm{ID}}$, replacing the term $\mathsf{Mac}^2_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle \mathsf{n}^j, \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))\rangle)$ by a fresh nonce n^{j,τ_1}; and we rewrite any term of (4.50) back into $\mathsf{accept}^{\mathrm{ID}}_{\tau_i}$. Doing this for every identity $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$, this yields:

$$\frac{\begin{array}{l}\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \left(\mathsf{n}^{j,\tau_1}\right)_{\tau_1 \in T_{\mathrm{ID}}, \mathrm{ID} \in \mathcal{S}_{\mathsf{id}}} \\ \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \left(\mathsf{Mac}^2_{\mathsf{k}^{\nu_{\tau_1}(\mathrm{ID})}_{\mathsf{m}}}(\langle \mathsf{n}^j, \mathsf{suc}(\sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}^{\nu_{\tau_1}(\mathrm{ID})}_{\mathrm{U}}))\rangle)\right)_{\tau_1 \in T_{\mathrm{ID}}, \mathrm{ID} \in \mathcal{S}_{\mathsf{id}}}\end{array}}{\begin{array}{l}\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \left(\mathsf{Mac}^2_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle \mathsf{n}^j, \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))\rangle)\right)_{\tau_1 \in T_{\mathrm{ID}}, \mathrm{ID} \in \mathcal{S}_{\mathsf{id}}} \\ \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \left(\mathsf{Mac}^2_{\mathsf{k}^{\nu_{\tau_1}(\mathrm{ID})}_{\mathsf{m}}}(\langle \mathsf{n}^j, \mathsf{suc}(\sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}^{\nu_{\tau_1}(\mathrm{ID})}_{\mathrm{U}}))\rangle)\right)_{\tau_1 \in T_{\mathrm{ID}}, \mathrm{ID} \in \mathcal{S}_{\mathsf{id}}}\end{array}}\ (\mathsf{Simp} + \text{PRF-MAC}^2)^*$$

We do the same thing on the Big-side, which yields (we omit the details):

$$\frac{\begin{array}{c}\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0} \ \sim \ \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0} \\ \hline \phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \left(\mathsf{n}^{j,\tau_1}\right)_{\tau_1 \in T_{\mathrm{ID}}, \mathrm{ID} \in \mathcal{S}_{\mathsf{id}}} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \left(\mathsf{n}^{j,\tau_1}\right)_{\tau_1 \in T_{\mathrm{ID}}, \mathrm{ID} \in \mathcal{S}_{\mathsf{id}}}\end{array}\ \mathsf{Fresh}^*}{\begin{array}{l}\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \left(\mathsf{n}^{j,\tau_1}\right)_{\tau_1 \in T_{\mathrm{ID}}, \mathrm{ID} \in \mathcal{S}_{\mathsf{id}}} \\ \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \left(\mathsf{Mac}^2_{\mathsf{k}^{\nu_{\tau_1}(\mathrm{ID})}_{\mathsf{m}}}(\langle \mathsf{n}^j, \mathsf{suc}(\sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}^{\nu_{\tau_1}(\mathrm{ID})}_{\mathrm{U}}))\rangle)\right)_{\tau_1 \in T_{\mathrm{ID}}, \mathrm{ID} \in \mathcal{S}_{\mathsf{id}}}\end{array}}\ (\mathsf{Simp} + \text{PRF-MAC}^2)^*$$

Combining this with (4.48), we get:

$$\frac{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0} \ \sim \ \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, t_{\tau} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, t_{\underline{\tau}}} \tag{4.51}$$

Part 6 We now deal with the $\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} \sim \mathsf{sync\text{-}diff}^{\nu_{\tau}(\mathrm{ID})}_{\underline{\tau}}$ part. We first handle the case where $\sigma^{\mathsf{in}}_{\tau}(\mathrm{sync}^{\mathrm{ID}}_{\mathrm{U}})$ is false. Observe that $\sigma^{\mathsf{in}}_{\tau}(\mathrm{sync}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_0}(\mathrm{sync}^{\mathrm{ID}}_{\mathrm{U}})$, $\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{sync}^{\nu_{\tau}(\mathrm{ID})}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\underline{\tau_0}}(\mathrm{sync}^{\nu_{\tau}(\mathrm{ID})}_{\mathrm{U}})$ and that the pair of terms $(\sigma^{\mathsf{in}}_{\tau_0}(\mathrm{sync}^{\mathrm{ID}}_{\mathrm{U}}), \sigma^{\mathsf{in}}_{\underline{\tau_0}}(\mathrm{sync}^{\nu_{\tau}(\mathrm{ID})}_{\mathrm{U}}))$ appears in reveal_{τ_0}. Moreover:

$$[\neg\sigma^{\mathsf{in}}_{\tau}(\mathrm{sync}^{\mathrm{ID}}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} \ = \ \mathsf{error} \qquad\qquad [\neg\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{sync}^{\nu_{\tau}(\mathrm{ID})}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\nu_{\tau}(\mathrm{ID})}_{\underline{\tau}} \ = \ \mathsf{error}$$

Hence:

$$\frac{\begin{array}{c}\mathsf{l\text{-}reveal}_{\tau_0}, [\sigma^{\mathsf{in}}_{\tau}(\mathrm{sync}^{\mathrm{ID}}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} \ \sim \ \mathsf{r\text{-}reveal}_{\tau_0}, [\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{sync}^{\nu_{\tau}(\mathrm{ID})}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\nu_{\tau}(\mathrm{ID})}_{\underline{\tau}} \\ \hline \mathsf{l\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_{\tau}(\mathrm{sync}^{\mathrm{ID}}_{\mathrm{U}}), \quad [\sigma^{\mathsf{in}}_{\tau}(\mathrm{sync}^{\mathrm{ID}}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau}, \quad [\neg\sigma^{\mathsf{in}}_{\tau}(\mathrm{sync}^{\mathrm{ID}}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau}\end{array}\ \mathsf{Simp}}{\begin{array}{l}\sim \mathsf{r\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{sync}^{\nu_{\tau}(\mathrm{ID})}_{\mathrm{U}}), [\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{sync}^{\nu_{\tau}(\mathrm{ID})}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\nu_{\tau}(\mathrm{ID})}_{\underline{\tau}}, [\neg\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{sync}^{\nu_{\tau}(\mathrm{ID})}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\nu_{\tau}(\mathrm{ID})}_{\underline{\tau}} \\ \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} \ \sim \ \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{sync\text{-}diff}^{\nu_{\tau}(\mathrm{ID})}_{\underline{\tau}}\end{array}\ \mathsf{FA}^*} \tag{4.52}$$

Therefore we can focus on the case where $\sigma^{\mathsf{in}}_{\tau}(\mathrm{sync}^{\mathrm{ID}}_{\mathrm{U}})$ is true. For all $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$, we let:

$$\mathsf{inc\text{-}SQN}^{\mathrm{ID}}_{\tau} \ \equiv \ \pi_2(\mathsf{dec}(\pi_1(g(\phi^{\mathsf{in}}_{\tau})), \mathsf{sk}^{\mathrm{ID}}_{\mathrm{N}})) \geq \sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$$

Then:

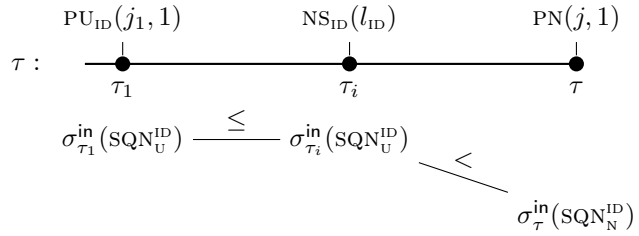
$$[\sigma^{\mathsf{in}}_{\tau}(\mathrm{sync}^{\mathrm{ID}}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} \ = \ \underset{\tau_1 \in T_{\mathrm{ID}}}{\mathsf{case}} \left(b^{\mathrm{ID}}_{\tau_1} : \begin{array}{l}\mathsf{if} \ \left(\begin{array}{r}\sigma^{\mathsf{in}}_{\tau}(\mathrm{sync}^{\mathrm{ID}}_{\mathrm{U}}) \\ \wedge\, \mathsf{inc\text{-}SQN}^{\mathrm{ID}}_{\tau}\end{array}\right) \ \mathsf{then} \ \sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) - \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})) \\ \qquad\qquad \mathsf{else} \ [\sigma^{\mathsf{in}}_{\tau}(\mathrm{sync}^{\mathrm{ID}}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau_0}\end{array} \right) \tag{4.53}$$

And:

$$[\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{sync}^{\nu_{\tau}(\mathrm{ID})}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\nu_{\tau}(\mathrm{ID})}_{\underline{\tau}} \ = \ \underset{\tau_1 \in T^{\mathrm{ID}}_{\mathrm{ID}} l_{\mathrm{ID}}}{\mathsf{case}} \left(\underline{b_{\tau_1}}^{\nu_{\tau}(\mathrm{ID})} : \begin{array}{l}\mathsf{if} \ \left(\begin{array}{r}\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{sync}^{\nu_{\tau}(\mathrm{ID})}_{\mathrm{U}}) \\ \wedge\, \mathsf{inc\text{-}SQN}^{\nu_{\tau}(\mathrm{ID})}_{\underline{\tau}}\end{array}\right) \ \mathsf{then} \ \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\nu_{\tau}(\mathrm{ID})}_{\mathrm{U}}) - \mathsf{suc}(\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\nu_{\tau}(\mathrm{ID})}_{\mathrm{N}})) \\ \qquad\qquad \mathsf{else} \ [\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{sync}^{\nu_{\tau}(\mathrm{ID})}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\nu_{\tau}(\mathrm{ID})}_{\underline{\tau_0}}\end{array} \right) \tag{4.54}$$

Take $\tau_1 \in T_{\mathrm{ID}}$, and let τ_i be such that $\tau_i = \_, \mathrm{NS}_{\mathrm{ID}}(l_{\mathrm{ID}})$ and $\tau_i \prec \tau$. We have two cases:

- If $\tau_1 \prec_\tau \mathrm{NS}_{\mathrm{ID}}(l_{\mathrm{ID}})$, then using **(B1)** and **(B6)**, we know that $\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) \leq \sigma^{\mathsf{in}}_{\tau_i}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$ and that $\sigma^{\mathsf{in}}_{\tau_1}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \rightarrow \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) > \sigma^{\mathsf{in}}_{\tau_i}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$. We summarize this below:



Hence $\neg(b^{\mathrm{ID}}_{\tau_1} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{inc\text{-}SQN}^{\mathrm{ID}}_{\tau})$.
Now we look at the right protocol: since $\tau_1 \prec_\tau \mathrm{NS}_{\mathrm{ID}}(l_{\mathrm{ID}})$, we know that $\nu_{\tau_1}(\mathrm{ID}) = \underline{\mathrm{ID}}_{l_{\mathrm{ID}}-p}$ for some $p > 0$. Hence $\nu_{\tau_1}(\mathrm{ID}) \neq \underline{\mathrm{ID}}_{l_{\mathrm{ID}}} = \nu_\tau(\mathrm{ID})$, which implies that:

$$\underline{b}^{\nu_{\tau_1}(\mathrm{ID})}_{\underline{\tau_1}} \rightarrow \mathsf{accept}^{\nu_{\tau_1}(\mathrm{ID})}_{\underline{\tau}} \rightarrow \neg\mathsf{accept}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}} \rightarrow \bigwedge_{\tau_2 \in T^{l_{\mathrm{ID}}}_{\mathrm{ID}}} \neg \underline{b}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau_2}}$$

We deduce that:

$$\begin{aligned}
[b^{\mathrm{ID}}_{\tau_1} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} &= [b^{\mathrm{ID}}_{\tau_1} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau_0} \\
[\underline{b}^{\nu_{\tau_1}(\mathrm{ID})}_{\underline{\tau_1}} \wedge \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}} &= [\underline{b}^{\nu_{\tau_1}(\mathrm{ID})}_{\underline{\tau_1}} \wedge \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau_0}}
\end{aligned}$$

Since $(\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau_0}, \mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau_0}}) \in \mathsf{reveal}_{\tau_0}$, we have:

$$\dfrac{\dfrac{\mathsf{l\text{-}reveal}_{\tau_0}, b^{\mathrm{ID}}_{\tau_1} \sim \mathsf{r\text{-}reveal}_{\tau_0}, \underline{b}^{\nu_{\tau_1}(\mathrm{ID})}_{\underline{\tau_1}}}{\mathsf{l\text{-}reveal}_{\tau_0}, b^{\mathrm{ID}}_{\tau_1}, \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}), \mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau_0} \sim \mathsf{r\text{-}reveal}_{\tau_0}, \underline{b}^{\nu_{\tau_1}(\mathrm{ID})}_{\underline{\tau_1}}, \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}}), \mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau_0}}}\,\mathsf{Dup}^*}{\mathsf{l\text{-}reveal}_{\tau_0}, [b^{\mathrm{ID}}_{\tau_1} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} \sim \mathsf{r\text{-}reveal}_{\tau_0}, [\underline{b}^{\nu_{\tau_1}(\mathrm{ID})}_{\underline{\tau_1}} \wedge \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}}}\,\mathsf{FA}^*$$

Combining this with (4.45), we can get rid of $b^{\mathrm{ID}}_{\tau_1} \sim \underline{b}^{\nu_{\tau_1}(\mathrm{ID})}_{\underline{\tau_1}}$:

$$\dfrac{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\begin{array}{l}\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, [b^{\mathrm{ID}}_{\tau_1} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} \\ \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, [\underline{b}^{\nu_{\tau_1}(\mathrm{ID})}_{\underline{\tau_1}} \wedge \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}}\end{array}} \tag{4.55}$$

- If $\tau_1 \not\prec_\tau \mathrm{NS}_{\mathrm{ID}}(l_{\mathrm{ID}})$, then $\nu_{\tau_1}(\mathrm{ID}) = \nu_\tau(\mathrm{ID})$. Let $\underline{\mathrm{ID}} = \nu_\tau(\mathrm{ID})$, and using (4.53) and (4.54) we get that:

$$\begin{aligned}
[b^{\mathrm{ID}}_{\tau_1} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} = {} & \left[b^{\mathrm{ID}}_{\tau_1} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})\right]\left(\sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) - \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}))\right) \\
& + \mathsf{if}\ b^{\mathrm{ID}}_{\tau_1} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{inc\text{-}SQN}^{\mathrm{ID}}_{\tau}\ \mathsf{then}\ \mathbf{-1}\ \mathsf{else}\ 0 \\
[\underline{b}^{\underline{\mathrm{ID}}}_{\underline{\tau_1}} \wedge \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\underline{\mathrm{ID}}}_{\underline{\tau}} = {} & \left[\underline{b}^{\underline{\mathrm{ID}}}_{\underline{\tau_1}} \wedge \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}})\right]\left(\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) - \mathsf{suc}(\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{N}}))\right) \\
& + \mathsf{if}\ \underline{b}^{\underline{\mathrm{ID}}}_{\underline{\tau_1}} \wedge \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \mathsf{inc\text{-}SQN}^{\underline{\mathrm{ID}}}_{\underline{\tau}}\ \mathsf{then}\ \mathbf{-1}\ \mathsf{else}\ 0
\end{aligned}$$

Hence using (4.45) we get:

$$\dfrac{\dfrac{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, b^{\mathrm{ID}}_{\tau_1} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{inc\text{-}SQN}^{\mathrm{ID}}_{\tau} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \underline{b}^{\underline{\mathrm{ID}}}_{\underline{\tau_1}} \wedge \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \mathsf{inc\text{-}SQN}^{\underline{\mathrm{ID}}}_{\underline{\tau}}}{\begin{array}{l}\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, b^{\mathrm{ID}}_{\tau_1}, \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}), \sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) - \sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}), b^{\mathrm{ID}}_{\tau_1} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{inc\text{-}SQN}^{\mathrm{ID}}_{\tau} \\ \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \underline{b}^{\underline{\mathrm{ID}}}_{\underline{\tau_1}}, \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}), \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) - \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{N}}), \underline{b}^{\underline{\mathrm{ID}}}_{\underline{\tau_1}} \wedge \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \mathsf{inc\text{-}SQN}^{\underline{\mathrm{ID}}}_{\underline{\tau}}\end{array}}\,\mathsf{Dup}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, [b^{\mathrm{ID}}_{\tau_1} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, [\underline{b}^{\underline{\mathrm{ID}}}_{\underline{\tau_1}} \wedge \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\underline{\mathrm{ID}}}_{\underline{\tau}}}\,\mathsf{FA}^*$$

We split the proof in two, depending on whether $\sigma^{\mathsf{in}}_{\tau_1}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})$ is true or not.

– If it is true, this is simple:

$$\left(\sigma^{\mathsf{in}}_{\tau_1}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge b^{\mathrm{ID}}_{\tau_1} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{inc\text{-}SQN}^{\mathrm{ID}}_{\tau}\right) \leftrightarrow \left(b^{\mathrm{ID}}_{\tau_1} \wedge \sigma^{\mathsf{in}}_{\tau_1}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) < \sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})\right)$$
$$\left(\sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \underline{b}^{\underline{\mathrm{ID}}}_{\underline{\tau_1}} \wedge \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \mathsf{inc\text{-}SQN}^{\underline{\mathrm{ID}}}_{\underline{\tau}}\right) \leftrightarrow \left(\underline{b}^{\underline{\mathrm{ID}}}_{\underline{\tau_1}} \wedge \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) < \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{N}})\right)$$

Hence using (4.45) we get:

$$\frac{\begin{array}{c}\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_{\tau_1}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) < \sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \\ \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) < \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{N}})\end{array}}{\begin{array}{c}\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, b^{\mathrm{ID}}_{\tau_1} \wedge \sigma^{\mathsf{in}}_{\tau_1}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) < \sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \\ \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \underline{b}^{\underline{\mathrm{ID}}}_{\underline{\tau_1}} \wedge \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) < \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{N}})\end{array}}\ \mathsf{Simp}$$

$$\frac{\begin{array}{c}\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, b^{\mathrm{ID}}_{\tau_1} \wedge \sigma^{\mathsf{in}}_{\tau_1}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) < \sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \\ \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \underline{b}^{\underline{\mathrm{ID}}}_{\underline{\tau_1}} \wedge \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) < \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{N}})\end{array}}{\begin{array}{c}\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_{\tau_1}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge b^{\mathrm{ID}}_{\tau_1} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{inc\text{-}SQN}^{\mathrm{ID}}_{\tau} \\ \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \underline{b}^{\underline{\mathrm{ID}}}_{\underline{\tau_1}} \wedge \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \mathsf{inc\text{-}SQN}^{\underline{\mathrm{ID}}}_{\underline{\tau}}\end{array}}\ R$$

We conclude the case $\sigma^{\mathsf{in}}_{\tau_1}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})$ using **(Der1)**:

$$\frac{\mathsf{l\text{-}reveal}_{\tau_0} \sim \mathsf{r\text{-}reveal}_{\tau_0}}{\begin{array}{c}\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_{\tau_1}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) < \sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \\ \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) < \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{N}})\end{array}}\ \mathsf{Simp}$$

– If $\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}$ is false at τ_1 and true at τ, then we know that there is an instant $\tau_1 \preceq \tau_a$ such that $\neg\sigma^{\mathsf{in}}_{\tau_a}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \sigma^{\mathsf{in}}_{\tau_a}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})$. Since $\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}$ is only updated at instant $\mathrm{PU}_{\mathrm{ID}}(\_,\_)$ and $\mathrm{NS}_{\mathrm{ID}}(\_)$, and since $\tau_1 \not\prec_{\tau} \mathrm{NS}_{\mathrm{ID}}(\_)$, the only possibilities are τ_a of the form $\_, \mathrm{PU}_{\mathrm{ID}}(j_a, 2)$. In that case, we must have $\mathsf{accept}^{\mathrm{ID}}_{\tau_a}$. Formally, it is straightforward to show by induction that:

$$b^{\mathrm{ID}}_{\tau_1} \wedge \neg\sigma^{\mathsf{in}}_{\tau_1}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \rightarrow \bigvee_{\substack{\tau_a = \_, \mathrm{PU}_{\mathrm{ID}}(j_a,2) \\ \tau_1 \prec_{\tau} \tau_a}} \neg\sigma^{\mathsf{in}}_{\tau_a}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{accept}^{\mathrm{ID}}_{\tau_a} \tag{4.56}$$

Using **(StrEqu4)**, we know that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau_a} \wedge \neg\sigma^{\mathsf{in}}_{\tau_a}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \rightarrow \sigma_{\tau_a}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma_{\tau_a}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$$

We know that $\sigma_{\tau_a}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_a}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$ and $\sigma_{\tau_a}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = \sigma^{\mathsf{in}}_{\tau_a}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$. Moreover using **(B1)**:

$$\sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) \leq \sigma_{\tau_a}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) \qquad \sigma_{\tau_a}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \leq \sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$$

Finally, we know that $\sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) + 1$, and therefore $\sigma_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) > \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$. We summarize this graphically in Figure 4.24. Therefore:

$$\neg\sigma^{\mathsf{in}}_{\tau_a}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{accept}^{\mathrm{ID}}_{\tau_a} \rightarrow \sigma^{\mathsf{in}}_{\tau_1}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) < \sigma^{\mathsf{in}}_{\tau_a}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{N}})$$

Hence we deduce from (4.56) that:

$$b^{\mathrm{ID}}_{\tau_1} \wedge \neg\sigma^{\mathsf{in}}_{\tau_1}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \rightarrow \mathsf{inc\text{-}SQN}^{\mathrm{ID}}_{\tau}$$

Similarly, we show that:

$$\underline{b}^{\underline{\mathrm{ID}}}_{\underline{\tau_1}} \wedge \neg\sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \rightarrow \mathsf{inc\text{-}SQN}^{\underline{\mathrm{ID}}}_{\underline{\tau}}$$

Hence using (4.45) we get:

$$\frac{\begin{array}{c}\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0} \\ \hline \phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_{\tau_1}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}), b^{\mathrm{ID}}_{\tau_1}, \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}), \underline{b}^{\underline{\mathrm{ID}}}_{\underline{\tau_1}}, \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}})\end{array}}{\begin{array}{c}\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \neg\sigma^{\mathsf{in}}_{\tau_1}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge b^{\mathrm{ID}}_{\tau_1} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \\ \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \neg\sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \underline{b}^{\underline{\mathrm{ID}}}_{\underline{\tau_1}} \wedge \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}})\end{array}}\ \mathsf{Dup}^{*}\ \mathsf{Simp}$$

$$\frac{\begin{array}{c}\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \neg\sigma^{\mathsf{in}}_{\tau_1}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge b^{\mathrm{ID}}_{\tau_1} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \\ \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \neg\sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \underline{b}^{\underline{\mathrm{ID}}}_{\underline{\tau_1}} \wedge \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}})\end{array}}{\begin{array}{c}\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \neg\sigma^{\mathsf{in}}_{\tau_1}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge b^{\mathrm{ID}}_{\tau_1} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{inc\text{-}SQN}^{\mathrm{ID}}_{\tau} \\ \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \neg\sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \underline{b}^{\underline{\mathrm{ID}}}_{\underline{\tau_1}} \wedge \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \mathsf{inc\text{-}SQN}^{\underline{\mathrm{ID}}}_{\underline{\tau}}\end{array}}\ R$$

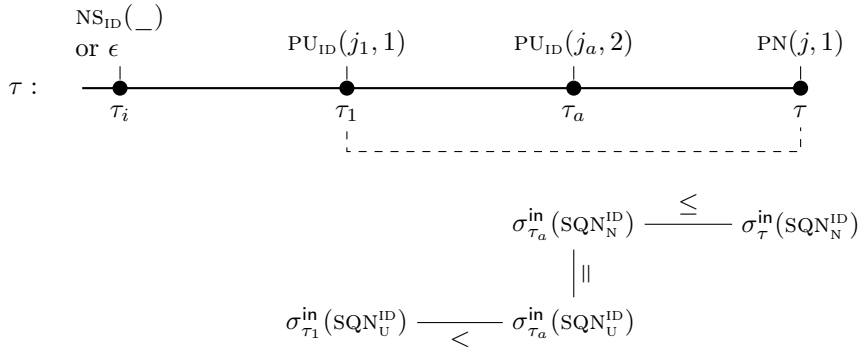


Figure 4.24: Graphical Representation Used in the Proof of the Case $\mathrm{PN}(j,1)$ of Lemma 4.15.

Combining the derivations we build above, we get a derivation of:

$$\frac{\phi^{\mathsf{in}}_{\tau},\mathsf{l\text{-}reveal}_{\tau_0} \;\sim\; \phi^{\mathsf{in}}_{\underline{\tau}},\mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_{\tau},\mathsf{l\text{-}reveal}_{\tau_0},[b^{\mathrm{ID}}_{\tau_1}\wedge\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} \sim \phi^{\mathsf{in}}_{\underline{\tau}},\mathsf{r\text{-}reveal}_{\tau_0},[\underline{b}^{\nu_{\tau_1}(\mathrm{ID})}_{\underline{\tau_1}}\wedge\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\underline{\tau}}} \tag{4.57}$$

Part 7 It only remains to put everything together. First combining (4.45), (4.55) and (4.57), we get:

$$\dfrac{\dfrac{\phi^{\mathsf{in}}_{\tau},\mathsf{l\text{-}reveal}_{\tau_0} \;\sim\; \phi^{\mathsf{in}}_{\underline{\tau}},\mathsf{r\text{-}reveal}_{\tau_0}}{\vdots}}{\begin{array}{c}\phi^{\mathsf{in}}_{\tau},\mathsf{l\text{-}reveal}_{\tau_0},\left(b^{\mathrm{ID}}_{\tau_1},[\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})\wedge b^{\mathrm{ID}}_{\tau_1}]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau}\right)_{\tau_1\in T_{\mathrm{ID}}}\\ \sim\; \phi^{\mathsf{in}}_{\underline{\tau}},\mathsf{r\text{-}reveal}_{\tau_0},\left(\underline{b}^{\nu_{\tau_1}(\mathrm{ID})}_{\underline{\tau_1}},[\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})\wedge\underline{b}^{\nu_{\tau_1}(\mathrm{ID})}_{\underline{\tau_1}}]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\underline{\tau}}\right)_{\tau_1\in T_{\mathrm{ID}}}\end{array}}\,\mathsf{FA}^{*}$$

$$\phi^{\mathsf{in}}_{\tau},\mathsf{l\text{-}reveal}_{\tau_0},[\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} \;\sim\; \phi^{\mathsf{in}}_{\underline{\tau}},\mathsf{r\text{-}reveal}_{\tau_0},[\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\underline{\tau}}$$

Combine with (4.52), this yields:

$$\frac{\phi^{\mathsf{in}}_{\tau},\mathsf{l\text{-}reveal}_{\tau_0} \;\sim\; \phi^{\mathsf{in}}_{\underline{\tau}},\mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_{\tau},\mathsf{l\text{-}reveal}_{\tau_0},\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} \;\sim\; \phi^{\mathsf{in}}_{\underline{\tau}},\mathsf{r\text{-}reveal}_{\tau_0},\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\underline{\tau}}}$$

We conclude the proof of this case by combining this derivation with (4.46) and (4.51) (recall that the Macs in $\mathsf{reveal}_{\tau}\backslash\mathsf{reveal}_{\tau_0}$ were handled in (4.49)).

4.12.5 Case $\mathsf{ai} = \mathrm{PU}_{\mathrm{ID}}(j,2)$

We know that $\underline{\mathsf{ai}} = \mathrm{PU}_{\nu_{\tau}(\mathrm{ID})}(j,2)$. Here $\mathsf{l\text{-}reveal}_{\tau}$ and $\mathsf{l\text{-}reveal}_{\tau_0}$ coincides everywhere except on the pairs:

$$\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} \;\sim\; \mathsf{sync\text{-}diff}^{\nu_{\tau}(\mathrm{ID})}_{\underline{\tau}} \qquad \sigma_{\tau}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) \;\sim\; \sigma_{\underline{\tau}}(\mathsf{e\text{-}auth}^{\nu_{\tau}(\mathrm{ID})}_{\mathrm{U}}) \qquad \sigma_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \;\sim\; \sigma_{\underline{\tau}}(\mathsf{sync}^{\nu_{\tau}(\mathrm{ID})}_{\mathrm{U}})$$

Therefore we are looking for a derivation of:

$$\begin{array}{rl}\Phi \;\equiv & \phi^{\mathsf{in}}_{\tau},\mathsf{l\text{-}reveal}_{\tau_0},\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau},\sigma_{\tau}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}),\sigma_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}),\mathsf{accept}^{\mathrm{ID}}_{\tau}\\ \sim & \phi^{\mathsf{in}}_{\underline{\tau}},\mathsf{r\text{-}reveal}_{\tau_0},\mathsf{sync\text{-}diff}^{\nu_{\tau}(\mathrm{ID})}_{\underline{\tau}},\sigma_{\underline{\tau}}(\mathsf{e\text{-}auth}^{\nu_{\tau}(\mathrm{ID})}_{\mathrm{U}}),\sigma_{\underline{\tau}}(\mathsf{sync}^{\nu_{\tau}(\mathrm{ID})}_{\mathrm{U}}),\mathsf{accept}^{\nu_{\tau}(\mathrm{ID})}_{\underline{\tau}}\end{array} \tag{4.58}$$

Let $\tau_2 = \_,\mathrm{PU}_{\mathrm{ID}}(j,1) \prec \tau$. We know that $\tau_2 \not\prec_{\tau} \mathrm{NS}_{\mathrm{ID}}(\_)$, and therefore $\underline{\tau_2} = \_,\mathrm{PU}_{\nu_{\tau}(\mathrm{ID})}(j,1)$. Also:

$$\sigma^{\mathsf{in}}_{\tau}(\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) \equiv \sigma_{\tau_2}(\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) \equiv g(\phi^{\mathsf{in}}_{\tau_2}) \qquad \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{b\text{-}auth}^{\nu_{\tau}(\mathrm{ID})}_{\mathrm{U}}) \equiv \sigma_{\underline{\tau_2}}(\mathsf{b\text{-}auth}^{\nu_{\tau}(\mathrm{ID})}_{\mathrm{U}}) \equiv g(\phi^{\mathsf{in}}_{\underline{\tau_2}})$$

Hence we can start deconstructing the terms using FA and simplifying with Dup:

$$\dfrac{\dfrac{\phi^{\mathsf{in}}_{\tau},\mathsf{l\text{-}reveal}_{\tau_0},\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau},\mathsf{accept}^{\mathrm{ID}}_{\tau} \;\sim\; \phi^{\mathsf{in}}_{\underline{\tau}},\mathsf{r\text{-}reveal}_{\tau_0},\mathsf{sync\text{-}diff}^{\nu_{\tau}(\mathrm{ID})}_{\underline{\tau}},\mathsf{accept}^{\nu_{\tau}(\mathrm{ID})}_{\underline{\tau}}}{\phi^{\mathsf{in}}_{\tau},\mathsf{l\text{-}reveal}_{\tau_0},\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\underline{\tau}},\mathsf{accept}^{\mathrm{ID}}_{\tau},g(\phi_{\tau_2})}\,\mathsf{Simp}}{\phi^{\mathsf{in}}_{\tau},\mathsf{r\text{-}reveal}_{\tau_0},\mathsf{sync\text{-}diff}^{\nu_{\tau}(\mathrm{ID})}_{\underline{\tau}},\mathsf{accept}^{\nu_{\tau}(\mathrm{ID})}_{\underline{\tau}},g(\phi_{\underline{\tau_2}})}\,\mathsf{Simp}$$

$$\Phi$$

Part 1 We now focus on $\mathsf{accept}^{\mathrm{ID}}_{\tau}$. Let:

$$T = \{\tau_1 \mid \tau_1 = \_, \mathrm{PN}(j_1, 1) \wedge \tau_2 \prec_\tau \tau_1 \prec \tau\}$$

Using **(Equ2)** we know that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \leftrightarrow \bigvee_{\tau_1 = \_, \mathrm{PN}(j_1,1) \in T} \begin{pmatrix} g(\phi^{\mathsf{in}}_{\tau}) = \mathsf{Mac}^2_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle \mathsf{n}^{j_1}, \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))\rangle) \ \wedge \ g(\phi^{\mathsf{in}}_{\tau_2}) = \mathsf{n}^{j_1} \\ \wedge\ \pi_1(g(\phi^{\mathsf{in}}_{\tau_1})) = \{\langle \mathrm{ID}, \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^j_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} \end{pmatrix}$$

Using again **(Equ2)** on $\underline{\tau}$ (which is a valid action trace) we also have:

$$\mathsf{accept}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}} \leftrightarrow \bigvee_{\tau_1 = \_, \mathrm{PN}(j_1,1) \in T} \begin{pmatrix} g(\phi^{\mathsf{in}}_{\underline{\tau}}) = \mathsf{Mac}^2_{\mathsf{k}^{\nu_\tau(\mathrm{ID})}_{\mathsf{m}}}(\langle \mathsf{n}^{j_1}, \mathsf{suc}(\sigma^{\mathsf{in}}_{\underline{\tau_2}}(\mathrm{SQN}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}}))\rangle) \ \wedge \ g(\phi^{\mathsf{in}}_{\underline{\tau_2}}) = \mathsf{n}^{j_1} \\ \wedge\ \pi_1(g(\phi^{\mathsf{in}}_{\underline{\tau_1}})) = \{\langle \mathrm{ID}^{\nu_\tau(\mathrm{ID})}, \sigma^{\mathsf{in}}_{\underline{\tau_2}}(\mathrm{SQN}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}})\rangle\}^{\mathsf{n}^j_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} \end{pmatrix}$$

It is straightforward to check that the formulas above can be decomposed using FA into matching elements of $\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}$. Indeed, for every $\tau_1 = \_, \mathrm{PN}(j_1, 1) \in T$, since $\tau_2 \prec_\tau \tau_1$ and $\tau_2 \not\prec_\tau \mathrm{NS}_{\mathrm{ID}}(\_)$:

$$\left(\mathsf{Mac}^2_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle \mathsf{n}^{j_1}, \mathsf{suc}(\sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}))\rangle), \mathsf{Mac}^2_{\mathsf{k}^{\nu_\tau(\mathrm{ID})}_{\mathsf{m}}}(\langle \mathsf{n}^{j_1}, \mathsf{suc}(\sigma^{\mathsf{in}}_{\underline{\tau_2}}(\mathrm{SQN}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}}))\rangle)\right) \in \mathsf{reveal}_{\tau_0} \qquad \left(\mathsf{n}^{j_1}, \mathsf{n}^{j_1}\right) \in \mathsf{reveal}_{\tau_0}$$

$$\left(\{\langle \mathrm{ID}, \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^j_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}}, \{\langle \mathrm{ID}^{\nu_\tau(\mathrm{ID})}, \sigma^{\mathsf{in}}_{\underline{\tau_2}}(\mathrm{SQN}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}})\rangle\}^{\mathsf{n}^j_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}}\right) \in \mathsf{reveal}_{\tau_0}$$

Hence:

$$\frac{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0} \ \sim \ \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{accept}^{\mathrm{ID}}_{\tau} \ \sim \ \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{accept}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}}} \tag{4.59}$$

Part 2 We focus on $\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau}$. First we get rid of the case where $\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})$ is true. Indeed, we have:

$$[\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} \ = \ [\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})]\mathsf{suc}(\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau_0})$$

$$[\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}} \ = \ [\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}})]\mathsf{suc}(\mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau_0}})$$

And:

$$\left(\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau_0}, \mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau_0}}\right) \in \mathsf{reveal}_{\tau_0} \qquad \left(\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}), \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}})\right) \in \mathsf{reveal}_{\tau_0}$$

Therefore:

$$\frac{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, [\neg\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} \ \sim \ \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, [\neg\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} \ \sim \ \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}}} \ \mathsf{Simp}$$

Similarly:

$$[\neg\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \neg\mathsf{accept}^{\mathrm{ID}}_{\tau}]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} \ = \ \mathsf{error} \qquad [\neg\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}}) \wedge \neg\mathsf{accept}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}}]\mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}} \ = \ \mathsf{error}$$

Hence we can go one step further:

$$\frac{\begin{array}{l}\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{accept}^{\mathrm{ID}}_{\tau}, \quad [\neg\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{accept}^{\mathrm{ID}}_{\tau}]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} \\ \sim \ \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{accept}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}}, [\neg\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}}) \wedge \mathsf{accept}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}}]\mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}}\end{array}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{accept}^{\mathrm{ID}}_{\tau}, \mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} \ \sim \ \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{accept}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}}, \mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}}} \ \mathsf{Simp} \tag{4.60}$$

Part 3 Using **(StrEqu4)** twice, we know that for every $\tau_1 \in T$:

$$\neg\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{accept}^{\mathrm{ID}}_{\tau} \ \rightarrow \ \mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} = 0 \qquad \neg\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}}) \wedge \mathsf{accept}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}} \ \rightarrow \ \mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}} = 0$$

Therefore we can extend the derivation in (4.60):

$$\frac{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{accept}^{\mathrm{ID}}_{\tau} \ \sim \ \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{accept}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{accept}^{\mathrm{ID}}_{\tau}, \mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} \ \sim \ \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{accept}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}}, \ \mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}}} \ \mathsf{Simp}$$

We conclude using the derivation in (4.59) and the induction hypothesis.

4.12.6 Case $\mathsf{ai} = \textsc{fn}(j)$

We know that $\mathsf{ai} = \textsc{fn}(j)$. Here $\mathsf{l\text{-}reveal}_\tau$ and $\mathsf{l\text{-}reveal}_{\tau_0}$ coincides everywhere except on the pairs:

$$\begin{aligned} \textsc{guti}^j &\sim \textsc{guti}^j \\ [\mathsf{net\text{-}e\text{-}auth}_\tau(\textsc{id}, j)]\big(\mathsf{t\text{-}suci\text{-}}\oplus_\tau(\textsc{id}, j)\big) &\sim [\underline{\mathsf{net\text{-}e\text{-}auth}}_{\underline{\tau}}(\textsc{id}, j)]\big(\underline{\mathsf{t\text{-}suci\text{-}}\oplus}_{\underline{\tau}}(\textsc{id}, j)\big) \\ [\mathsf{net\text{-}e\text{-}auth}_\tau(\textsc{id}, j)]\big(\mathsf{t\text{-}mac}_\tau(\textsc{id}, j)\big) &\sim [\underline{\mathsf{net\text{-}e\text{-}auth}}_{\underline{\tau}}(\textsc{id}, j)]\big(\underline{\mathsf{t\text{-}mac}}_{\underline{\tau}}(\textsc{id}, j)\big) \end{aligned}$$

for every identity $\textsc{id} \in \mathcal{S}_{\mathsf{id}}$.

Part 1 Let $\textsc{id} \in \mathcal{S}_{\mathsf{id}}$. Using Lemma 4.7, we know that:

$$\sigma_\tau(\mathsf{e\text{-}auth}^j_{\mathrm{N}}) = \textsc{id} \rightarrow \bigvee_{\tau' \preceq \tau} \sigma_{\tau'}(\mathsf{b\text{-}auth}^{\textsc{id}}_{\mathrm{U}}) = \mathsf{n}^j$$

We check that:

$$t_\tau = \begin{array}{l} \mathsf{if\ net\text{-}e\text{-}auth}_\tau(\mathsf{A}_1, j)\ \mathsf{then} \\ \quad \langle \mathsf{t\text{-}suci\text{-}}\oplus_\tau(\mathsf{A}_1, j)\,,\, \mathsf{t\text{-}mac}_\tau(\mathsf{A}_1, j)\rangle \\ \mathsf{else\ if\ net\text{-}e\text{-}auth}_\tau(\mathsf{A}_2, j)\ \mathsf{then} \\ \quad \langle \mathsf{t\text{-}suci\text{-}}\oplus_\tau(\mathsf{A}_2, j)\,,\, \mathsf{t\text{-}mac}_\tau(\mathsf{A}_2, j)\rangle \\ \quad \cdots \\ \mathsf{else\ UnknownId} \end{array} \qquad t_{\underline{\tau}} = \begin{array}{l} \mathsf{if}\ \underline{\mathsf{net\text{-}e\text{-}auth}}_{\underline{\tau}}(\mathsf{A}_1, j)\ \mathsf{then} \\ \quad \langle \underline{\mathsf{t\text{-}suci\text{-}}\oplus}_{\underline{\tau}}(\mathsf{A}_1, j)\,,\, \underline{\mathsf{t\text{-}mac}}_{\underline{\tau}}(\mathsf{A}_1, j)\rangle \\ \mathsf{else\ if}\ \underline{\mathsf{net\text{-}e\text{-}auth}}_{\underline{\tau}}(\mathsf{A}_2, j)\ \mathsf{then} \\ \quad \langle \underline{\mathsf{t\text{-}suci\text{-}}\oplus}_{\underline{\tau}}(\mathsf{A}_2, j)\,,\, \underline{\mathsf{t\text{-}mac}}_{\underline{\tau}}(\mathsf{A}_2, j)\rangle \\ \quad \cdots \\ \mathsf{else\ UnknownId} \end{array}$$

Using the FA axiom, we split t_τ and $t_{\underline{\tau}}$ as follows:

$$\frac{\begin{array}{r} \big(\mathsf{net\text{-}e\text{-}auth}_\tau(\mathsf{A}_i, j),\, [\mathsf{net\text{-}e\text{-}auth}_\tau(\mathsf{A}_i, j)]\mathsf{t\text{-}suci\text{-}}\oplus_\tau(\mathsf{A}_i, j),\, [\mathsf{net\text{-}e\text{-}auth}_\tau(\mathsf{A}_i, j)]\mathsf{t\text{-}mac}_\tau(\mathsf{A}_i, j)\big)_{i \le B} \\ \sim \big(\underline{\mathsf{net\text{-}e\text{-}auth}}_{\underline{\tau}}(\mathsf{A}_i, j),\, [\underline{\mathsf{net\text{-}e\text{-}auth}}_{\underline{\tau}}(\mathsf{A}_i, j)]\underline{\mathsf{t\text{-}suci\text{-}}\oplus}_{\underline{\tau}}(\mathsf{A}_i, j),\, [\underline{\mathsf{net\text{-}e\text{-}auth}}_{\underline{\tau}}(\mathsf{A}_i, j)]\underline{\mathsf{t\text{-}mac}}_{\underline{\tau}}(\mathsf{A}_i, j)\big)_{i \le B} \end{array}}{t_\tau \sim t_{\underline{\tau}}}\ \mathsf{FA}^*$$

Since:

$$\big(\mathsf{net\text{-}e\text{-}auth}_\tau(\mathsf{A}_i, j), \underline{\mathsf{net\text{-}e\text{-}auth}}_{\underline{\tau}}(\mathsf{A}_i, j)\big) \in \mathsf{reveal}_{\tau_0}$$

We just need to prove that there is a derivation of:

$$\begin{aligned} &\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}\,, \big([\mathsf{net\text{-}e\text{-}auth}_\tau(\mathsf{A}_i, j)]\mathsf{t\text{-}suci\text{-}}\oplus_\tau(\mathsf{A}_i, j),\, [\mathsf{net\text{-}e\text{-}auth}_\tau(\mathsf{A}_i, j)]\mathsf{t\text{-}mac}_\tau(\mathsf{A}_i, j)\big)_{i \le B} \\ \sim\ &\phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \big([\underline{\mathsf{net\text{-}e\text{-}auth}}_{\underline{\tau}}(\mathsf{A}_i, j)]\underline{\mathsf{t\text{-}suci\text{-}}\oplus}_{\underline{\tau}}(\mathsf{A}_i, j),\, [\underline{\mathsf{net\text{-}e\text{-}auth}}_{\underline{\tau}}(\mathsf{A}_i, j)]\underline{\mathsf{t\text{-}mac}}_{\underline{\tau}}(\mathsf{A}_i, j)\big)_{i \le B} \end{aligned}$$

Assume that we have a proof of

$$\begin{aligned} &\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0},\ \big([\mathsf{net\text{-}e\text{-}auth}_\tau(\mathsf{A}_i, j)]\mathsf{t\text{-}suci\text{-}}\oplus_\tau(\mathsf{A}_i, j),\ [\mathsf{net\text{-}e\text{-}auth}_\tau(\mathsf{A}_i, j)]\mathsf{t\text{-}mac}_\tau(\mathsf{A}_i, j)\big)_{i \le B} \\ \sim\ &\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0},\ \big(\mathsf{n}_{i,j},\ \mathsf{n}'_{i,j}\big)_{i \le B} \end{aligned} \tag{4.61}$$

And:

$$\begin{aligned} &\phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0},\ \big(\mathsf{n}_{i,j},\ \mathsf{n}'_{i,j}\big)_{i \le B} \\ \sim\ &\phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0},\ \big([\underline{\mathsf{net\text{-}e\text{-}auth}}_{\underline{\tau}}(\mathsf{A}_i, j)]\underline{\mathsf{t\text{-}suci\text{-}}\oplus}_{\underline{\tau}}(\mathsf{A}_i, j),\ [\underline{\mathsf{net\text{-}e\text{-}auth}}_{\underline{\tau}}(\mathsf{A}_i, j)]\underline{\mathsf{t\text{-}mac}}_{\underline{\tau}}(\mathsf{A}_i, j)\big)_{i \le B} \end{aligned} \tag{4.62}$$

Where for all $\{\mathsf{n}_{i,j}, \mathsf{n}'_{i,j} \mid 1 \le i \le B\}$ are fresh distinct nonces. Since:

$$\frac{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0},\ \big(\mathsf{n}_{i,j},\ \mathsf{n}'_{i,j}\big)_{i \le B} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0},\ \big(\mathsf{n}_{i,j},\ \mathsf{n}'_{i,j}\big)_{i \le B}}\ \mathsf{Fresh}$$

We can conclude using the transitivity axiom Trans and the induction hypothesis.

Part 2 It only remains to give derivations of the formulas in (4.61) and (4.62). We only give the proof for Eq. (4.62) (the derivation of (4.61) is similar).

Instead of doing the proof simultaneously for all i in $\{1,\dots,B\}$, we give the proof for a single i. We let the reader check that the syntactic side-conditions necessary for the derivations for i and i', with $i \neq i'$, are compatible. Therefore the derivations can be sequentially composed, which yield the full proof.

Let $1 \leq i \leq B$. By transitivity, we only have to show that:

$$\phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0},\ \mathsf{n}_{i,j},\ \mathsf{n}'_{i,j} \ \sim\ \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0},\ \mathsf{n}_{i,j},\ [\underline{\mathsf{net\text{-}e\text{-}auth}_{\tau}}(\mathsf{A}_i,j)]\underline{\mathsf{t\text{-}mac}_{\tau}}(\mathsf{A}_i,j) \tag{4.63}$$

And:

$$\begin{aligned} &\phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0},\ \mathsf{n}_{i,j},\ [\underline{\mathsf{net\text{-}e\text{-}auth}_{\tau}}(\mathsf{A}_i,j)]\underline{\mathsf{t\text{-}mac}_{\tau}}(\mathsf{A}_i,j) \\ \sim\ &\phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0},\ [\underline{\mathsf{net\text{-}e\text{-}auth}_{\tau}}(\mathsf{A}_i,j)]\underline{\mathsf{t\text{-}suci\text{-}}\oplus_{\tau}}(\mathsf{A}_i,j),\ [\underline{\mathsf{net\text{-}e\text{-}auth}_{\tau}}(\mathsf{A}_i,j)]\underline{\mathsf{t\text{-}mac}_{\tau}}(\mathsf{A}_i,j) \end{aligned} \tag{4.64}$$

Derivation of (4.64) Let $\{\underline{\mathrm{ID}}_1,\dots,\underline{\mathrm{ID}}_l\} = \mathsf{copies\text{-}id}_C(\mathrm{ID}_i)$. We define, for every $0 \leq y \leq l$, the partially randomized terms $\underline{\mathsf{t\text{-}suci\text{-}}\oplus^y_{\tau}}(\mathrm{ID}_i,j)$:

$$\begin{aligned} \underline{\mathsf{t\text{-}suci\text{-}}\oplus^y_{\tau}}(\mathrm{ID}_i,j) \ \equiv\ &\mathsf{if}\ \mathsf{eq}(\sigma_{\underline{\tau}}(\mathsf{e\text{-}auth}^j_{\mathrm{N}}),\underline{\mathrm{ID}}_1)\ \mathsf{then}\ \mathsf{n}^1_{i,j} \\ &\dots \\ &\mathsf{else\ if}\ \mathsf{eq}(\sigma_{\underline{\tau}}(\mathsf{e\text{-}auth}^j_{\mathrm{N}}),\underline{\mathrm{ID}}_{y-1})\ \mathsf{then}\ \mathsf{n}^{y-1}_{i,j} \\ &\mathsf{else\ if}\ \mathsf{eq}(\sigma_{\underline{\tau}}(\mathsf{e\text{-}auth}^j_{\mathrm{N}}),\underline{\mathrm{ID}}_{y})\ \mathsf{then}\ \mathrm{GUTI}^j \oplus \mathsf{f}^{\mathsf{r}}_{\mathsf{k}^{\underline{\mathrm{ID}}_y}}(\mathsf{n}^j) \\ &\dots \\ &\mathsf{else}\ \mathrm{GUTI}^j \oplus \mathsf{f}^{\mathsf{r}}_{\mathsf{k}^{\underline{\mathrm{ID}}_l}}(\mathsf{n}^j) \end{aligned}$$

Remark that:

$$[\underline{\mathsf{net\text{-}e\text{-}auth}_{\tau}}(\mathsf{A}_i,j)]\underline{\mathsf{t\text{-}suci\text{-}}\oplus^0_{\tau}}(\mathrm{ID}_i,j) \ =\ [\underline{\mathsf{net\text{-}e\text{-}auth}_{\tau}}(\mathsf{A}_i,j)]\underline{\mathsf{t\text{-}suci\text{-}}\oplus_{\tau}}(\mathsf{A}_i,j)$$

And that:

$$\frac{}{\begin{aligned} &\phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0},\ \mathsf{n}_{i,j}, && [\underline{\mathsf{net\text{-}e\text{-}auth}_{\tau}}(\mathsf{A}_i,j)]\underline{\mathsf{t\text{-}mac}_{\tau}}(\mathsf{A}_i,j) \\ \sim\ &\phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0},\ [\underline{\mathsf{net\text{-}e\text{-}auth}_{\tau}}(\mathsf{A}_i,j)]\underline{\mathsf{t\text{-}suci\text{-}}\oplus^l_{\tau}}(\mathrm{ID}_i,j), && [\underline{\mathsf{net\text{-}e\text{-}auth}_{\tau}}(\mathsf{A}_i,j)]\underline{\mathsf{t\text{-}mac}_{\tau}}(\mathsf{A}_i,j) \end{aligned}}\ \mathsf{indep\text{-}branch}$$

Hence by transitivity, to prove that there exists a derivation of Formula (4.64) it is sufficient to prove that, for every $0 < y \leq l$, that we have a derivation of $\phi_{y-1} \sim \phi_y$, where:

$$\begin{aligned} \phi_{y-1} \ &\equiv\ \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0},\ [\underline{\mathsf{net\text{-}e\text{-}auth}_{\tau}}(\mathsf{A}_i,j)]\underline{\mathsf{t\text{-}suci\text{-}}\oplus^{y-1}_{\tau}}(\mathrm{ID}_i,j),\ [\underline{\mathsf{net\text{-}e\text{-}auth}_{\tau}}(\mathsf{A}_i,j)]\underline{\mathsf{t\text{-}mac}_{\tau}}(\mathsf{A}_i,j) \\ \phi_{y} \ &\equiv\ \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0},\ [\underline{\mathsf{net\text{-}e\text{-}auth}_{\tau}}(\mathsf{A}_i,j)]\underline{\mathsf{t\text{-}suci\text{-}}\oplus^{y}_{\tau}}(\mathrm{ID}_i,j),\ [\underline{\mathsf{net\text{-}e\text{-}auth}_{\tau}}(\mathsf{A}_i,j)]\underline{\mathsf{t\text{-}mac}_{\tau}}(\mathsf{A}_i,j) \end{aligned}$$

Let $1 \leq y \leq B$, we are going to give a derivation of $\phi_{y-1} \sim \phi_y$. This is done in two times:

- First, we are going to use the PRF-f^{r} axiom applied to f^{r}, with key $\mathsf{k}^{\underline{\mathrm{ID}}_y}$, to replace $\mathrm{GUTI}^j \oplus \mathsf{f}^{\mathsf{r}}_{\mathsf{k}^{\underline{\mathrm{ID}}_y}}(\mathsf{n}^j)$ with $\mathrm{GUTI}^j \oplus \mathsf{n}''^y_{i,j}$ (where $\mathsf{n}''^y_{i,j}$ is a fresh nonce).

  Observe that there is only one occurrence of $\mathsf{f}^{\mathsf{r}}_{\mathsf{k}^{\underline{\mathrm{ID}}_y}}(\mathsf{n}^j)$ in ϕ_{y-1} (and none in ϕ_y). Moreover:

$$\begin{aligned} \mathsf{set\text{-}prf}^{\mathsf{f}^{\mathsf{r}}}_{\mathsf{k}^{\underline{\mathrm{ID}}_y}}(\phi_{y-1},\phi_y)\setminus\{\mathsf{n}^j\} \ =\ &\left\{\sigma^{\mathsf{in}}_{\tau_1}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) \mid \tau_1 = \_, \mathrm{FU}_{\underline{\mathrm{ID}}_y}(p) \prec \tau\right\} \\ &\cup\ \{\mathsf{n}^p \mid \tau_1 = \_, \mathrm{FN}(p) \prec \tau\} \end{aligned}$$

  Let $\tau_1 = \_, \mathrm{FN}(p) \prec \tau$. We know that $p \neq j$, and therefore that $\neg(\mathsf{n}^p = \mathsf{n}^j)$.We still need guards for $\sigma^{\mathsf{in}}_{\tau_1}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^j$, for every $\tau_1 = \_, \mathrm{FU}_{\underline{\mathrm{ID}}_y}(p) \prec \tau$. The problem is that we do not have $(\sigma^{\mathsf{in}}_{\tau_1}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^j) = \mathsf{false}$. We solve this problem by rewriting ϕ_{y-1} (resp. ϕ_y) into the vector of terms ϕ'_{y-1} (resp. ϕ'_y) obtained by replacing any occurrence of $\mathsf{accept}^{\underline{\mathrm{ID}}_y}_{\tau_1}$ by:

$$\bigvee_{\substack{\tau_0 = \_^{\mathrm{FN}(j_0)\prec\tau_1} \\ \tau_0 \not\prec \tau_1\ {}^{\mathrm{NS}}\underline{\mathrm{ID}}_y(\_)}} \left(\begin{array}{l} \mathsf{inj\text{-}auth}_{\tau_1}(\underline{\mathrm{ID}}_y, j_0) \wedge \sigma^{\mathsf{in}}_{\tau_1}(\mathsf{e\text{-}auth}^{j_0}_{\mathrm{N}}) \neq \mathsf{UnknownId} \\ \wedge\ \pi_1(g(\phi^{\mathsf{in}}_{\tau_1})) = \mathrm{GUTI}^{j_0} \oplus \mathsf{f}^{\mathsf{r}}_{\mathsf{k}^{\underline{\mathrm{ID}}_y}}(\mathsf{n}^{j_0}) \wedge\ \pi_2(g(\phi^{\mathsf{in}}_{\tau_1})) = \mathsf{Mac}^5_{\mathsf{k}^{\underline{\mathrm{ID}}_y}_{\mathsf{m}}}(\langle \mathrm{GUTI}^{j_0}\ ,\ \mathsf{n}^{j_0}\rangle) \end{array}\right) \tag{4.65}$$

Which is sound using **(Equ1)**. We then have:

$$\mathsf{set\text{-}prf}^{\mathsf{f^r}}_{\mathsf{k}^{\underline{\mathrm{ID}}_y}}(\phi') \;=\; \{\mathsf{n}^p \mid \tau_1 = \_, \mathrm{FN}(p) \prec \tau\}$$

Therefore we can apply the PRF-$\mathsf{f^r}$ axioms as wanted: first we replace ϕ_{y-1} and ϕ_y by ϕ'_{y-1} and ϕ'_y using rule R; then we apply the PRF-$\mathsf{f^r}$ axiom; and finally we rewrite any term of the form (4.65) back into $\mathsf{accept}^{\underline{\mathrm{ID}}_y}_{\tau_1}$.

- Then, we use the $\oplus$-ind axiom to replace $\mathrm{GUTI}^j \oplus \mathsf{n}''^y_{i,j}$ with $\mathsf{n}^y_{i,j}$.

Derivation of (4.63) We use the same proof technique. We define, for every $0 \leq y \leq l$, the partially randomized terms $\underline{\mathsf{t\text{-}mac}}^y_{\underline{\tau}}(\mathrm{ID}_i, j)$:

$$\begin{aligned}
\underline{\mathsf{t\text{-}mac}}^y_{\underline{\tau}}(\mathrm{ID}_i, j) \;\equiv\; & \mathsf{if}\ \mathsf{eq}(\sigma_{\underline{\tau}}(\mathsf{e\text{-}auth}^j_{\mathrm{N}}), \underline{\mathrm{ID}}_1)\ \mathsf{then}\ \mathsf{n}'^1_{i,j} \\
& \quad \ldots \\
& \quad \mathsf{else\ if}\ \mathsf{eq}(\sigma_{\underline{\tau}}(\mathsf{e\text{-}auth}^j_{\mathrm{N}}), \underline{\mathrm{ID}}_{y-1})\ \mathsf{then}\ \mathsf{n}'^{y-1}_{i,j} \\
& \quad \mathsf{else\ if}\ \mathsf{eq}(\sigma_{\underline{\tau}}(\mathsf{e\text{-}auth}^j_{\mathrm{N}}), \underline{\mathrm{ID}}_y)\ \mathsf{then}\ \mathsf{Mac}^5_{\mathsf{k}^{\underline{\mathrm{ID}}_y}_{\mathbf{m}}}(\langle \mathrm{GUTI}^j\,,\, \mathsf{n}^j\rangle) \\
& \quad \ldots \\
& \quad \mathsf{else}\ \mathsf{Mac}^5_{\mathsf{k}^{\underline{\mathrm{ID}}_l}_{\mathbf{m}}}(\langle \mathrm{GUTI}^j\,,\, \mathsf{n}^j\rangle)
\end{aligned}$$

Remark that:

$$[\underline{\mathsf{net\text{-}e\text{-}auth}}_{\underline{\tau}}(\mathsf{A}_i, j)]\underline{\mathsf{t\text{-}mac}}^0_{\underline{\tau}}(\mathrm{ID}_i, j) \;=\; [\underline{\mathsf{net\text{-}e\text{-}auth}}_{\underline{\tau}}(\mathsf{A}_i, j)]\underline{\mathsf{t\text{-}mac}}_{\underline{\tau}}(\mathsf{A}_i, j)$$

And that:

$$\frac{}{\phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0},\ \mathsf{n}_{i,j}, \mathsf{n}'_{i,j} \;\sim\; \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0},\ \mathsf{n}_{i,j},\ [\underline{\mathsf{net\text{-}e\text{-}auth}}_{\underline{\tau}}(\mathsf{A}_i, j)]\underline{\mathsf{t\text{-}mac}}^l_{\underline{\tau}}(\mathsf{A}_i, j)}\ \mathsf{indep\text{-}branch}$$

Hence by transitivity, to prove that there exists a derivation of Formula (4.63) it is sufficient to prove that, for every $0 < y \leq l$, that we have a derivation of $\psi_{y-1} \sim \psi_y$, where:

$$\begin{aligned}
\psi_{y-1} \;&\equiv\; \psi_{\underline{\tau_0}}, \mathsf{r\text{-}reveal}_{\tau_0},\ \mathsf{n}_{i,j},\ [\underline{\mathsf{net\text{-}e\text{-}auth}}_{\underline{\tau}}(\mathsf{A}_i, j)]\underline{\mathsf{t\text{-}mac}}^{y-1}_{\underline{\tau}}(\mathrm{ID}_i, j) \\
\psi_y \;&\equiv\; \psi_{\underline{\tau_0}}, \mathsf{r\text{-}reveal}_{\tau_0},\ \mathsf{n}_{i,j},\ [\underline{\mathsf{net\text{-}e\text{-}auth}}_{\underline{\tau}}(\mathsf{A}_i, j)]\underline{\mathsf{t\text{-}mac}}^{y}_{\underline{\tau}}(\mathrm{ID}_i, j)
\end{aligned}$$

Let $1 \leq y \leq B$, we are going to give a derivation of $\psi_{y-1} \sim \psi_y$. For this, we are going to use the PRF-MAC^5 axiom with key $\mathsf{k}^{\underline{\mathrm{ID}}_y}_{\mathbf{m}}$, to replace $\mathsf{Mac}^5_{\mathsf{k}^{\underline{\mathrm{ID}}_y}_{\mathbf{m}}}(\langle \mathrm{GUTI}^j\,,\, \mathsf{n}^j\rangle)$ with a fresh nonce $\tilde{\mathsf{n}}^y_{i,j}$. Observe that there is only one occurrence of $\mathsf{Mac}^5_{\mathsf{k}^{\underline{\mathrm{ID}}_y}_{\mathbf{m}}}(\langle \mathrm{GUTI}^j\,,\, \mathsf{n}^j\rangle)$ in ψ_{y-1} (and none in ψ_y). Moreover:

$$\begin{aligned}
\mathsf{set\text{-}mac}^5_{\mathsf{k}^{\underline{\mathrm{ID}}_y}}(\psi_{y-1}, \psi_y) \setminus \{\langle \mathrm{GUTI}^j\,,\, \mathsf{n}^j\rangle\} \;=\; & \\
& \{\langle \mathrm{GUTI}^p\,,\, \mathsf{n}^p\rangle \mid \tau_1 = _, \mathrm{FN}(p) \prec \tau\} \\
\cup\; & \left\{\langle \pi_1(g(\phi^{\mathsf{in}}_{\tau_1})) \oplus \mathsf{f^r_k}(\sigma^{\mathsf{in}}_{\tau_1}(\mathsf{e\text{-}auth}^{\mathsf{k}^{\underline{\mathrm{ID}}_y}}_{\mathrm{U}}))\,,\, \sigma^{\mathsf{in}}_{\tau_1}(\mathsf{e\text{-}auth}^{\mathsf{k}^{\underline{\mathrm{ID}}_y}}_{\mathrm{U}})\rangle \mid \tau_1 = _, \mathrm{FN}(p) \prec \tau\right\}
\end{aligned}$$

Let $\tau_1 = \_, \mathrm{FN}(p) \prec \tau$. Since GUTI^j is a fresh nonce, using =-ind and the injectivity of the pair:

$$\neg\big(\langle \mathrm{GUTI}^j\,,\, \mathsf{n}^j\rangle = \langle \mathrm{GUTI}^p\,,\, \mathsf{n}^p\rangle\big) \qquad \neg\big(\langle \mathrm{GUTI}^j\,,\, \mathsf{n}^j\rangle = \langle \pi_1(g(\phi^{\mathsf{in}}_{\tau_1})) \oplus \mathsf{f^r_k}(\sigma^{\mathsf{in}}_{\tau_1}(\mathsf{e\text{-}auth}^{\mathsf{k}^{\underline{\mathrm{ID}}_y}}_{\mathrm{U}}))\,,\, \sigma^{\mathsf{in}}_{\tau_1}(\mathsf{e\text{-}auth}^{\mathsf{k}^{\underline{\mathrm{ID}}_y}}_{\mathrm{U}})\rangle\big)$$

Therefore we can directly apply the PRF-MAC^5 axiom, which concludes this case.

4.12.7 Case $\mathsf{ai} = \mathrm{FU}_{\mathrm{ID}}(j)$

We know that $\underline{\mathsf{ai}} = \mathrm{FU}_{\nu_\tau(\mathrm{ID})}(j)$. Here $\mathsf{l\text{-}reveal}_\tau$ and $\mathsf{l\text{-}reveal}_{\tau_0}$ coincides everywhere except on the pairs:

$$\underbrace{\begin{array}{c}\sigma_\tau(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \\ \mathsf{if}\ \sigma_\tau(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}})\ \mathsf{then}\ \sigma_\tau(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) \\ \mathsf{else\ defaut}\end{array}}_{\mathsf{m\text{-}suci}^{\mathrm{ID}}_{\tau}} \;\sim\; \underbrace{\begin{array}{c}\sigma_{\underline{\tau}}(\mathsf{valid\text{-}guti}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}}) \\ \mathsf{if}\ \sigma_{\underline{\tau}}(\mathsf{valid\text{-}guti}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}})\ \mathsf{then}\ \sigma_{\underline{\tau}}(\mathrm{GUTI}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}}) \\ \mathsf{else\ defaut}\end{array}}_{\mathsf{m\text{-}suci}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}}}$$

Moreover, we need to show that $\mathsf{accept}^{\mathrm{ID}}_{\tau} \sim \mathsf{accept}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}}$.First, using FA and Dup, we check that it is sufficient to give a derivation of:

$$\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{accept}^{\mathrm{ID}}_{\tau}, \mathsf{m\text{-}suci}\,{}^{\mathrm{ID}}_{\tau} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{accept}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}}, \mathsf{m\text{-}suci}\,{}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}} \tag{4.66}$$

Using **(Equ1)** twice:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \leftrightarrow \bigvee_{\substack{\tau_1 = \_, \mathrm{FN}(j_0) \prec \tau \\ \tau_1 \not\prec_\tau \mathrm{NS}_{\mathrm{ID}}(\_)}} \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau} \qquad \mathsf{accept}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}} \leftrightarrow \bigvee_{\substack{\tau_1 = \_, \mathrm{FN}(j_0) \prec \tau \\ \tau_1 \not\prec_\tau \mathrm{NS}_{\mathrm{ID}}(\_)}} \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\underline{\tau}}$$

Let:

$$\{j_0, \ldots, j_l\} = \{i \mid \tau' = \_, \mathrm{FN}(i) \prec \tau \wedge \tau' \not\prec_\tau \mathrm{NS}_{\mathrm{ID}}(\_)\}$$

We check that:

$$\{j_0, \ldots, j_l\} = \{i \mid \tau' = \_, \mathrm{FN}(i) \prec \underline{\tau} \wedge \tau' \not\prec_{\underline{\tau}} \mathrm{NS}_{\nu_\tau(\mathrm{ID})}(\_)\}$$

For all $0 \leq i \leq l$, let τ_{j_i} be such that $\tau_{j_i} = \_, \mathrm{FN}(j_i) \prec \tau$. One can check that:

$$\begin{aligned}
\mathsf{m\text{-}suci}\,{}^{\mathrm{ID}}_{\tau} = {} & \mathsf{if}\ \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_{j_0}}_{\mathsf{u}:\tau}\ \mathsf{then}\ \mathrm{GUTI}^{j_0} \\
& \mathsf{else\ if}\ \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_{j_1}}_{\mathsf{u}:\tau}\ \mathsf{then}\ \mathrm{GUTI}^{j_1} \\
& \quad \ldots \\
& \mathsf{else}\ \mathrm{GUTI}^{j_l}
\end{aligned}
\qquad
\begin{aligned}
\mathsf{m\text{-}suci}\,{}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}} = {} & \mathsf{if}\ \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_{j_0}}_{\mathsf{u}:\underline{\tau}}\ \mathsf{then}\ \mathrm{GUTI}^{j_0} \\
& \mathsf{else\ if}\ \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_{j_1}}_{\mathsf{u}:\underline{\tau}}\ \mathsf{then}\ \mathrm{GUTI}^{j_1} \\
& \quad \ldots \\
& \mathsf{else}\ \mathrm{GUTI}^{j_l}
\end{aligned}$$

We can now start giving a derivation of (4.66):

$$\dfrac{\dfrac{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \left(\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_{j_i}}_{\mathsf{u}:\tau}\right)_{i\leq l} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \left(\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_{j_i}}_{\mathsf{u}:\underline{\tau}}\right)_{i\leq l}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \left(\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_{j_i}}_{\mathsf{u}:\tau}\right)_{i\leq l}, \left(\mathrm{GUTI}^{j_i}\right)_{i\leq l} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \left(\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_{j_i}}_{\mathsf{u}:\underline{\tau}}\right)_{i\leq l}, \left(\mathrm{GUTI}^{j_i}\right)_{i\leq l}}\ \mathsf{Dup}^*}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{accept}^{\mathrm{ID}}_{\tau}, \mathsf{m\text{-}suci}\,{}^{\mathrm{ID}}_{\tau} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{accept}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}}, \mathsf{m\text{-}suci}\,{}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}}}\ \mathsf{FA}^*$$

Since for all $1 \leq i \leq l$, $(\mathrm{GUTI}^{j_i} \sim \mathrm{GUTI}^{j_i}) \in \mathsf{reveal}_{\tau_0}$. We conclude using **(Der2)** for every $0 \leq i \leq l$:

$$\dfrac{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \left(\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_{j_i}}_{\mathsf{u}:\tau}\right)_{i\leq l} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \left(\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_{j_i}}_{\mathsf{u}:\underline{\tau}}\right)_{i\leq l}}\ \mathsf{FA}^*$$

4.12.8 Case $\mathsf{ai} = \mathrm{TU}_{\mathrm{ID}}(j, 0)$

Let $\underline{\mathrm{ID}} = \nu_\tau(\mathrm{ID})$, we know that $\underline{\mathsf{ai}} = \mathrm{TU}_{\underline{\mathrm{ID}}}(j, 0)$. $\mathsf{l\text{-}reveal}_{\tau}$ and $\mathsf{l\text{-}reveal}_{\tau_0}$ coincides everywhere except on:

$$\sigma_\tau(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \sim \sigma_{\underline{\tau}}(\mathsf{valid\text{-}guti}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \qquad \sigma_\tau(\mathsf{s\text{-}valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \sim \sigma_{\underline{\tau}}(\mathsf{s\text{-}valid\text{-}guti}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \qquad \mathsf{m\text{-}suci}\,{}^{\mathrm{ID}}_{\tau} \sim \mathsf{m\text{-}suci}\,{}^{\underline{\mathrm{ID}}}_{\underline{\tau}}$$

Handling these is simple since:

$$\sigma_\tau(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \equiv \mathsf{false} \qquad \sigma_{\underline{\tau}}(\mathsf{valid\text{-}guti}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \equiv \mathsf{false} \qquad \sigma_\tau(\mathsf{s\text{-}valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \equiv \sigma^{\mathsf{in}}_{\tau}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}})$$

$$\sigma_{\underline{\tau}}(\mathsf{s\text{-}valid\text{-}guti}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \equiv \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{valid\text{-}guti}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \qquad \mathsf{m\text{-}suci}\,{}^{\mathrm{ID}}_{\tau} = \mathsf{defaut} \qquad \mathsf{m\text{-}suci}\,{}^{\underline{\mathrm{ID}}}_{\underline{\tau}} = \mathsf{defaut}$$

Observe that:

$$t_\tau = \mathsf{if}\ \sigma^{\mathsf{in}}_{\tau}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}})\ \mathsf{then}\ \mathsf{m\text{-}suci}\,{}^{\mathrm{ID}}_{\tau_0}\ \mathsf{else}\ \mathsf{NoGuti} \qquad t_{\underline{\tau}} = \mathsf{if}\ \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{valid\text{-}guti}^{\underline{\mathrm{ID}}}_{\mathrm{U}})\ \mathsf{then}\ \mathsf{m\text{-}suci}\,{}^{\underline{\mathrm{ID}}}_{\underline{\tau_0}}\ \mathsf{else}\ \mathsf{NoGuti}$$

Since $(\sigma^{\mathsf{in}}_{\tau}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}), \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{valid\text{-}guti}^{\underline{\mathrm{ID}}}_{\mathrm{U}})) \in \mathsf{reveal}_{\tau_0}$ and $(\mathsf{m\text{-}suci}\,{}^{\mathrm{ID}}_{\tau_0} \sim \mathsf{m\text{-}suci}\,{}^{\underline{\mathrm{ID}}}_{\underline{\tau_0}}) \in \mathsf{reveal}_{\tau_0}$, we conclude:

$$\dfrac{\dfrac{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_{\tau}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}), \mathsf{m\text{-}suci}\,{}^{\mathrm{ID}}_{\tau}, \mathsf{NoGuti} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{valid\text{-}guti}^{\underline{\mathrm{ID}}}_{\mathrm{U}}), \mathsf{m\text{-}suci}\,{}^{\underline{\mathrm{ID}}}_{\underline{\tau}}, \mathsf{NoGuti}}\ \mathsf{Dup}^*}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, t_\tau \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, t_{\underline{\tau}}}\ \mathsf{Simp}$$

4.12.9 Case $\mathsf{ai} = \mathrm{TN}(j,0)$

We know that $\underline{\mathsf{ai}} = \mathrm{TN}(j,0)$. Using **(A6)**, we know that for every $\mathrm{ID} \neq \mathrm{ID}'$, $\neg\mathsf{accept}^{\mathrm{ID}}_{\tau} \leftrightarrow \neg\mathsf{accept}^{\mathrm{ID}'}_{\tau}$. Therefore the answer from the network does not depend on the order in which we make the $\mathsf{accept}^{\mathrm{ID}}_{\tau}$ tests. Formally, the following list of conditionals is a CS partition:

$$\Big((\mathsf{accept}^{\mathrm{ID}}_{\tau})_{\mathrm{ID}\in\mathcal{S}_{\mathsf{id}}} \,,\, \bigwedge_{\mathrm{ID}\in\mathcal{S}_{\mathsf{id}}} \neg\mathsf{accept}^{\mathrm{ID}}_{\tau} \Big)$$

To get a uniform notation, we let $\mathsf{accept}^{\mathrm{ID}_{\mathsf{dum}}}_{\tau} \equiv \bigwedge_{\mathrm{ID}\in\mathcal{S}_{\mathsf{id}}} \neg\mathsf{accept}^{\mathrm{ID}}_{\tau}$, and $\mathcal{S}_{\mathsf{ext\text{-}id}} = \mathcal{S}_{\mathsf{id}} \cup \{\mathrm{ID}_{\mathsf{dum}}\}$. Hence using Proposition 4.18 we get that:

$$t_\tau \;=\; \underset{\mathrm{ID}\in\mathcal{S}_{\mathsf{ext\text{-}id}}}{\mathsf{case}} \; (\mathsf{accept}^{\mathrm{ID}}_{\tau} : \mathsf{msg}^{\mathrm{ID}}_{\tau})$$

We are now going to show that for every $\mathrm{ID} \in \mathcal{S}_{\mathsf{ext\text{-}id}}$, the term $\mathsf{msg}^{\mathrm{ID}}_{\tau}$ can be replaced by $\langle \mathsf{n}^j , \mathsf{n}^{\oplus}_{\mathrm{ID}} , \mathsf{n}^{\mathsf{Mac}}_{\mathrm{ID}} \rangle$ (where $(\mathsf{n}^{\oplus}_{\mathrm{ID}})_{\mathrm{ID}\in\mathcal{S}_{\mathsf{ext\text{-}id}}}$ and $(\mathsf{n}^{\mathsf{Mac}}_{\mathrm{ID}})_{\mathrm{ID}\in\mathcal{S}_{\mathsf{ext\text{-}id}}}$ are fresh distinct nonces). We will then conclude easily using the Fresh axiom.

Let $\mathrm{ID}_1, \ldots, \mathrm{ID}_l$ be an arbitrary enumeration of $\mathcal{S}_{\mathsf{ext\text{-}id}}$. For every $1 \leq n \leq l$, and for every $\mathrm{ID}_i \in \{\mathrm{ID}_1, \ldots, \mathrm{ID}_l\}$, we let:

$$\mathsf{rnd\text{-}msg}^{\mathrm{ID}_i}_{n} \equiv \begin{cases} \langle \mathsf{n}^j , \mathsf{n}^{\oplus}_{\mathrm{ID}_i} , \mathsf{n}^{\mathsf{Mac}}_{\mathrm{ID}_i} \rangle & \text{if } i \leq n \\ \mathsf{rnd\text{-}msg}^{\mathrm{ID}_i}_{\tau} & \text{if } i > n \end{cases}$$

And we let t_n be the term t_τ where the subterms $\mathsf{msg}^{\mathrm{ID}}_{\tau}$ have been replaced by $\langle \mathsf{n}^j , \mathsf{n}^{\oplus}_{\mathrm{ID}} , \mathsf{n}^{\mathsf{Mac}}_{\mathrm{ID}} \rangle$ for the first n identities:

$$t_n \equiv \underset{\mathrm{ID}\in\mathcal{S}_{\mathsf{ext\text{-}id}}}{\mathsf{case}} \; (\mathsf{accept}^{\mathrm{ID}}_{\tau} : \mathsf{rnd\text{-}msg}^{\mathrm{ID}}_{n})$$

We check that $t_0 \equiv t_\tau$.

Part 1 We now show that for every $1 \leq n \leq l$, we have a derivation of:

$$\phi^{\mathsf{in}}_{\tau}, \mathsf{I\text{-}reveal}_{\tau_0}, t_{n-1} \;\sim\; \phi^{\mathsf{in}}_{\tau}, \mathsf{I\text{-}reveal}_{\tau_0}, t_n \tag{4.67}$$

Let n be in $\{1, \ldots, l\}$. Let $\mathrm{ID} = \mathrm{ID}_n$, $\mathsf{k} = \mathsf{k}^{\mathrm{ID}}$ and $\mathsf{k_m} = \mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}$. We are going to apply PRF-f axiom with key k to replace $\mathsf{f_k}(\mathsf{n}^j)$ by n_{ID}, where n_{ID} is a fresh nonce. Recall that:

$$\mathsf{msg}^{\mathrm{ID}}_{\tau} \;\equiv\; \langle \mathsf{n}^j , \underbrace{\sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})}_{u_{\mathrm{SQN}}} \oplus \mathsf{f}_{\mathsf{k}^{\mathrm{ID}}}(\mathsf{n}^j) , \underbrace{\mathsf{Mac}^3_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle \mathsf{n}^j , \sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) , \sigma^{\mathsf{in}}_{\tau}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \rangle)}_{u_{\mathsf{Mac}}} \rangle$$

We let ψ be the context with one hole (which has only one occurrence) such that:

$$\psi[\langle \mathsf{n}^j , u_{\mathrm{SQN}} \oplus \mathsf{f}_{\mathsf{k}^{\mathrm{ID}}}(\mathsf{n}^j) , u_{\mathsf{Mac}} \rangle] \equiv \phi^{\mathsf{in}}_{\tau}, \mathsf{I\text{-}reveal}_{\tau_0}, t_{n-1} \qquad \psi[\langle \mathsf{n}^j , \mathsf{n}^{\oplus}_{\mathrm{ID}} , \mathsf{n}^{\mathsf{Mac}}_{\mathrm{ID}} \rangle] \equiv \phi^{\mathsf{in}}_{\tau}, \mathsf{I\text{-}reveal}_{\tau_0}, t_n$$

Let $\psi_0[] \equiv \psi[\langle \mathsf{n}^j , u_{\mathrm{SQN}} \oplus [] , u_{\mathsf{Mac}} \rangle]$. Notice that:

$$\mathsf{set\text{-}prf}^{\mathsf{f}}_{\mathsf{k}}(\psi_0[]) \;=\; \{\pi_1(\phi^{\mathsf{in}}_{\tau_1}) \mid \tau_1 = \_, \mathrm{TU}_{\mathrm{ID}}(p,1) \prec \tau\} \;\cup\; \{\mathsf{n}^p \mid \tau_1 = \_, \mathrm{TN}(p) \prec \tau\}$$

We want to get rid of the sub-terms of the form $\mathsf{f_k}(\pi_1(\phi^{\mathsf{in}}_{\tau_1}))$, for any τ_1 such that $\tau_1 = \_, \mathrm{TU}_{\mathrm{ID}}(p,1) \prec \tau$. To do this, for every $\tau_1 = \_, \mathrm{TU}_{\mathrm{ID}}(p,1) \prec \tau$, we let $\tau_3 = \_, \mathrm{TU}_{\mathrm{ID}}(j_p,0) \prec \tau$, and we apply **(StrEqu2)** to rewrite all occurrence of $\mathsf{accept}^{\mathrm{ID}}_{\tau_1}$ in ψ_0 using:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau_1} \leftrightarrow \bigvee_{\substack{\tau_2 = \_, \mathrm{TN}(j_1,0) \\ \tau_3 \prec \tau_1 \; \tau_2 \prec \tau_1 \; \tau_1}} \mathsf{part\text{-}tr}^{\mathsf{n}:\tau_2}_{\mathsf{u}:\tau_3,\tau_1} \tag{4.68}$$

This yields a vector of terms $\psi'_0[]$ with one hole. It is easy to check that:

$$\mathsf{set\text{-}prf}^{\mathsf{f}}_{\mathsf{k}}(\psi'_0[]) \;=\; \{\mathsf{n}^p \mid \tau_1 = \_, \mathrm{TN}(p) \prec \tau\}$$

By validity of τ, we know that for every $\tau_1 = \_, \mathrm{TN}(p) \prec \tau$, we have $p \neq j$. Therefore using Fresh we have $\neg(\mathsf{n}^j = \mathsf{n}^P)$. It follows that we can apply the PRF-f axiom in $\psi'_0[\mathsf{f_k}(\mathsf{n}^j)]$, replacing $\mathsf{f_k}(\mathsf{n}^j)$ by n_{ID}, which

yields $\psi_0'[\mathsf{n}_{\text{ID}}]$. More precisely, we deconstruct the context ψ_0' using FA, without touching at the mac terms, until we get $\vec{w}, \mathsf{f}_{\mathsf{k}}(\mathsf{n}^j) \sim \vec{w}, \mathsf{n}_{\text{ID}}$, at which point we can apply the PRF-f axiom. We then rewrite any term of the form in (4.68) back into $\mathsf{accept}^{\text{ID}}_{\tau_1}$, obtaining $\psi_0[\mathsf{n}_{\text{ID}}] \equiv \psi[\langle \mathsf{n}^j, u_{\text{SQN}} \oplus \mathsf{n}_{\text{ID}}, u_{\mathsf{Mac}}\rangle]$. We then use $\oplus$-ind to replace $u_{\text{SQN}} \oplus \mathsf{n}_{\text{ID}}$ by $\mathsf{n}^{\oplus}_{\text{ID}}$. For this, we use the fact that $\mathsf{len}(u_{\text{SQN}}) = \mathsf{len}(\mathsf{n}_{\text{ID}})$ by Proposition 4.11.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vec{w}, \mathsf{f}_{\mathsf{k}}(\mathsf{n}^j) \sim \vec{w}, \mathsf{n}_{\text{ID}}}\ \text{PRF-f}}{\vdots}}{\psi_0'[\mathsf{f}_{\mathsf{k}}(\mathsf{n}^j)] \sim \psi_0'[\mathsf{n}_{\text{ID}}]}\ \text{FA}^*}\quad \dfrac{\dfrac{\psi[\langle \mathsf{n}^j, \mathsf{n}^{\oplus}_{\text{ID}}, u_{\mathsf{Mac}}\rangle] \sim \psi[\langle \mathsf{n}^j, \mathsf{n}^{\oplus}_{\text{ID}}, \mathsf{n}^{\mathsf{Mac}}_{\text{ID}}\rangle]}{\psi[\langle \mathsf{n}^j, u_{\text{SQN}} \oplus \mathsf{n}_{\text{ID}}, u_{\mathsf{Mac}}\rangle] \sim \psi[\langle \mathsf{n}^j, \mathsf{n}^{\oplus}_{\text{ID}}, \mathsf{n}^{\mathsf{Mac}}_{\text{ID}}\rangle]}\ \oplus\text{-ind}}{\psi_0'[\mathsf{n}_{\text{ID}}] \sim \psi[\langle \mathsf{n}^j, \mathsf{n}^{\oplus}_{\text{ID}}, \mathsf{n}^{\mathsf{Mac}}_{\text{ID}}\rangle]}\ R}{\psi_0'[\mathsf{f}_{\mathsf{k}}(\mathsf{n}^j)] \sim \psi[\langle \mathsf{n}^j, \mathsf{n}^{\oplus}_{\text{ID}}, \mathsf{n}^{\mathsf{Mac}}_{\text{ID}}\rangle]}\ \text{Trans}$$

$$\dfrac{\dfrac{\psi_0'[\mathsf{f}_{\mathsf{k}}(\mathsf{n}^j)] \sim \psi[\langle \mathsf{n}^j, \mathsf{n}^{\oplus}_{\text{ID}}, \mathsf{n}^{\mathsf{Mac}}_{\text{ID}}\rangle]}{\psi[\langle \mathsf{n}^j, u_{\text{SQN}} \oplus \mathsf{f}_{\mathsf{k}^{\text{ID}}}(\mathsf{n}^j), u_{\mathsf{Mac}}\rangle] \sim \psi[\langle \mathsf{n}^j, \mathsf{n}^{\oplus}_{\text{ID}}, \mathsf{n}^{\mathsf{Mac}}_{\text{ID}}\rangle]}\ R}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, t_{n-1} \sim \phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, t_n}\ R$$

We now the same thing with u_{Mac}, applying PRF-MAC^3 axiom to replace u_{Mac} by $\mathsf{n}^{\mathsf{Mac}}_{\text{ID}}$. The proof is similar to the one we just did for PRF-f, and we omit the details. We conclude using Refl. This yields:

$$\dfrac{\dfrac{\psi[\langle \mathsf{n}^j, \mathsf{n}^{\oplus}_{\text{ID}}, \mathsf{n}^{\mathsf{Mac}}_{\text{ID}}\rangle] \sim \psi[\langle \mathsf{n}^j, \mathsf{n}^{\oplus}_{\text{ID}}, \mathsf{n}^{\mathsf{Mac}}_{\text{ID}}\rangle]}{\vdots}\ \text{Refl}}{\psi[\langle \mathsf{n}^j, \mathsf{n}^{\oplus}_{\text{ID}}, u_{\mathsf{Mac}}\rangle] \sim \psi[\langle \mathsf{n}^j, \mathsf{n}^{\oplus}_{\text{ID}}, \mathsf{n}^{\mathsf{Mac}}_{\text{ID}}\rangle]}$$

Part 2 Using the fact that $t_0 \equiv t_\tau$ and (4.67), and using the transitivity axiom, we get:

$$\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, t_\tau \sim \phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, t_l$$

Moreover, using the indep-branch axiom we know that:

$$\dfrac{}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, t_l \sim \phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{n}}\ \text{indep-branch}$$

where n is a fresh nonce. Using transitivity again, we get a derivation of:

$$\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, t_\tau \sim \phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{n} \tag{4.69}$$

Repeating everything we did in **Part 1**, we can show that we have a derivation of:

$$\phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{n}' \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, t_{\underline{\tau}} \tag{4.70}$$

where n' is a fresh nonce. We then conclude using the transitivity and Fresh:

$$\dfrac{\dfrac{(4.69)}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, t_\tau \sim \phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{n}} \quad \dfrac{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{n} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{n}'}\ \text{Fresh} \quad \dfrac{(4.70)}{\phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{n}' \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, t_{\underline{\tau}}}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, t_\tau \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, t_{\underline{\tau}}}\ \text{Trans}$$

4.12.10 Case $\mathsf{ai} = \text{TU}_{\text{ID}}(j, 1)$

We know that $\underline{\mathsf{ai}} = \text{TU}_{\nu_\tau(\text{ID})}(j, 1)$. Let $\underline{\text{ID}} = \nu_\tau(\text{ID})$. By validity of τ, we know that there exists $\tau_2 = \_, \text{TU}_{\text{ID}}(j, 0)$ such that $\tau_2 \prec \tau$. Here $\mathsf{l\text{-}reveal}_\tau$ and $\mathsf{l\text{-}reveal}_{\tau_0}$ coincides everywhere except on:

$$\sigma_\tau(\text{SQN}^{\text{ID}}_{\text{U}}) - \sigma^{\mathsf{in}}_{\tau}(\text{SQN}^{\text{ID}}_{\text{U}}) \sim \sigma_{\underline{\tau}}(\text{SQN}^{\underline{\text{ID}}}_{\text{U}}) - \sigma^{\mathsf{in}}_{\underline{\tau}}(\text{SQN}^{\underline{\text{ID}}}_{\text{U}}) \qquad \sigma_\tau(\text{e-auth}^{\text{ID}}_{\text{U}}) \sim \sigma_{\underline{\tau}}(\text{e-auth}^{\underline{\text{ID}}}_{\text{U}})$$

$$\left(\mathsf{Mac}^4_{\mathsf{k}^{\text{ID}}_{\mathbf{m}}}(\mathsf{n}^{j_0}) \sim \mathsf{Mac}^4_{\mathsf{k}^{\underline{\text{ID}}}_{\mathbf{m}}}(\mathsf{n}^{j_0})\right)_{\substack{\tau_1 = \_, \text{TN}(j_0, 0) \\ \tau_2 \prec \tau\, \tau_1}}$$

First, using **(StrEqu2)** twice we know that:

$$\mathsf{accept}^{\text{ID}}_{\tau} \leftrightarrow \bigvee_{\substack{\tau_1 = \_, \text{TN}(j_1, 0) \\ \tau_2 \prec \tau\, \tau_1}} \mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2, \tau} \qquad \mathsf{accept}^{\underline{\text{ID}}}_{\underline{\tau}} \leftrightarrow \bigvee_{\substack{\tau_1 = \_, \text{TN}(j_1, 0) \\ \tau_2 \prec \tau\, \tau_1}} \mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\underline{\tau_2}, \underline{\tau}}$$

Using **(Der3)** we know that for every $\tau_1 = \_, \mathrm{TN}(j_1, 0)$ such that $\tau_2 \prec_\tau \tau_1$ we have a derivation:

$$\frac{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{part\text{-}tr}^{\mathsf{n}:\underline{\tau_1}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}}}\ \mathsf{Simp} \tag{4.71}$$

Therefore we can build the following derivation:

$$\frac{\dfrac{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \left(\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau}\right)_{\substack{\tau_1 = \_, \mathrm{TN}(j_1,0) \\ \tau_2 \prec_\tau \tau_1}} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \left(\mathsf{part\text{-}tr}^{\mathsf{n}:\underline{\tau_1}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}}\right)_{\substack{\tau_1 = \_, \mathrm{TN}(j_1,0) \\ \tau_2 \prec_\tau \tau_1}}}\ \mathsf{Simp}}{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{accept}^{\mathrm{ID}}_\tau \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{accept}^{\mathrm{ID}}_{\underline{\tau}}}\ \mathsf{Simp} \tag{4.72}$$

Part 1 We can check that for every $\tau_1 = \_, \mathrm{TN}(j_1, 0)$ such that $\tau_2 \prec_\tau \tau_1$:

$$\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \to \sigma_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^{j_1} \qquad \mathsf{part\text{-}tr}^{\mathsf{n}:\underline{\tau_1}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}} \to \sigma_{\underline{\tau}}(\mathsf{e\text{-}auth}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) = \mathsf{n}^{j_1}$$
$$\neg\mathsf{accept}^{\mathrm{ID}}_\tau \to \sigma_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{fail} \qquad \neg\mathsf{accept}^{\underline{\mathrm{ID}}}_{\underline{\tau}} \to \sigma_{\underline{\tau}}(\mathsf{e\text{-}auth}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) = \mathsf{fail}$$

And $(\mathsf{n}^{j_1}, \mathsf{n}^{j_1}) \in \mathsf{reveal}_{\tau_0}$. Therefore we can decompose $\sigma_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}})$ and $\sigma_{\underline{\tau}}(\mathsf{e\text{-}auth}^{\underline{\mathrm{ID}}}_{\mathrm{U}})$ using FA and get rid of the resulting terms using (4.71) and (4.72):

$$\frac{\dfrac{\dfrac{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{accept}^{\mathrm{ID}}_\tau, \left(\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau}, \mathsf{n}^{j_1}\right)_{\substack{\tau_1 = \_, \mathrm{TN}(j_1,0) \\ \tau_2 \prec_\tau \tau_1}}, \mathsf{fail} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{accept}^{\underline{\mathrm{ID}}}_{\underline{\tau}}, \left(\mathsf{part\text{-}tr}^{\mathsf{n}:\underline{\tau_1}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}}, \mathsf{n}^{j_1}\right)_{\substack{\tau_1 = \_, \mathrm{TN}(j_1,0) \\ \tau_2 \prec_\tau \tau_1}}, \mathsf{fail}}\ \mathsf{Simp}}{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{if}\ \mathsf{accept}^{\mathrm{ID}}_\tau\ \mathsf{then}\ \underset{\substack{\tau_1 = \_, \mathrm{TN}(j_1,0) \\ \tau_2 \prec_\tau \tau_1}}{\mathsf{case}}\ (\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} : \mathsf{n}^{j_1})\ \mathsf{else}\ \mathsf{fail} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{if}\ \mathsf{accept}^{\underline{\mathrm{ID}}}_{\underline{\tau}}\ \mathsf{then}\ \underset{\substack{\tau_1 = \_, \mathrm{TN}(j_1,0) \\ \tau_2 \prec_\tau \tau_1}}{\mathsf{case}}\ (\mathsf{part\text{-}tr}^{\mathsf{n}:\underline{\tau_1}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}} : \mathsf{n}^{j_1})\ \mathsf{else}\ \mathsf{fail}}\ \mathsf{Simp}}{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \sigma_\tau(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \sigma_{\underline{\tau}}(\mathsf{e\text{-}auth}^{\underline{\mathrm{ID}}}_{\mathrm{U}})}\ R \tag{4.73}$$

Part 2 Observe that for every $\tau_1 = \_, \mathrm{TN}(j_1, 0)$ such that $\tau_2 \prec_\tau \tau_1$:

$$\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \to \sigma_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) - \sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) = \mathbf{1} \qquad \mathsf{part\text{-}tr}^{\mathsf{n}:\underline{\tau_1}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}} \to \sigma_{\underline{\tau}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) - \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) = \mathbf{1}$$
$$\neg\mathsf{accept}^{\mathrm{ID}}_\tau \to \sigma_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) - \sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) = 0 \qquad \neg\mathsf{accept}^{\underline{\mathrm{ID}}}_{\underline{\tau}} \to \sigma_{\underline{\tau}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) - \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) = 0$$

It is then easy to adapt the derivation in (4.73) to get a derivation of (we omit the details):

$$\frac{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \sigma_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) - \sigma^{\mathsf{in}}_\tau(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \sigma_{\underline{\tau}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) - \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{U}})}\ \mathsf{Simp} \tag{4.74}$$

Part 3 We finally take care of t_τ and the Mac^4 terms. First, we check that for every $\tau_1 = \_, \mathrm{TN}(j_1, 0)$ such that $\tau_2 \prec_\tau \tau_1$:

$$\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \to t_\tau = \mathsf{Mac}^4_{\mathsf{k}^{\mathrm{ID}}_{\mathbf{m}}}(\mathsf{n}^{j_0}) \qquad \mathsf{part\text{-}tr}^{\mathsf{n}:\underline{\tau_1}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}} \to t_{\underline{\tau}} = \mathsf{Mac}^4_{\mathsf{k}^{\underline{\mathrm{ID}}}_{\mathbf{m}}}(\mathsf{n}^{j_0})$$
$$\neg\mathsf{accept}^{\mathrm{ID}}_\tau \to t_\tau = \mathsf{error} \qquad \neg\mathsf{accept}^{\underline{\mathrm{ID}}}_{\underline{\tau}} \to t_{\underline{\tau}} = \mathsf{error}$$

Similarly to what we did in (4.73), we decompose t_τ and $t_{\underline{\tau}}$ using (4.71) and (4.72). Omitting the detail of the derivation, this yield:

$$\frac{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \left(\mathsf{Mac}^4_{\mathsf{k}^{\mathrm{ID}}_{\mathbf{m}}}(\mathsf{n}^{j_0})\right)_{\substack{\tau_1 = \_, \mathrm{TN}(j_0,0) \\ \tau_2 \prec_\tau \tau_1}} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \left(\mathsf{Mac}^4_{\mathsf{k}^{\underline{\mathrm{ID}}}_{\mathbf{m}}}(\mathsf{n}^{j_0})\right)_{\substack{\tau_1 = \_, \mathrm{TN}(j_0,0) \\ \tau_2 \prec_\tau \tau_1}}}{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, t_\tau \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, t_{\underline{\tau}}}\ \mathsf{Simp}$$

Observe that the Mac^4 terms here are exactly the Mac^4 terms in $\mathsf{l\text{-}reveal}_\tau \backslash \mathsf{l\text{-}reveal}_{\tau_0}$. To conclude this proof, it only remains to give a derivation of:

$$\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \left(\mathsf{Mac}^4_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\mathsf{n}^{j_0})\right)_{\substack{\tau_1 = \_, \mathrm{TN}(j_0,0) \\ \tau_2 \prec_\tau \tau_1}} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \left(\mathsf{Mac}^4_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\mathsf{n}^{j_0})\right)_{\substack{\tau_1 = \_, \mathrm{TN}(j_0,0) \\ \tau_2 \prec_\tau \tau_1}}$$

For every $\tau_1 = \_, \mathrm{TN}(j_1, 0)$ such that $\tau_2 \prec_\tau \tau_1$, we are going to apply the PRF-MAC^4 axiom with key $\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}$ to replace $\mathsf{Mac}^4_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\mathsf{n}^{j_0})$ by a fresh nonce n_{τ_1}. Let $\psi \equiv \phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}$, observe that:

$$\mathsf{set\text{-}mac}^4_{\mathrm{ID}}(\psi) = \left\{\pi_1(g(\phi^{\mathsf{in}}_{\tau_a})) \mid \tau_a = \_, \mathrm{TU}_{\mathrm{ID}}(j_a, 1) \prec \tau\right\} \cup \left\{\mathsf{n}^{j_n} \mid \tau_n = \_, \mathrm{TN}(j_n, 1) \prec \tau\right\}$$

Let:

$$T = \left\{\mathsf{n}^{j_0} \mid \tau_1 = \_, \mathrm{TN}(j_0, 0) \wedge \tau_2 \prec_\tau \tau_1\right\}$$

Our goal is to rewrite ψ into a vector of terms ψ_1 such that $\mathsf{set\text{-}mac}^4_{\mathrm{ID}}(\psi_1) \cap T = \emptyset$. This will allow us to apply the PRF-MAC^4 axiom. We are going to rewrite ψ as follows:

- Let $\tau_a = \_, \mathrm{TU}_{\mathrm{ID}}(j_a, 1) \prec \tau$. By validity of τ, we know that $\tau_a \prec_\tau \tau_2$, and that there exists $\tau_b = \_, \mathrm{TU}_{\mathrm{ID}}(j_a, 0) \prec_\tau \tau_a$. Using **(StrEqu2)**, we know that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau_a} \leftrightarrow \bigvee_{\substack{\tau_x = \_, \mathrm{TN}(j_x, 0) \\ \tau_b \prec_\tau \tau_x \prec_\tau \tau_a}} \mathsf{part\text{-}tr}^{\mathsf{n}:\tau_x}_{\mathsf{u}:\tau_b, \tau_a}$$

  We let $\alpha^{\mathrm{ID}}_{\tau_a}$ be the right-hand side of the equation above. Using this, we can check that:

$$t_{\tau_a} = \mathsf{if}\ \alpha^{\mathrm{ID}}_{\tau_a}\ \mathsf{then}\ \underset{\substack{\tau_x = \_, \mathrm{TN}(j_x, 0) \\ \tau_b \prec_\tau \tau_x \prec_\tau \tau_a}}{\mathsf{case}} \left(\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_x}_{\mathsf{u}:\tau_b, \tau_a} : \mathsf{Mac}^4_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\mathsf{n}^{j_x})\right)\ \mathsf{else\ error}$$

  Let $\kappa^{\mathrm{ID}}_{\tau_a}$ be the right-hand side of the equation above. For every $\tau_x = \_, \mathrm{TN}(j_x, 0)$, if $\mathsf{n}^{j_x} \in \mathsf{set\text{-}mac}^4_{\mathrm{ID}}(\alpha^{\mathrm{ID}}_{\tau_a}, \kappa^{\mathrm{ID}}_{\tau_a})$ then $\tau_x \prec_\tau \tau_a$. Therefore:

$$\begin{aligned} & \mathsf{set\text{-}mac}^4_{\mathrm{ID}}\left(\alpha^{\mathrm{ID}}_{\tau_a}, \kappa^{\mathrm{ID}}_{\tau_a}\right) \cap T \\ \subseteq\ & \left\{\mathsf{n}^{j_x} \mid \tau_x = \_, \mathrm{TN}(j_x, 0) \wedge \tau_x \prec_\tau \tau_a\right\} \cap \left\{\mathsf{n}^{j_0} \mid \tau_1 = \_, \mathrm{TN}(j_0, 0) \wedge \tau_2 \prec_\tau \tau_1\right\} \\ =\ & \left\{\mathsf{n}^{j_x} \mid \tau_x = \_, \mathrm{TN}(j_x, 0) \wedge \tau_x \prec_\tau \tau_a \wedge \tau_2 \prec_\tau \tau_x\right\} \end{aligned}$$

  By validity of τ, we know that $\tau_a \prec_\tau \tau_2$. This implies that whenever $\tau_x \prec_\tau \tau_a$ and $\tau_2 \prec_\tau \tau_x$, we have $\tau_x \prec_\tau \tau_2 \prec_\tau \tau_x$. Hence:

$$\mathsf{set\text{-}mac}^4_{\mathrm{ID}}\left(\alpha^{\mathrm{ID}}_{\tau_a}, \kappa^{\mathrm{ID}}_{\tau_a}\right) \cap T = \emptyset \tag{4.75}$$

  Let ψ_0 be ψ in which we replace, for every $\tau_a = \_, \mathrm{TU}_{\mathrm{ID}}(j_a, 1) \prec \tau$, any occurrence of $\mathsf{accept}^{\mathrm{ID}}_{\tau_a}$ and t_{τ_a} by, respectively, $\alpha^{\mathrm{ID}}_{\tau_a}$ and $\kappa^{\mathrm{ID}}_{\tau_a}$, for every τ_a. We then have:

$$\mathsf{set\text{-}mac}^4_{\mathrm{ID}}(\psi_0) = \left\{\mathsf{n}^{j_n} \mid \tau_n = \_, \mathrm{TN}(j_n, 1) \prec \tau\right\} \cup \bigcup_{\substack{\tau_a = \_, \mathrm{TU}_{\mathrm{ID}}(j_a, 1) \\ \tau_a \prec \tau}} \mathsf{set\text{-}mac}^4_{\mathrm{ID}}\left(\alpha^{\mathrm{ID}}_{\tau_a}, \kappa^{\mathrm{ID}}_{\tau_a}\right)$$

  And using (4.75), we know that:

$$\mathsf{set\text{-}mac}^4_{\mathrm{ID}}(\psi_0) \cap T = \left\{\mathsf{n}^{j_n} \mid \tau_n = \_, \mathrm{TN}(j_n, 1) \prec \tau\right\} \tag{4.76}$$

- Let $\tau_n = \_, \mathrm{TN}(j_n, 1)$ and $\tau_n{}' = \_, \mathrm{TN}(j_n, 0)$ such that $\tau_n{}' \prec_\tau \tau_n$. Using **(StrEqu3)**, we know that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau_n} \leftrightarrow \bigvee_{\substack{\tau_i{}' = \_, \mathrm{TU}_{\mathrm{ID}}(j_i, 0) \\ \tau_i = \_, \mathrm{TU}_{\mathrm{ID}}(j_i, 1) \\ \tau_i{}' \prec_\tau \tau_n{}' \prec_\tau \tau_i \prec_\tau \tau_n}} \mathsf{full\text{-}tr}^{\mathsf{n}:\tau_n{}', \tau_n}_{\mathsf{u}:\tau_i{}', \tau_i}$$

  Let $\lambda^{\mathrm{ID}}_{\tau_n}$ be the right-hand side of the equation above. We check that if $\mathsf{n}^{j_n} \in \mathsf{set\text{-}mac}^4_{\mathrm{ID}}(\lambda^{\mathrm{ID}}_{\tau_n})$ then there exists $\tau_i = \_, \mathrm{TU}_{\mathrm{ID}}(j_i, 1)$ such that $\tau_i \prec_\tau \tau_n$. Since $\tau_i \prec \tau$, we know that $j_i \neq j$. Therefore $\tau_i \prec_\tau \tau_2$, and we can show that:

$$\mathsf{set\text{-}mac}^4_{\mathrm{ID}}\left(\lambda^{\mathrm{ID}}_{\tau_n}\right) \cap T = \emptyset \tag{4.77}$$

  Let ψ_1 be ψ_0 in which we replace, for every $\tau_n = \_, \mathrm{TN}(j_n, 1)$ and $\tau_n{}' = \_, \mathrm{TN}(j_n, 0)$ such that $\tau_n{}' \prec_\tau \tau_n$, any occurrence of $\mathsf{accept}^{\mathrm{ID}}_{\tau_n}$ by $\lambda^{\mathrm{ID}}_{\tau_n}$. Using (4.76) and (4.77), we can check that:

$$\mathsf{set\text{-}mac}^4_{\mathrm{ID}}(\psi_1) \cap T = \emptyset$$

  Which is what we wanted to show.

Part 4 Let $\tau_1 = \_, \mathrm{TN}(j_0, 0)$ be such that $\tau_2 \prec_\tau \tau_1$. For every $\tau_1' = \_, \mathrm{TN}(j_0', 0)$ be such that $\tau_2 \prec_\tau \tau_1'$, if $j_0' \neq j_0$ then $(\mathsf{n}^{j_0'} = \mathsf{n}^{j_0}) \leftrightarrow \mathsf{false}$. Moreover, since $\mathsf{set\text{-}mac}^4_{\mathrm{ID}}(\psi_1) \cap T = \emptyset$, we know that for every $\mathsf{n} \in \mathsf{set\text{-}mac}^4_{\mathrm{ID}}(\psi_1)$, $(\mathsf{n} = \mathsf{n}^{j_0}) \leftrightarrow \mathsf{false}$.

We can therefore apply simultaneously the PRF-MAC^4 axiom with key $\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}$ for every $\tau_1 = \_, \mathrm{TN}(j_0, 0)$ such that $\tau_2 \prec_\tau \tau_1$, to replace $\mathsf{Mac}^4_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\mathsf{n}^{j_0})$ by a fresh nonce n_{τ_1}. We then rewrite back ψ_1 into ψ. This yield the derivation:

$$\dfrac{\dfrac{\dfrac{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, (\mathsf{n}_{\tau_1})_{\tau_1 = \_, \mathrm{TN}(j_0,0)\ \tau_2 \prec_\tau \tau_1} \sim \zeta}{\psi_1, (\mathsf{n}_{\tau_1})_{\tau_1 = \_, \mathrm{TN}(j_0,0)\ \tau_2 \prec_\tau \tau_1} \sim \zeta}\, R}{\psi_1, \left(\mathsf{Mac}^4_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\mathsf{n}^{j_0})\right)_{\tau_1 = \_, \mathrm{TN}(j_0,0)\ \tau_2 \prec_\tau \tau_1} \sim \zeta}\, \text{PRF-MAC}^4}{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \left(\mathsf{Mac}^4_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\mathsf{n}^{j_0})\right)_{\tau_1 = \_, \mathrm{TN}(j_0,0)\ \tau_2 \prec_\tau \tau_1} \sim \zeta}\, R$$

where:

$$\zeta \equiv \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \left(\left(\mathsf{Mac}^4_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\mathsf{n}^{j_0})\right)_{\tau_1 = \_, \mathrm{TN}(j_0,0)\ \tau_2 \prec_\tau \tau_1}\right.$$

Observe that we never used the fact that τ was a *basic* trace of actions above, but only the fact that τ is a *valid* trace of actions. Therefore the same reasoning applies to ζ, and for every $\tau_1 = \_, \mathrm{TN}(j_0, 0)$ such that $\tau_2 \prec_\tau \tau_1$, we replace $\mathsf{Mac}^4_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\mathsf{n}^{j_0})$ by a fresh nonce n'_{τ_1}. We conclude using Fresh:

$$\dfrac{\dfrac{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, (\mathsf{n}_{\tau_1})_{\tau_1 = \_, \mathrm{TN}(j_0,0)\ \tau_2 \prec_\tau \tau_1} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, (\mathsf{n}'_{\tau_1})_{\tau_1 = \_, \mathrm{TN}(j_0,0)\ \tau_2 \prec_\tau \tau_1}}\, \mathsf{Fresh}}{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, (\mathsf{n}_{\tau_1})_{\tau_1 = \_, \mathrm{TN}(j_0,0)\ \tau_2 \prec_\tau \tau_1} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \left(\left(\mathsf{Mac}^4_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\mathsf{n}^{j_0})\right)_{\tau_1 = \_, \mathrm{TN}(j_0,0)\ \tau_2 \prec_\tau \tau_1}\right.}\, R + \text{PRF-MAC}^4$$

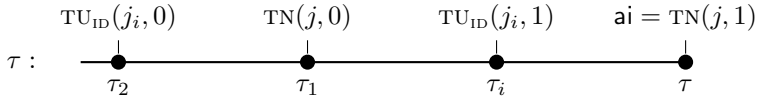
Which concludes this proof.

4.12.11 Case $\mathsf{ai} = \mathrm{TN}(j, 1)$

We know that $\underline{\mathsf{ai}} = \mathrm{TN}(j, 1)$. Here $\mathsf{l\text{-}reveal}_\tau$ and $\mathsf{l\text{-}reveal}_{\tau_0}$ coincides everywhere except on:

$$\mathsf{net\text{-}e\text{-}auth}_\tau(\mathrm{ID}, j) \sim \underline{\mathsf{net\text{-}e\text{-}auth}_{\underline{\tau}}}(\mathrm{ID}, j) \qquad \mathsf{sync\text{-}diff}^{\mathrm{ID}}_\tau \sim \mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}}$$

Let $\mathrm{ID} \in \mathcal{S}_{\mathsf{id}}$, $\tau_i = \_, \mathrm{TU}_{\mathrm{ID}}(j_i, 1)$, $\tau_1 = \_, \mathrm{TN}(j, 0)$, $\tau_2 = \_, \mathrm{TU}_{\mathrm{ID}}(j_i, 0)$ such that $\tau_2 \prec_\tau \tau_1 \prec_\tau \tau_i$:



Let $\mathsf{f} \equiv \mathsf{full\text{-}tr}^{\mathsf{n}:\tau_1,\tau}_{\mathsf{u}:\tau_2,\tau_i}$ and $\underline{\mathsf{f}} \equiv \mathsf{full\text{-}tr}^{\mathsf{n}:\underline{\tau_1},\underline{\tau}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau_i}}$. Using **(Der4)** we know that we have the following derivation:

$$\dfrac{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{f} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \underline{\mathsf{f}}}\, \mathsf{Simp} \tag{4.78}$$

Since $\mathsf{f} \to \mathsf{accept}^{\mathrm{ID}}_\tau$, we have:

$$[\mathsf{f} \wedge \sigma^{\mathsf{in}}_\tau(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_\tau = [\mathsf{f} \wedge \sigma^{\mathsf{in}}_\tau(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})] \begin{pmatrix} \mathsf{if}\ \sigma^{\mathsf{in}}_\tau(\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}) = \mathsf{n}^j\ \mathsf{then}\ \mathsf{suc}(\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau_0}) \\ \mathsf{else}\ \mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau_0} \end{pmatrix}$$

Case 1 Assume that $\tau_i = \_, \mathrm{TU}_{\mathrm{ID}}(j_i, 1) \prec_\tau \mathrm{NS}_{\mathrm{ID}}(\_)$. Let $\tau_{\mathrm{NS}} = \_, \mathrm{NS}_{\mathrm{ID}}(j_{\mathrm{NS}})$ be the latest session reset in τ, i.e. $\tau_{\mathrm{NS}} \prec_\tau \tau$ and $\tau_{\mathrm{NS}} \not\prec_\tau \mathrm{NS}_{\mathrm{ID}}(\_)$. We show by induction that for every τ' such that $\tau_{\mathrm{NS}} \preceq \tau'$ we have:

$$f \wedge \sigma^{\mathsf{in}}_\tau(\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}) = \mathsf{n}^j \ \to\ \sigma_{\tau_{\mathrm{NS}}}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = \sigma_{\tau'}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \tag{4.79}$$

Let τ' be such that $\tau_{\mathrm{NS}} \preceq \tau'$:

- If $\tau' = \tau_{\mathrm{NS}}$ then the property trivially holds.

- If $\tau_{\text{NS}} \prec_\tau \tau'$. The only cases where $\text{SQN}_{\text{N}}^{\text{ID}}$ is updated are $\text{PN}(j',1)$ and $\text{TN}(j',1)$:
  - If $\tau' = \_, \text{PN}(j',1)$: since $\tau = \text{TN}(j,1)$ we know by validity of τ that $j' \neq j$. Therefore:
$$\mathsf{inc\text{-}accept}_{\tau'}^{\text{ID}} \rightarrow \sigma_{\tau'}(\mathsf{session}_{\text{N}}^{\text{ID}}) = \mathsf{n}^{j'} \rightarrow \sigma_{\tau'}(\mathsf{session}_{\text{N}}^{\text{ID}}) \neq \mathsf{n}^{j} \rightarrow \sigma_{\tau}^{\mathsf{in}}(\mathsf{session}_{\text{N}}^{\text{ID}}) \neq \mathsf{n}^{j}$$
    It follows that:
$$\sigma_{\tau}^{\mathsf{in}}(\mathsf{session}_{\text{N}}^{\text{ID}}) = \mathsf{n}^{j} \rightarrow \neg\mathsf{inc\text{-}accept}_{\tau'}^{\text{ID}} \rightarrow \sigma_{\tau'}^{\mathsf{in}}(\text{SQN}_{\text{N}}^{\text{ID}}) = \sigma_{\tau'}(\text{SQN}_{\text{N}}^{\text{ID}})$$
    And we conclude by applying the induction hypothesis.
  - If $\tau' = \_, \text{TN}(j',1)$: since $\tau = \text{TN}(j,1)$ and $\tau' \prec \tau$, we know that $j' \neq j$ (by validity of τ). Therefore:
$$\sigma_{\tau}^{\mathsf{in}}(\mathsf{session}_{\text{N}}^{\text{ID}}) = \mathsf{n}^{j} \rightarrow \neg\mathsf{inc\text{-}accept}_{\tau'}^{\text{ID}} \rightarrow \sigma_{\tau'}^{\mathsf{in}}(\text{SQN}_{\text{N}}^{\text{ID}}) = \sigma_{\tau'}(\text{SQN}_{\text{N}}^{\text{ID}})$$
    And we conclude by applying the induction hypothesis.

This concludes the proof of (4.79). We prove by induction over τ' in $\text{NS}_{\text{ID}}(j_{\text{NS}}) \preceq \tau' \preceq \tau$ that:

$$f \wedge \sigma_{\tau}^{\mathsf{in}}(\mathsf{session}_{\text{N}}^{\text{ID}}) = \mathsf{n}^{j} \rightarrow \neg\sigma_{\tau'}(\mathsf{sync}_{\text{U}}^{\text{ID}}) \tag{4.80}$$

Let ai' be such that $\tau' = \_, \mathsf{ai}'$.

- The case $\mathsf{ai}' = \text{NS}_{\text{ID}}(j_{\text{NS}})$ is trivial since we then have $\sigma_{\tau'}(\mathsf{sync}_{\text{U}}^{\text{ID}}) = \mathsf{false}$.
- If $\mathsf{ai}' \neq \text{PU}_{\text{ID}}(\_,2)$, then since $\text{NS}(j_{\text{NS}}) \not\prec_\tau \text{NS}(\_)$ we know that $\mathsf{ai}' \neq \text{NS}(\_)$. Hence $\sigma_{\tau'}^{\mathsf{up}}(\mathsf{sync}_{\text{U}}^{\text{ID}}) = \bot$, which implies $\sigma_{\tau'}(\mathsf{sync}_{\text{U}}^{\text{ID}}) \equiv \sigma_{\tau'}^{\mathsf{in}}(\mathsf{sync}_{\text{U}}^{\text{ID}})$. By induction hypothesis we know that:
$$f \wedge \sigma_{\tau}^{\mathsf{in}}(\mathsf{session}_{\text{N}}^{\text{ID}}) = \mathsf{n}^{j} \rightarrow \neg\sigma_{\tau'}^{\mathsf{in}}(\mathsf{sync}_{\text{U}}^{\text{ID}})$$
  which concludes this case.
- If $\mathsf{ai}' = \text{PU}_{\text{ID}}(j',2)$. Let $\tau''' = \_, \text{PU}_{\text{ID}}(j',1) \preceq \tau$. By validity of τ we know that $\tau_{\text{NS}} \prec_\tau \tau'''$. Using **(Equ2)** we know that:
$$\mathsf{accept}_{\tau'}^{\text{ID}} \leftrightarrow \bigvee_{\substack{\tau''=\_,\text{PN}(j'',1) \\ \tau''' \prec_\tau \tau'' \prec_\tau \tau'}} \mathsf{supi\text{-}tr}_{\mathsf{u}:\tau'''\!,\tau'}^{\mathsf{n}:\tau''}$$
  And using **(StrEqu4)**:
$$\neg\sigma_{\tau'}^{\mathsf{in}}(\mathsf{sync}_{\text{U}}^{\text{ID}}) \wedge \mathsf{supi\text{-}tr}_{\mathsf{u}:\tau'''\!,\tau'}^{\mathsf{n}:\tau''} \rightarrow \sigma_{\tau'}(\text{SQN}_{\text{U}}^{\text{ID}}) - \sigma_{\tau'}(\text{SQN}_{\text{N}}^{\text{ID}}) = 0$$
  Using (4.79), we know that:
$$f \wedge \sigma_{\tau}^{\mathsf{in}}(\mathsf{session}_{\text{N}}^{\text{ID}}) = \mathsf{n}^{j} \rightarrow \sigma_{\tau_{\text{NS}}}(\text{SQN}_{\text{N}}^{\text{ID}}) = \sigma_{\tau'''}^{\mathsf{in}}(\text{SQN}_{\text{N}}^{\text{ID}}) \wedge \sigma_{\tau_{\text{NS}}}(\text{SQN}_{\text{N}}^{\text{ID}}) = \sigma_{\tau'}(\text{SQN}_{\text{N}}^{\text{ID}})$$
  Therefore:
$$f \wedge \sigma_{\tau}^{\mathsf{in}}(\mathsf{session}_{\text{N}}^{\text{ID}}) = \mathsf{n}^{j} \rightarrow \sigma_{\tau'''}^{\mathsf{in}}(\text{SQN}_{\text{N}}^{\text{ID}}) = \sigma_{\tau'}(\text{SQN}_{\text{N}}^{\text{ID}})$$
  Using **(B5)** we know that $\sigma_{\tau'''}^{\mathsf{in}}(\text{SQN}_{\text{N}}^{\text{ID}}) \leq \sigma_{\tau'''}^{\mathsf{in}}(\text{SQN}_{\text{U}}^{\text{ID}})$, and by **(B1)** we know that $\sigma_{\tau'''}(\text{SQN}_{\text{N}}^{\text{ID}}) \leq \sigma_{\tau'}(\text{SQN}_{\text{U}}^{\text{ID}})$. Moreover $\sigma_{\tau'''}(\text{SQN}_{\text{N}}^{\text{ID}}) = \mathsf{suc}(\sigma_{\tau'''}^{\mathsf{in}}(\text{SQN}_{\text{N}}^{\text{ID}})) < \sigma_{\tau'''}^{\mathsf{in}}(\text{SQN}_{\text{N}}^{\text{ID}})$. We summarize all of this graphically in Figure 4.25. Putting everything together we get that:
$$\mathsf{f} \wedge \neg\sigma_{\tau'}^{\mathsf{in}}(\mathsf{sync}_{\text{U}}^{\text{ID}}) \wedge \mathsf{supi\text{-}tr}_{\mathsf{u}:\tau'''\!,\tau'}^{\mathsf{n}:\tau''} \rightarrow \sigma_{\tau'}(\text{SQN}_{\text{U}}^{\text{ID}}) < \sigma_{\tau'}(\text{SQN}_{\text{U}}^{\text{ID}}) \rightarrow \mathsf{false}$$
  We deduce that:
$$\mathsf{f} \wedge \neg\sigma_{\tau'}^{\mathsf{in}}(\mathsf{sync}_{\text{U}}^{\text{ID}}) \wedge \mathsf{accept}_{\tau'}^{\text{ID}} \rightarrow \bigvee_{\substack{\tau''=\_,\text{PN}(j'',1) \\ \text{PU}_{\text{ID}}(j',1) \prec_\tau \tau'' \prec_\tau \tau'}} \mathsf{f} \wedge \neg\sigma_{\tau'}^{\mathsf{in}}(\mathsf{sync}_{\text{U}}^{\text{ID}}) \wedge \mathsf{supi\text{-}tr}_{\mathsf{u}:\_\!,\tau'}^{\mathsf{n}:\tau''} \rightarrow \mathsf{false}$$
  Moreover, using the induction hypothesis we know that:
$$f \wedge \sigma_{\tau}^{\mathsf{in}}(\mathsf{session}_{\text{N}}^{\text{ID}}) = \mathsf{n}^{j} \rightarrow \neg\sigma_{\tau'}^{\mathsf{in}}(\mathsf{sync}_{\text{U}}^{\text{ID}})$$
  Therefore:
$$f \wedge \sigma_{\tau}^{\mathsf{in}}(\mathsf{session}_{\text{N}}^{\text{ID}}) = \mathsf{n}^{j} \rightarrow \neg\mathsf{accept}_{\tau'} \rightarrow \neg\sigma_{\tau'}(\mathsf{sync}_{\text{U}}^{\text{ID}})$$

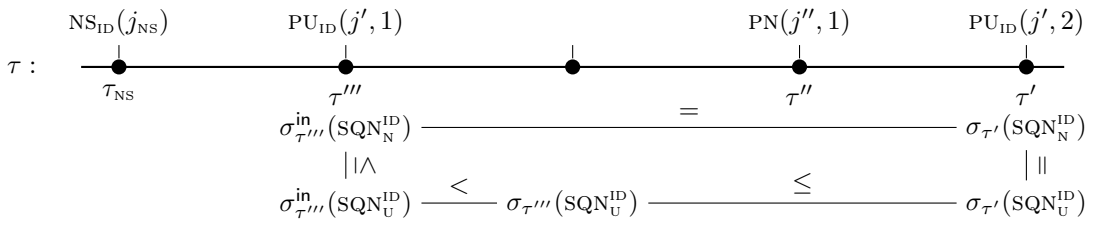


Figure 4.25: First Graphical Representation Used in the Proof of the Case $\mathrm{TN}(j,1)$ of Lemma 4.15.

This concludes the proof of (4.80). Using (4.80) we get that $\mathsf{f} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \rightarrow \sigma^{\mathsf{in}}_{\tau}(\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}) \neq \mathsf{n}^{j}$. Hence:

$$[\mathsf{f}]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} = [\mathsf{f} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau_0}$$

We know that $\underline{\mathsf{f}} \rightarrow \mathsf{accept}^{\nu_{\tau_2}(\mathrm{ID})}_{\underline{\tau}}$. Moreover, $\nu_{\tau_2}(\mathrm{ID}) \neq \nu_{\tau}(\mathrm{ID})$, hence using **(A5)** we know that $\underline{\mathsf{f}} \rightarrow \neg\mathsf{accept}^{\nu_{\tau}(\mathrm{ID})}_{\underline{\tau}}$. Hence:

$$[\underline{\mathsf{f}}]\mathsf{sync\text{-}diff}^{\nu_{\tau}(\mathrm{ID})}_{\underline{\tau}} = [\underline{\mathsf{f}} \wedge \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\nu_{\tau}(\mathrm{ID})}_{\mathrm{U}})]\mathsf{sync\text{-}diff}^{\nu_{\tau}(\mathrm{ID})}_{\underline{\tau_0}}$$

Using the derivation in (4.78) and the fact that:

$$\left(\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}), \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\nu_{\tau}(\mathrm{ID})}_{\mathrm{U}})\right) \in \mathsf{reveal}_{\tau_0} \qquad \left(\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau_0}, \mathsf{sync\text{-}diff}^{\nu_{\tau}(\mathrm{ID})}_{\underline{\tau_0}}\right) \in \mathsf{reveal}_{\tau_0}$$

We can build the derivation:

$$\dfrac{\dfrac{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{f}, \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}), \quad \mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \underline{\mathsf{f}}, \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\nu_{\tau}(\mathrm{ID})}_{\mathrm{U}}), \mathsf{sync\text{-}diff}^{\nu_{\tau}(\mathrm{ID})}_{\underline{\tau_0}}}\ \mathsf{Simp}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, [\mathsf{f}]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, [\underline{\mathsf{f}}]\mathsf{sync\text{-}diff}^{\nu_{\tau}(\mathrm{ID})}_{\underline{\tau}}}\ \mathsf{Simp} \tag{4.81}$$

Case 2 Assume that $\tau_i = \_, \mathrm{TU}_{\mathrm{ID}}(j_i,1) \not\prec_{\tau} \mathrm{NS}_{\mathrm{ID}}(\_)$. We introduce the term θ_{PN} (resp. θ_{TN}) which states that no SUPI (resp. GUTI) network session has been initiated which ID between τ_1 and τ:

$$\theta_{\mathrm{PN}} \equiv \bigwedge_{\substack{\tau' = \_, \mathrm{PN}(\_,1) \\ \tau_1 \prec_{\tau} \tau'}} \neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau'} \qquad \theta_{\mathrm{TN}} \equiv \bigwedge_{\substack{\tau' = \mathrm{TN}(\_,0) \\ \tau_1 \prec_{\tau} \tau'}} \neg\mathsf{accept}^{\mathrm{ID}}_{\tau'}$$

It is easy to show that:

$$\left(\mathsf{f} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}) = \mathsf{n}^{j}\right) \leftrightarrow (\mathsf{f} \wedge \theta_{\mathrm{PN}} \wedge \theta_{\mathrm{TN}})$$

We are now going to show that for every $\tau_1 \preceq \tau'$, $P(\tau')$ holds where $P(\tau')$:

$$P(\tau') \equiv (\mathsf{f} \wedge \theta_{\mathrm{PN}}) \rightarrow \left(\sigma_{\tau'}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) = \mathsf{UnSet} \wedge \sigma_{\tau'}(\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}) = \mathsf{n}^{j} \wedge \bigwedge_{\substack{\tau_1 \prec_{\tau} \tau'' \preceq \tau' \\ \tau'' = \mathrm{TN}(\_,0)}} \neg\mathsf{accept}^{\mathrm{ID}}_{\tau''}\right) \tag{4.82}$$

Since $\mathsf{f} \rightarrow \mathsf{accept}_{\tau_1}$, we know that $\mathsf{f} \rightarrow \sigma_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) = \mathsf{UnSet}$. This shows that $P(\tau_1)$ holds. Let $\tau_1 \prec_{\tau} \tau'$, where $\tau' = \tau'_0, \mathsf{ai}'$, and assume $P(\tau'_0)$ holds by induction. We have four cases:

- If $\mathsf{ai}' \notin \{\mathrm{TN}(\_,0), \mathrm{TN}(\_,1), \mathrm{PN}(\_,1)\}$ then $P(\tau') \equiv P(\tau'_0)$, which concludes this case.
- If $\mathsf{ai}' = \mathrm{TN}(\_,0)$, then using the induction hypothesis $P(\tau'_0)$ we know that $\mathsf{f} \wedge \theta_{\mathrm{PN}} \rightarrow \sigma^{\mathsf{in}}_{\tau'}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) = \mathsf{UnSet}$. Therefore $\mathsf{f} \wedge \theta_{\mathrm{PN}} \rightarrow \neg\mathsf{accept}^{\mathrm{ID}}_{\tau'}$. We know that $\mathsf{f} \wedge \theta_{\mathrm{PN}} \rightarrow \sigma^{\mathsf{in}}_{\tau'}(\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}) = \mathsf{n}^{j}$. We conclude by observing that:

  $$\neg\mathsf{accept}^{\mathrm{ID}}_{\tau'} \wedge \sigma^{\mathsf{in}}_{\tau'}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) = \mathsf{UnSet} \wedge \sigma^{\mathsf{in}}_{\tau'}(\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}) = \mathsf{n}^{j} \rightarrow \left(\sigma_{\tau'}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) = \mathsf{UnSet} \wedge \sigma_{\tau'}(\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}) = \mathsf{n}^{j}\right)$$

- If $\mathsf{ai}' = \mathrm{TN}(j',1)$. Since $\tau' \prec \tau$, we know that $j \neq j'$. Therefore $\sigma^{\mathsf{in}}_{\tau'}(\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}) = \mathsf{n}^{j} \rightarrow \sigma^{\mathsf{in}}_{\tau'}(\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}) \neq \mathsf{n}^{j'}$. We deduce that $\mathsf{f} \wedge \theta_{\mathrm{PN}} \rightarrow \neg\mathsf{accept}^{\mathrm{ID}}_{\tau'}$. This concludes this case.

- If $\mathsf{ai}' = \_, \mathrm{PN}(\_, 1)$. We know that $\mathsf{f} \wedge \theta_{\mathrm{PN}} \rightarrow \neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau'}$. We conclude using the facts that:

$$\neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau'} \rightarrow \sigma_{\tau'}(\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}) = \sigma^{\mathsf{in}}_{\tau'}(\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}) \qquad \neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau'} \rightarrow \sigma_{\tau'}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) = \sigma^{\mathsf{in}}_{\tau'}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}})$$

By applying (4.82) at instant τ_0, we get that:

$$\left(\mathsf{f} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}) = \mathsf{n}^{j}\right) \leftrightarrow (\mathsf{f} \wedge \theta_{\mathrm{PN}} \wedge \theta_{\mathrm{TN}}) \leftrightarrow (\mathsf{f} \wedge \theta_{\mathrm{PN}}) \tag{4.83}$$

Part 1 Let $\tau' = \_, \mathrm{PN}(j', 1)$, with $\tau_1 \prec_\tau \tau'$. Let $\tau'_0 = \mathrm{PN}(j', 0)$. Using **(Equ3)** we know that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau'} \leftrightarrow \bigvee_{\substack{\tau_a = \_, \mathrm{PU_{ID}}(j_a,1) \\ \tau'_0 \prec_\tau \tau_a \prec_\tau \tau'}} \underbrace{\left(\begin{array}{c} g(\phi^{\mathsf{in}}_{\tau_a}) = \mathsf{n}^{j'} \wedge \pi_1(g(\phi^{\mathsf{in}}_{\tau'})) = \{\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_{\tau_a}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{ja}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}} \\ \wedge\ \pi_2(g(\phi^{\mathsf{in}}_{\tau'})) = \mathsf{Mac}^{1}_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle\{\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_{\tau_a}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{ja}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}}\,,\, g(\phi^{\mathsf{in}}_{\tau_a})\rangle)) \end{array}\right)}_{\lambda^{\tau'}_{\tau_a}} \tag{4.84}$$

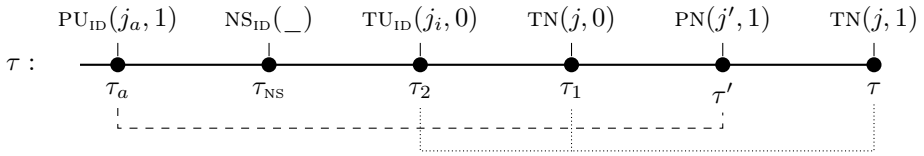
We define:

$$\tau_{\mathrm{NS}} = \begin{cases} \mathrm{NS_{ID}}(j_{\mathrm{NS}}) & \text{if there exists } j_{\mathrm{NS}} \text{ s.t. } \mathrm{NS_{ID}}(j_{\mathrm{NS}}) \prec_\tau \tau \text{ and } \mathrm{NS_{ID}}(j_{\mathrm{NS}}) \not\prec_\tau \mathrm{NS_{ID}}(\_). \\ \epsilon & \text{otherwise} \end{cases}$$

Let $\tau_a = \_, \mathrm{PU_{ID}}(j_a, 1)$ such that $\tau'_0 \prec_\tau \tau_a \prec_\tau \tau'$. Since $\tau_i = \_, \mathrm{TU_{ID}}(j_i, 1) \not\prec_\tau \mathrm{NS_{ID}}(\_)$, we have only three interleavings possible: $\tau_a \prec_\tau \tau_{\mathrm{NS}}$, $\tau_{\mathrm{NS}} \prec_\tau \tau_a \prec_\tau \tau_2$, $\tau_i \prec_\tau \tau_a$. First, we are going to show that in the first two cases we have:

$$\mathsf{f} \wedge \lambda^{\tau'}_{\tau_a} \rightarrow \neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau'}$$

- If $\tau_a \prec_\tau \tau_{\mathrm{NS}}$, we have the following interleaving:



By definition of $\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau'}$, and using the fact that $\lambda^{\tau'}_{\tau_a} \rightarrow \mathsf{accept}^{\mathrm{ID}}_{\tau'}$ we know that:

$$\lambda^{\tau'}_{\tau_a} \wedge \mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau'} \rightarrow \sigma^{\mathsf{in}}_{\tau'}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \leq \sigma^{\mathsf{in}}_{\tau_a}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$$

To conclude this case, we only need to show that:

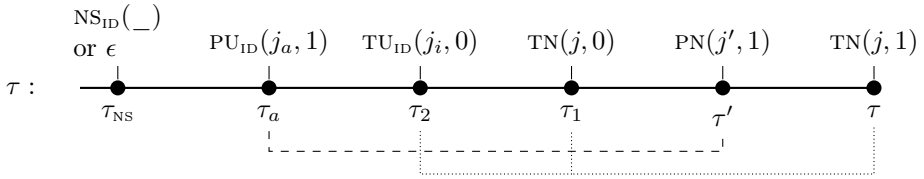
$$\lambda^{\tau'}_{\tau_a} \wedge \mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau'} \rightarrow \sigma^{\mathsf{in}}_{\tau_a}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) < \sigma^{\mathsf{in}}_{\tau'}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \tag{4.85}$$

From which we obtain directly a contradiction, implies that:

$$\mathsf{f} \wedge \lambda^{\tau'}_{\tau_a} \rightarrow \neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau'} \qquad \text{when } \tau_a \prec_\tau \tau_{\mathrm{NS}} \tag{4.86}$$

The proof of (4.85) is by **(B1)** and **(B6)**[9]. We give a graphical representation in Figure 4.26.

- If $\tau_{\mathrm{NS}} \prec_\tau \tau_a \prec_\tau \tau_2$, we have the following interleaving:



We know that $\lambda^{\tau'}_{\tau_a} \rightarrow \sigma_{\tau_a}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{UnSet}$, and that $\mathsf{f} \rightarrow \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}})$. By **(B3)**, we get $\mathsf{f} \rightarrow \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) \neq \mathsf{UnSet}$. This means that $\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}$ is unset at τ_a, but set at τ_2. Therefore there

[9] Using the fact that $\mathsf{f} \rightarrow \sigma^{\mathsf{in}}_{\tau_2}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})$ and $\sigma^{\mathsf{in}}_{\tau_2}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \rightarrow \sigma_{\tau'}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})$

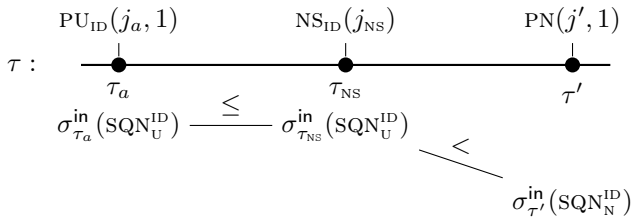


Figure 4.26: Second Graphical Representation Used in the Proof of the Case $\mathrm{TN}(j, 1)$ of Lemma 4.15.

was a successful run of the protocol (SUPI or GUTI) between τ_a and τ_2. More precisely, using Proposition 4.13 we have:

$$\begin{aligned} \mathsf{f} \wedge \lambda^{\tau'}_{\tau_a} &\rightarrow \sigma_{\tau_a}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{UnSet} \wedge \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) \neq \mathsf{UnSet} \\ &\rightarrow \bigvee_{\substack{\tau''=\_,\mathrm{FU}_{\mathrm{ID}}(j'') \\ \tau_a \prec_\tau \tau'' \prec_\tau \tau_2}} \mathsf{accept}^{\mathrm{ID}}_{\tau''} \end{aligned} \tag{4.87}$$

Let $\tau'' = \_, \mathrm{FU}_{\mathrm{ID}}(j'')$ such that $\tau_a \prec_\tau \tau'' \prec_\tau \tau_2$. We then have two cases:

– Assume $j'' = j_a$. In order to have $\mathsf{accept}^{\mathrm{ID}}_{\tau''}$, we need the SUPI or GUTI session j'' to have been executed before τ''. Intuitively, this cannot happen if $j'' = j_a$ because the user session j_a is interacting with the network session j', and $\tau'' \prec_\tau \tau'$. Formally, using the fact that $j'' = j_a$ we are going to show that:

$$\neg\big(\lambda^{\tau'}_{\tau_a} \wedge \mathsf{accept}^{\mathrm{ID}}_{\tau''}\big) \tag{4.88}$$

First, by **(Equ1)** we know that:

$$\begin{aligned} \mathsf{accept}^{\mathrm{ID}}_{\tau''} &\rightarrow \bigvee_{\mathrm{FN}(j_x) \not\prec_{\tau''} \mathrm{NS}_{\mathrm{ID}}(\_)} \mathsf{inj\text{-}auth}_{\tau''}(\mathrm{ID}, j_x) \\ &\rightarrow \bigvee_{\mathrm{FN}(j_x) \not\prec_{\tau''} \mathrm{NS}_{\mathrm{ID}}(\_)} \sigma^{\mathsf{in}}_{\tau''}(\mathsf{b\text{-}auth}^{j_x}_{\mathrm{N}}) = \mathrm{ID} \wedge \sigma^{\mathsf{in}}_{\tau''}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^{j_x} \end{aligned}$$

By **(A8)** we get:

$$\rightarrow \bigvee_{\mathrm{FN}(j_x) \not\prec_{\tau''} \mathrm{NS}_{\mathrm{ID}}(\_)} \sigma^{\mathsf{in}}_{\tau''}(\mathsf{b\text{-}auth}^{j_x}_{\mathrm{N}}) = \mathrm{ID} \wedge \sigma^{\mathsf{in}}_{\tau''}(\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^{j_x} \tag{4.89}$$

We know that $\lambda^{\tau'}_{\tau_a} \rightarrow \sigma^{\mathsf{in}}_{\tau_a}(\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^{j'}$. Moreover, using the validity of τ we know that $\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}$ is not updated between τ_a and τ'', therefore $\lambda^{\tau'}_{\tau_a} \rightarrow \sigma^{\mathsf{in}}_{\tau''}(\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^{j'}$. Putting this together with (4.89), and using the fact that:

$$\big(\sigma^{\mathsf{in}}_{\tau''}(\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^{j_x} \wedge \sigma^{\mathsf{in}}_{\tau''}(\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^{j'}\big) \rightarrow \mathsf{false} \qquad \text{if } j_x \neq j'$$

We get:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau''} \wedge \lambda^{\tau'}_{\tau_a} \rightarrow \sigma^{\mathsf{in}}_{\tau''}(\mathsf{b\text{-}auth}^{j'}_{\mathrm{N}}) = \mathrm{ID} \wedge \sigma^{\mathsf{in}}_{\tau''}(\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^{j'}$$

Since $\tau'' \prec_\tau \tau'$, we know that $\sigma^{\mathsf{in}}_{\tau''}(\mathsf{b\text{-}auth}^{j'}_{\mathrm{N}}) = \mathsf{fail}$. Hence:

$$\rightarrow \sigma^{\mathsf{in}}_{\tau''}(\mathsf{b\text{-}auth}^{j'}_{\mathrm{N}}) = \mathrm{ID} \wedge \sigma^{\mathsf{in}}_{\tau''}(\mathsf{b\text{-}auth}^{j'}_{\mathrm{N}}) = \mathsf{fail} \rightarrow \mathsf{false}$$

Which concludes the proof of (4.88).

– Assume $j'' \neq j_a$. Intuitively, we know that $\mathsf{accept}^{\mathrm{ID}}_{\tau''}$ implies that $\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}$ and $\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}$ have been incremented and synchronized between τ_a and τ'. Therefore we know that the test $\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau'}$ fails. Formally, we show that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau''} \rightarrow \sigma_{\tau_a}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) < \sigma^{\mathsf{in}}_{\tau''}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \tag{4.90}$$

We give the outline of the proof. First, we apply **(StrEqu1)** to τ''. Then, we take $\tau_0'' = \_, \mathrm{FN}(j_e) \prec \tau''$. We let $\tau_1'' = \_, \mathrm{PN}(j_e, 1)$ or $\_, \mathrm{TN}(j_e, 1)$ such that $\tau_1'' \prec \tau_0''$, and we do a case disjunction on τ_1'':

* If $\tau_1'' = \_, \mathrm{PN}(j_e, 1)$, then we use **(StrEqu4)** on it, and we show that $\sigma_{\tau_a}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) < \sigma^{\mathsf{in}}_{\tau''}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}})$ by doing a case disjunction on $\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_1''}$.
* If $\tau_1'' = \_, \mathrm{TN}(j_e, 1)$, then we use **(StrEqu2)** on it, and we show that $\sigma_{\tau_a}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) < \sigma^{\mathsf{in}}_{\tau''}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}})$ using **(B4)**

We omit the details.

Using **(B1)** we know that $\sigma^{\mathsf{in}}_{\tau''}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) \leq \sigma^{\mathsf{in}}_{\tau'}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}})$ and $\sigma^{\mathsf{in}}_{\tau_a}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) \leq \sigma_{\tau_a}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})$. Hence, we deduce from (4.90) that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau''} \;\rightarrow\; \sigma^{\mathsf{in}}_{\tau_a}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}}) < \sigma^{\mathsf{in}}_{\tau'}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}})$$

Moreover, by definition of $\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau'}$, and using the fact that $\lambda^{\tau'}_{\tau_a} \rightarrow \mathsf{accept}^{\mathrm{ID}}_{\tau'}$ we know that:

$$\lambda^{\tau'}_{\tau_a} \wedge \mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau'} \;\rightarrow\; \sigma^{\mathsf{in}}_{\tau'}(\mathrm{SQN}_{\mathrm{N}}^{\mathrm{ID}}) \leq \sigma^{\mathsf{in}}_{\tau_a}(\mathrm{SQN}_{\mathrm{U}}^{\mathrm{ID}})$$

Putting the two equations above together:

$$\lambda^{\tau'}_{\tau_a} \wedge \mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau'} \wedge \mathsf{accept}^{\mathrm{ID}}_{\tau''} \;\rightarrow\; \mathsf{false}$$

Hence:

$$\lambda^{\tau'}_{\tau_a} \wedge \mathsf{accept}^{\mathrm{ID}}_{\tau''} \;\rightarrow\; \neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau'}$$

From (4.87), (4.88) and the equation above, we deduce that:

$$\mathsf{f} \wedge \lambda^{\tau'}_{\tau_a} \;\rightarrow \bigvee_{\substack{\tau''=\_,\mathrm{FU_{ID}}(j'') \\ \tau_a \prec_\tau \tau'' \prec_\tau \tau_2}} \mathsf{f} \wedge \lambda^{\tau'}_{\tau_a} \wedge \mathsf{accept}^{\mathrm{ID}}_{\tau''} \;\rightarrow \bigvee_{\substack{\tau''=\_,\mathrm{FU_{ID}}(j'') \\ \tau_a \prec_\tau \tau'' \prec_\tau \tau_2}} \neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau'}$$

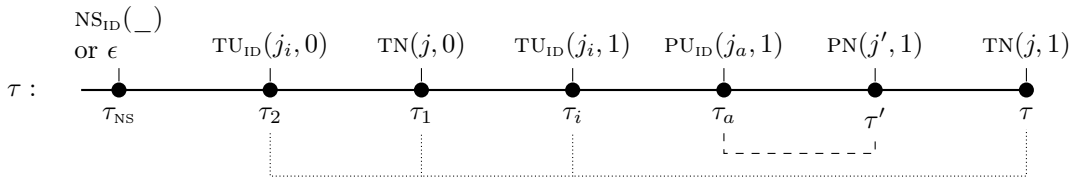
Hence:

$$\mathsf{f} \wedge \lambda^{\tau'}_{\tau_a} \;\rightarrow\; \neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau'} \qquad \text{when } \tau_{\mathrm{NS}} \prec_\tau \tau_a \prec_\tau \tau_2 \tag{4.91}$$

Part 2 Using (4.86) and (4.91), we know that we can focus on the (partial) SUPI sessions that started after τ_i, i.e. the sessions with transcript of the from $\lambda^{\tau'}_{\tau_a}$, where $\tau_a = \_, \mathrm{PU_{ID}}(j_a, 1)$, $\tau' = \_, \mathrm{PN}(j', 1)$ and $\tau_i \prec_\tau \tau_a \prec_\tau \tau'$. Formally, we have:

$$\begin{aligned}
(\mathsf{f} \wedge \theta_{\mathrm{PN}}) \;&\leftrightarrow\; \mathsf{f} \wedge \bigwedge_{\substack{\tau'=_,\mathrm{PN}(_,1) \\ \tau_1 \prec_\tau \tau'}} \neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau'} \\
&\leftrightarrow\; \mathsf{f} \wedge \bigwedge_{\substack{\tau'=_,\mathrm{PN}(_,1) \\ \tau_1 \prec_\tau \tau'}} \mathsf{accept}^{\mathrm{ID}}_{\tau'} \rightarrow \neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau'} \\
&\leftrightarrow\; \mathsf{f} \wedge \bigwedge_{\substack{\tau'=_,\mathrm{PN}(j',1) \\ \tau_0'=_,\mathrm{PN}(j',0) \\ \tau_a=_,\mathrm{PU_{ID}}(j_a,1) \\ \tau_1 \prec_\tau \tau' \\ \tau_0' \prec_\tau \tau_a \prec_\tau \tau'}} \lambda^{\tau'}_{\tau_a} \rightarrow \neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau'} \qquad \text{(By (4.84))} \\
&\leftrightarrow\; \mathsf{f} \wedge \bigwedge_{\substack{\tau_a=_,\mathrm{PU_{ID}}(j_a,1) \\ \tau'=_,\mathrm{PN}(j',1) \\ \tau_i \prec_\tau \tau_a \prec_\tau \tau'}} \lambda^{\tau'}_{\tau_a} \rightarrow \neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau'} \qquad \text{(By (4.86) and (4.91))}
\end{aligned}$$

We represent graphically the shape of the interleavings that we need to consider:



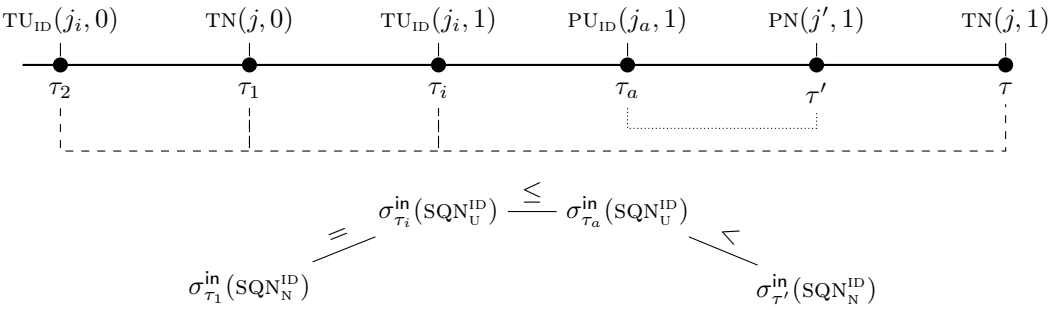


Figure 4.27: Third Graphical Representation Used in the Proof of the Case $\mathrm{TN}(j,1)$ of Lemma 4.15.

Part 3 We are now going to show that if at least one partial SUPI session that started after τ_i accepts (i.e. $\mathsf{f} \wedge \lambda^{\tau'}_{\tau_a}$ holds), then we have $\sigma^{\mathsf{in}}_{\tau}(\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}) \neq \mathsf{n}^j$. First, from what we showed in **Part 2**, and using (4.83) we know that:

$$\neg\left(\mathsf{f} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}) = \mathsf{n}^j\right) \;\leftrightarrow\; \neg\mathsf{f} \vee \bigvee_{\substack{\tau_a = \_,\mathrm{PU}_{\mathrm{ID}}(j_a,1)\\ \tau' = \_,\mathrm{PN}(j',1)\\ \tau_i \prec_\tau \tau_a \prec_\tau \tau'}} \mathsf{f} \wedge \lambda^{\tau'}_{\tau_a} \wedge \mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau'}$$
$$\rightarrow\; \neg\mathsf{f} \vee \bigvee_{\substack{\tau_a = \_,\mathrm{PU}_{\mathrm{ID}}(j_a,1)\\ \tau' = \_,\mathrm{PN}(j',1)\\ \tau_i \prec_\tau \tau_a \prec_\tau \tau'}} \mathsf{f} \wedge \lambda^{\tau'}_{\tau_a}$$

We know show that the converse implication holds. In a first time, assume that for every $\tau_a = \_,\mathrm{PU}_{\mathrm{ID}}(j_a,1)$ and $\tau' = \_,\mathrm{PN}(j',1)$ such that $\tau_i \prec_\tau \tau_a \prec_\tau \tau'$ we have:

$$\mathsf{f} \wedge \lambda^{\tau'}_{\tau_a} \wedge \neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau'} \;\rightarrow\; \sigma^{\mathsf{in}}_{\tau}(\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}) \neq \mathsf{n}^j \tag{4.92}$$

Then we know that:

$$\neg\mathsf{f} \vee \bigvee_{\substack{\tau_a = \_,\mathrm{PU}_{\mathrm{ID}}(j_a,1)\\ \tau' = \_,\mathrm{PN}(j',1)\\ \tau_i \prec_\tau \tau_a \prec_\tau \tau'}} \mathsf{f} \wedge \lambda^{\tau'}_{\tau_a} \;\rightarrow\; \neg\left(\mathsf{f} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}) = \mathsf{n}^j\right)$$

Therefore:

$$\neg\left(\mathsf{f} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}) = \mathsf{n}^j\right) \;\leftrightarrow\; \neg\mathsf{f} \vee \bigvee_{\substack{\tau_a = \_,\mathrm{PU}_{\mathrm{ID}}(j_a,1)\\ \tau' = \_,\mathrm{PN}(j',1)\\ \tau_i \prec_\tau \tau_a \prec_\tau \tau'}} \mathsf{f} \wedge \lambda^{\tau'}_{\tau_a} \tag{4.93}$$

We now give the proof of (4.92). Let $\tau_a = \_,\mathrm{PU}_{\mathrm{ID}}(j_a,1)$ and $\tau' = \_,\mathrm{PN}(j',1)$ such that $\tau_i \prec_\tau \tau_a \prec_\tau \tau'$. We know that:

$$\lambda^{\tau'}_{\tau_a} \wedge \neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau'} \;\rightarrow\; \sigma^{\mathsf{in}}_{\tau_a}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) < \sigma^{\mathsf{in}}_{\tau'}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \qquad \mathsf{f} \;\rightarrow\; \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = \sigma^{\mathsf{in}}_{\tau_i}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$$

Moreover by **(B1)** we know that $\sigma^{\mathsf{in}}_{\tau_i}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) \leq \sigma^{\mathsf{in}}_{\tau_a}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})$. We summarize this graphically in Figure 4.27. We deduce that:

$$\mathsf{f} \wedge \lambda^{\tau'}_{\tau_a} \wedge \neg\mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau'} \;\rightarrow\; \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) < \sigma^{\mathsf{in}}_{\tau'}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$$

Moreover:

$$\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) < \sigma^{\mathsf{in}}_{\tau'}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \;\rightarrow\; \bigvee_{\substack{\tau_x = \mathrm{PN}(j_x,1)\\ \tau_1 \prec_\tau \tau_x \prec_\tau \tau'}} \mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_x} \vee \bigvee_{\substack{\tau_x = \mathrm{TN}(j_x,1)\\ \tau_1 \prec_\tau \tau_x \prec_\tau \tau'}} \mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_x}$$

For every $\tau_x = \mathrm{PN}(j_x,1)$ such that $\tau_1 \prec_\tau \tau_x \prec_\tau \tau'$ we have $j_x \neq j$. Therefore:

$$\bigvee_{\substack{\tau_x = \mathrm{PN}(j_x,1)\\ \tau_1 \prec_\tau \tau_x \prec_\tau \tau'}} \mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_x} \;\rightarrow\; \bigvee_{\substack{\tau_x = \mathrm{PN}(j_x,1)\\ \tau_1 \prec_\tau \tau_x \prec_\tau \tau'}} \sigma_{\tau_x}(\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}) = \mathsf{n}^{j_x} \;\rightarrow\; \sigma^{\mathsf{in}}_{\tau}(\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}) \neq \mathsf{n}^j$$

And:

$$\bigvee_{\substack{\tau_x=\mathrm{TN}(j_x,1)\\ \tau_1\prec_\tau\tau_x\prec_\tau\tau'}} \mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_x} \;\rightarrow\; \bigvee_{\substack{\tau_x=\mathrm{TN}(j_x,1)\\ \tau_1\prec_\tau\tau_x\prec_\tau\tau'}} \sigma^{\mathsf{in}}_{\tau_x}(\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}) = \mathsf{n}^{j_x} \;\rightarrow\; \sigma^{\mathsf{in}}_{\tau}(\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}) \neq \mathsf{n}^{j}$$

This concludes the proof of (4.92).

The proofs in **Part 1** to **3** only used the fact that τ is a valid action trace. We never used the fact that τ is a basic trace. Therefore, carrying out the same proof, we can show that:

$$\neg\big(\underline{\mathsf{f}} \wedge \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{session}^{\nu_\tau(\mathrm{ID})}_{\mathrm{N}}) = \mathsf{n}^{j}\big) \;\leftrightarrow\; \neg\underline{\mathsf{f}} \vee \bigvee_{\substack{\tau_a=\_,\mathrm{PU_{ID}}(j_a,1)\\ \tau'=\_,\mathrm{PN}(j',1)\\ \tau_i\prec_\tau\tau_a\prec_\tau\tau'}} \underline{\mathsf{f}} \wedge \lambda^{\tau'}_{\underline{\tau_a}} \tag{4.94}$$

Part 4 Let $\tau_a = \_, \mathrm{PU_{ID}}(j_a, 1)$ and $\tau' = \_, \mathrm{PN}(j', 1)$ be such that $\tau_i \prec_\tau \tau_a \prec_\tau \tau'$. Observing that:

$$\big(\mathsf{n}^{j'}, \mathsf{n}^{j'}\big) \in \mathsf{reveal}_{\tau_0} \qquad \big(\{\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_{\tau_a}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j_a}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}}, \{\langle \nu_\tau(\mathrm{ID})\,,\, \sigma^{\mathsf{in}}_{\underline{\tau_a}}(\mathrm{SQN}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j_a}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}}\big) \in \mathsf{reveal}_{\tau_0}$$

$$\big(\mathsf{Mac}^1_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle\{\langle \mathrm{ID}\,,\, \sigma^{\mathsf{in}}_{\tau_a}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j_a}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}}, g(\phi^{\mathsf{in}}_{\tau_a})\rangle), \mathsf{Mac}^1_{\mathsf{k}^{\nu_\tau(\mathrm{ID})}_{\mathsf{m}}}(\langle\{\langle \nu_\tau(\mathrm{ID})\,,\, \sigma^{\mathsf{in}}_{\underline{\tau_a}}(\mathrm{SQN}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}})\rangle\}^{\mathsf{n}^{j_a}_{\mathsf{e}}}_{\mathsf{pk}_{\mathrm{N}}}, g(\phi^{\mathsf{in}}_{\underline{\tau_a}})\rangle)\big) \in \mathsf{reveal}_{\tau_0}$$

It is straightforward to show that we have a derivation of:

$$\frac{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \lambda^{\tau'}_{\tau_a} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \lambda^{\tau'}_{\underline{\tau_a}}}\ \mathsf{Simp}$$

Using (4.93) and (4.94), and combining the derivation above with the derivation in (4.78), we can build the following derivation:

$$\frac{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0},}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{f} \wedge \neg \bigvee_{\substack{\tau_a=\_,\mathrm{PU_{ID}}(j_a,1)\\ \tau'=\_,\mathrm{PN}(j',1)\\ \tau_i\prec_\tau\tau_a\prec_\tau\tau'}} \mathsf{f} \wedge \lambda^{\tau'}_{\tau_a} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \underline{\mathsf{f}} \wedge \neg \bigvee_{\substack{\tau_a=\_,\mathrm{PU_{ID}}(j_a,1)\\ \tau'=\_,\mathrm{PN}(j',1)\\ \tau_i\prec_\tau\tau_a\prec_\tau\tau'}} \underline{\mathsf{f}} \wedge \lambda^{\tau'}_{\underline{\tau_a}}}\ (\mathsf{Dup}, \mathsf{FA})^*$$
$$\frac{}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{f} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}) = \mathsf{n}^{j} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \underline{\mathsf{f}} \wedge \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{session}^{\nu_\tau(\mathrm{ID})}_{\mathrm{N}}) = \mathsf{n}^{j}}\ R \tag{4.95}$$

We know that:

$$[\mathsf{f}]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} = \mathsf{if}\ \mathsf{f} \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{session}^{\mathrm{ID}}_{\mathrm{N}}) = \mathsf{n}^{j}\ \mathsf{then}\ \mathsf{suc}(\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau_0})\ \mathsf{else}\ \mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau_0}$$

$$[\underline{\mathsf{f}}]\mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}} = \mathsf{if}\ \underline{\mathsf{f}} \wedge \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}}) \wedge \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{session}^{\nu_\tau(\mathrm{ID})}_{\mathrm{N}}) = \mathsf{n}^{j}\ \mathsf{then}\ \mathsf{suc}(\mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau_0}})\ \mathsf{else}\ \mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau_0}}$$

Hence, using (4.95) and the fact that:

$$\big(\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}), \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\nu_\tau(\mathrm{ID})}_{\mathrm{U}})\big) \in \mathsf{reveal}_{\tau_0} \qquad \big(\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau_0}, \mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau_0}}\big) \in \mathsf{reveal}_{\tau_0}$$

We have a derivation of:

$$\frac{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, [\mathsf{f}]\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, [\underline{\mathsf{f}}]\mathsf{sync\text{-}diff}^{\nu_\tau(\mathrm{ID})}_{\underline{\tau}}}\ \mathsf{Simp} \tag{4.96}$$

Part 5 Using **(StrEqu3)**, we know that:

$$\mathsf{accept}^{\mathrm{ID}}_{\tau} \;\leftrightarrow\; \bigvee_{\substack{\tau_i=\_,\mathrm{TU_{ID}}(j_i,1)\\ \tau_1=\_,\mathrm{TN}(j,0)\\ \tau_2=\_,\mathrm{TU_{ID}}(j_i,0)\\ \tau_2\prec_\tau\tau_1\prec_\tau\tau_i}} \mathsf{full\text{-}tr}^{\mathsf{n}:\tau_1,\tau}_{\mathsf{u}:\tau_2,\tau_i}$$

We split between the cases $\tau_i \prec_\tau \tau_{\text{NS}}$ and $\tau_i \not\prec_\tau \tau_{\text{NS}}$:

$$\leftrightarrow \bigvee_{\substack{\tau_i=\_,\text{TU}_{\text{ID}}(j_i,1)\\ \tau_1=\_,\text{TN}(j,0)\\ \tau_2=\_,\text{TU}_{\text{ID}}(j_i,0)\\ \tau_2\prec_\tau\tau_1\prec_\tau\tau_i\prec_\tau\tau_{\text{NS}}}} \mathsf{full\text{-}tr}^{\mathsf{n}:\tau_1,\tau}_{\mathsf{u}:\tau_2,\tau_i} \vee \bigvee_{\substack{\tau_i=\_,\text{TU}_{\text{ID}}(j_i,1)\\ \tau_1=\_,\text{TN}(j,0)\\ \tau_2=\_,\text{TU}_{\text{ID}}(j_i,0)\\ \tau_{\text{NS}}\prec_\tau\tau_2\prec_\tau\tau_1\prec_\tau\tau_i}} \mathsf{full\text{-}tr}^{\mathsf{n}:\tau_1,\tau}_{\mathsf{u}:\tau_2,\tau_i}$$

If $\tau_i \prec_\tau \tau_{\text{NS}}$ then $\nu_{\tau_2}(\text{ID}) = \nu_{\tau_i}(\text{ID}) \neq \nu_\tau(\text{ID})$, and if $\tau_i \not\prec_\tau \tau_{\text{NS}}$ then $\nu_{\tau_2}(\text{ID}) = \nu_{\tau_i}(\text{ID}) = \nu_\tau(\text{ID})$. It follows, using **(StrEqu3)** on $\underline{\tau}$, that:

$$\bigvee_{\substack{\underline{\text{ID}}\in\mathsf{copies\text{-}id}_C(\text{ID})\\ \underline{\text{ID}}\neq\nu_\tau(\text{ID})}} \mathsf{accept}^{\underline{\text{ID}}}_{\underline{\tau}} \leftrightarrow \bigvee_{\substack{\tau_i=\_,\text{TU}_{\underline{\text{ID}}}(j_i,1)\\ \tau_1=\_,\text{TN}(j,0)\\ \tau_2=\_,\text{TU}_{\underline{\text{ID}}}(j_i,0)\\ \tau_2\prec_\tau\tau_1\prec_\tau\tau_i\prec_\tau\tau_{\text{NS}}}} \mathsf{full\text{-}tr}^{\mathsf{n}:\underline{\tau_1},\underline{\tau}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau_i}} \qquad \mathsf{accept}^{\nu_\tau(\text{ID})}_{\underline{\tau}} \leftrightarrow \bigvee_{\substack{\tau_i=\_,\text{TU}_{\nu_\tau(\text{ID})}(j_i,1)\\ \tau_1=\_,\text{TN}(j,0)\\ \tau_2=\_,\text{TU}_{\nu_\tau(\text{ID})}(j_i,0)\\ \tau_{\text{NS}}\prec_\tau\tau_2\prec_\tau\tau_1\prec_\tau\tau_i}} \mathsf{full\text{-}tr}^{\mathsf{n}:\underline{\tau_1},\underline{\tau}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau_i}}$$

Hence, using (4.81) if $\tau_i \prec_\tau \text{NS}_{\text{ID}}$, and (4.96) if $\tau_i \not\prec_\tau \text{NS}_{\text{ID}}$, we can build the following derivation:

$$\frac{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{sync\text{-}diff}^{\text{ID}}_\tau \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{sync\text{-}diff}^{\nu_\tau(\text{ID})}_{\underline{\tau}}}\ \mathsf{Simp}$$

Part 6 Observe that:

$$\mathsf{net\text{-}e\text{-}auth}_\tau(\text{ID}, j) \leftrightarrow \mathsf{accept}^{\text{ID}}_\tau \qquad \underline{\mathsf{net\text{-}e\text{-}auth}}_{\underline{\tau}}(\text{ID}, j) \leftrightarrow \bigvee_{\underline{\text{ID}}\in\mathsf{copies\text{-}id}_C(\text{ID})} \mathsf{accept}^{\underline{\text{ID}}}_{\underline{\tau}}$$

We therefore easily obtain the derivation:

$$\frac{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{net\text{-}e\text{-}auth}_\tau(\text{ID}, j) \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \underline{\mathsf{net\text{-}e\text{-}auth}}_{\underline{\tau}}(\text{ID}, j)}$$

Finally, we know that:

$$\bigvee_{\text{ID}\in\mathcal{S}_{\mathsf{id}}} \mathsf{accept}^{\text{ID}}_\tau \leftrightarrow \bigvee_{\text{ID}\in\mathcal{S}_{\mathsf{id}}} \mathsf{accept}^{\text{ID}}_\tau \mathsf{net\text{-}e\text{-}auth}_\tau(\text{ID}, j)$$

$$\bigvee_{\text{ID}\in\mathcal{S}_{\mathsf{id}}, \underline{\text{ID}}\in\mathsf{copies\text{-}id}_C(\text{ID})} \mathsf{accept}^{\underline{\text{ID}}}_{\underline{\tau}} \leftrightarrow \bigvee_{\text{ID}\in\mathcal{S}_{\mathsf{id}}} \underline{\mathsf{net\text{-}e\text{-}auth}}_{\underline{\tau}}(\text{ID}, j)$$

It follows that:

$$\dfrac{\dfrac{\dfrac{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \bigvee_{\text{ID}\in\mathcal{S}_{\mathsf{id}}} \mathsf{net\text{-}e\text{-}auth}^{\text{ID}}_\tau \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \bigvee_{\text{ID}\in iddom} \underline{\mathsf{net\text{-}e\text{-}auth}}_{\underline{\tau}}(\text{ID}, j)}\ \mathsf{Simp}}{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \bigvee_{\text{ID}\in\mathcal{S}_{\mathsf{id}}} \mathsf{accept}^{\text{ID}}_\tau \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \bigvee_{\text{ID}\in\mathcal{S}_{\mathsf{id}}, \underline{\text{ID}}\in\mathsf{copies\text{-}id}_C(\text{ID})} \mathsf{accept}^{\underline{\text{ID}}}_{\underline{\tau}}}\ R}{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, t_\tau \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, t_{\underline{\tau}}}\ \mathsf{FA}^*$$

4.13 ⋆ (p. 168) Proof of Proposition 4.20

Proof of ***(Der1)****.* We have two cases:

- either there exists l such that $\text{NS}_{\text{ID}}(l) \prec \tau$ and $\text{NS}_{\text{ID}}(l) \not\prec_\tau \text{NS}_{\text{ID}}(\_)$. In that case we have $\text{NS}_{\text{ID}}(l) \prec_\tau \tau_1$.
- or for every i, $\text{NS}_{\text{ID}}(i) \not\prec_\tau \tau_1$.

Let $\underline{\text{ID}} = \nu_\tau(\text{ID})$. We summarize this graphically in Figure 4.28. In both case, for every $\tau_1 \preceq \tau' \prec \tau$:

$$\big(\sigma_{\tau'}(\text{SQN}^{\text{ID}}_{\text{U}}) - \sigma^{\mathsf{in}}_{\tau'}(\text{SQN}^{\text{ID}}_{\text{U}}), \sigma_{\underline{\tau'}}(\text{SQN}^{\underline{\text{ID}}}_{\text{U}}) - \sigma^{\mathsf{in}}_{\underline{\tau'}}(\text{SQN}^{\underline{\text{ID}}}_{\text{U}})\big) \in \mathsf{reveal}_{\tau_0}$$
$$\big([\sigma^{\mathsf{in}}_\tau(\mathsf{sync}^{\text{ID}}_{\text{U}})]\big(\sigma^{\mathsf{in}}_\tau(\text{SQN}^{\text{ID}}_{\text{N}}) - \sigma^{\mathsf{in}}_\tau(\text{SQN}^{\text{ID}}_{\text{U}})\big), [\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\underline{\text{ID}}}_{\text{U}})]\big(\sigma^{\mathsf{in}}_{\underline{\tau}}(\text{SQN}^{\underline{\text{ID}}}_{\text{N}}) - \sigma^{\mathsf{in}}_{\underline{\tau}}(\text{SQN}^{\underline{\text{ID}}}_{\text{U}})\big)\big) \in \mathsf{reveal}_{\tau_0}$$

We know that:

$$\sigma^{\mathsf{in}}_\tau(\text{SQN}^{\text{ID}}_{\text{U}}) - \sigma^{\mathsf{in}}_{\tau_1}(\text{SQN}^{\text{ID}}_{\text{U}}) = \sigma_{\tau_0}(\text{SQN}^{\text{ID}}_{\text{U}}) - \sigma^{\mathsf{in}}_{\tau_1}(\text{SQN}^{\text{ID}}_{\text{U}}) = \sum_{\tau_1\preceq\tau'} \sigma_{\tau'}(\text{SQN}^{\text{ID}}_{\text{U}}) - \sigma^{\mathsf{in}}_{\tau'}(\text{SQN}^{\text{ID}}_{\text{U}})$$

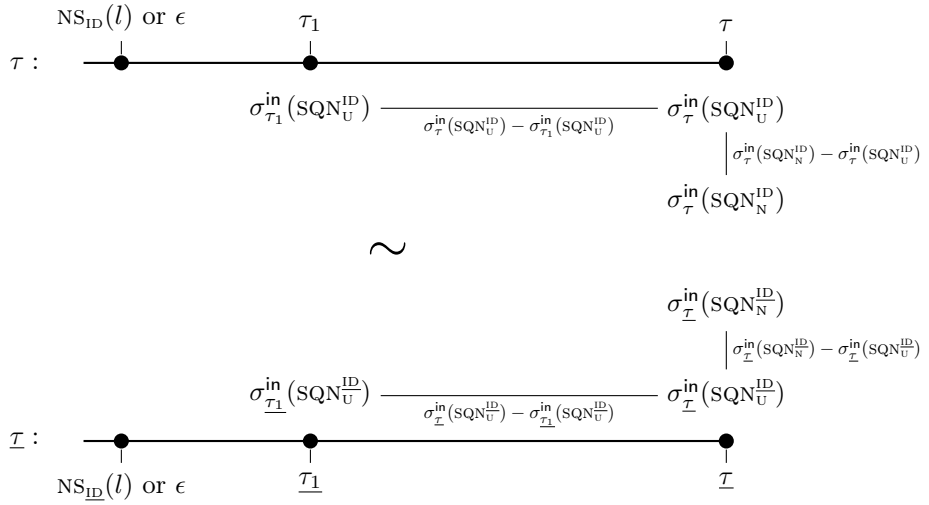


Figure 4.28: First Graphical Representation of the Proof of **(Der1)**

And:

$$\begin{aligned} & \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) < \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) \\ \leftrightarrow \quad & \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \big(\big(\sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) - \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\big) + [\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})]\big(\sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) - \sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\big) < 0\big) \end{aligned}$$

Similarly:

$$\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) - \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) = \sum_{\tau_1 \preceq \tau'} \sigma_{\underline{\tau'}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) - \sigma^{\mathsf{in}}_{\underline{\tau'}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{U}})$$

And:

$$\begin{aligned} & \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{N}}) < \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \\ \leftrightarrow \quad & \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \big(\big(\big(\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) - \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{U}})\big) + [\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}})]\big(\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{N}}) - \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{U}})\big)\big) < 0\big) \end{aligned}$$

Putting everything together, we get:

$$\dfrac{\dfrac{\mathsf{l\text{-}reveal}_{\tau_0} \sim \mathsf{r\text{-}reveal}_{\tau_0}}{\begin{array}{c}\mathsf{l\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}), [\sigma^{\mathsf{in}}_{\tau}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})]\big(\sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) - \sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}})\big), \big(\sigma_{\tau'}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) - \sigma^{\mathsf{in}}_{\tau'}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}),\big)_{\tau_1 \preceq \tau'} \\ \sim \mathsf{r\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}), [\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}})]\big(\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{N}}) - \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{U}})\big), \big(\sigma_{\underline{\tau'}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) - \sigma^{\mathsf{in}}_{\underline{\tau'}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{U}}),\big)_{\tau_1 \preceq \tau'} \end{array}}\ \mathsf{Dup}^*}{\begin{array}{c}\mathsf{l\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_{\tau_1}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) < \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) \\ \sim \mathsf{r\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{N}}) < \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \end{array}}\ \mathsf{Simp}$$

The derivation of (4.20) is very similar. We omit the details, and only give the graphical representation of its proof in Figure 4.29. ■

Proof of ***(Der3)****.* Since τ is valid, we know that for every τ', if $\tau_2 \prec_\tau \tau'$ then $\tau' \neq \mathrm{NS}_{\mathrm{ID}}(\_)$. It follows that $\underline{\tau_2} = \_, \mathrm{TU}_{\nu_\tau(\mathrm{ID})}(j, 0)$ and $\underline{\tau} = \_, \mathrm{TU}_{\nu_\tau(\mathrm{ID})}(j, 1)$. The fact that $\underline{\tau_2} \prec_{\underline{\tau}} \underline{\tau_1}$ is then straightforward. Letting $\underline{\mathrm{ID}} = \nu_\tau(\mathrm{ID})$, we can then check that $\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau}$ and $\mathsf{part\text{-}tr}^{\mathsf{n}:\underline{\tau_1}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}}$ are as described in Figure 4.30.

We have two cases.

Case 1 Assume that for all $\tau' \prec_\tau \tau_1$ such that $\tau' \not\prec_\tau \mathrm{NS}_{\mathrm{ID}}(\_)$ we have $\tau' \neq \_, \mathrm{FU}_{\mathrm{ID}}(\_)$.

Then we know that for all $\underline{\tau'} <_{\underline{\tau}} \underline{\tau_1}$ such that $\underline{\tau'} \not<_{\underline{\tau}} \mathrm{NS}_{\nu_\tau(\mathrm{ID})}(\_)$ we have $\underline{\tau'} \neq \_, \mathrm{FU}_{\nu_\tau(\mathrm{ID})}(\_)$. Therefore using **(B7)** twice we get:

$$\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \rightarrow \mathsf{false} \qquad\qquad \mathsf{part\text{-}tr}^{\mathsf{n}:\underline{\tau_1}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}} \rightarrow \mathsf{false}$$

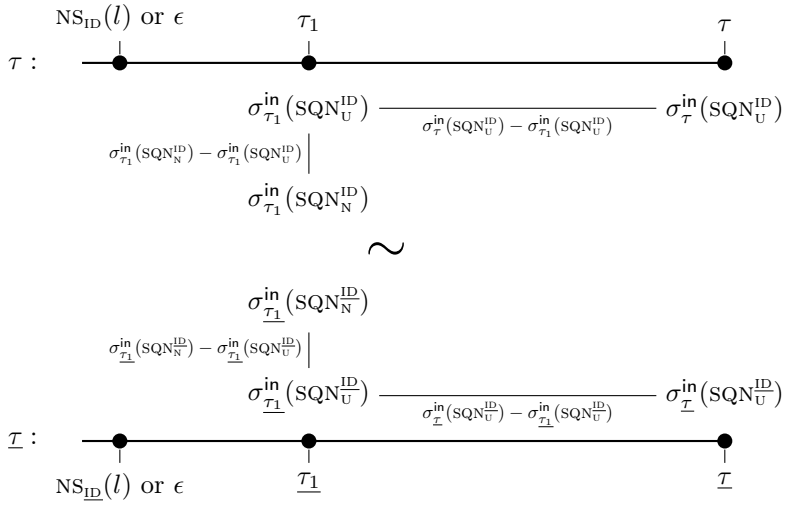


Figure 4.29: Second Graphical Representation for the Proof of **(Der1)**

$$\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \equiv \left(\begin{array}{l} \pi_1(g(\phi^{\mathsf{in}}_{\tau})) = \mathsf{n}^{j_1} \ \wedge \ \pi_2(g(\phi^{\mathsf{in}}_{\tau})) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \oplus \mathsf{f}_{\mathsf{k}^{\mathrm{ID}}}(\mathsf{n}^{j_1}) \\ \wedge \ \pi_3(g(\phi^{\mathsf{in}}_{\tau})) = \mathsf{Mac}^3_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle \mathsf{n}^{j_1}, \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}), \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) \rangle) \\ \wedge \ g(\phi^{\mathsf{in}}_{\tau_1}) = \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) \ \wedge \ \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \ \wedge \ \sigma^{\mathsf{in}}_{\tau_2}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \\ \wedge \ \mathsf{range}(\sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}), \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})) \end{array} \right)$$

$$\mathsf{part\text{-}tr}^{\mathsf{n}:\underline{\tau_1}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}} \equiv \left(\begin{array}{l} \pi_1(g(\phi^{\mathsf{in}}_{\underline{\tau}})) = \mathsf{n}^{j_1} \ \wedge \ \pi_2(g(\phi^{\mathsf{in}}_{\underline{\tau}})) = \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{N}}) \oplus \mathsf{f}_{\mathsf{k}^{\underline{\mathrm{ID}}}}(\mathsf{n}^{j_1}) \\ \wedge \ \pi_3(g(\phi^{\mathsf{in}}_{\underline{\tau}})) = \mathsf{Mac}^3_{\mathsf{k}^{\underline{\mathrm{ID}}}_{\mathsf{m}}}(\langle \mathsf{n}^{j_1}, \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{N}}), \sigma^{\mathsf{in}}_{\underline{\tau_2}}(\mathrm{GUTI}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \rangle) \\ \wedge \ g(\phi^{\mathsf{in}}_{\underline{\tau_1}}) = \sigma^{\mathsf{in}}_{\underline{\tau_2}}(\mathrm{GUTI}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \ \wedge \ \sigma^{\mathsf{in}}_{\underline{\tau_2}}(\mathrm{GUTI}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{GUTI}^{\underline{\mathrm{ID}}}_{\mathrm{N}}) \ \wedge \ \sigma^{\mathsf{in}}_{\underline{\tau_2}}(\mathsf{valid\text{-}guti}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \\ \wedge \ \mathsf{range}(\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{U}}), \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}^{\underline{\mathrm{ID}}}_{\mathrm{N}})) \end{array} \right)$$

Figure 4.30: Terms $\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau}$ and $\mathsf{part\text{-}tr}^{\mathsf{n}:\underline{\tau_1}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}}$ in the Proof of **(Der3)**.

Therefore we have a trivial derivation:

$$\frac{\dfrac{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{false} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{false}}\ \mathsf{FA}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \ \sim \ \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{part\text{-}tr}^{\mathsf{n}:\underline{\tau_1}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}}}\ R \tag{4.97}$$

Case 2 First, we are going to introduce various instants corresponding to previous sessions of the protocol. Eventually, we will be in the situation depicted in Figure 4.31.

Assume that there exists $\tau_3 = \_, \mathrm{FU}_{\mathrm{ID}}(j_0)$ such that $\tau_3 \prec_\tau \tau_1$, $\tau_3 \not\prec_\tau \mathrm{NS}_{\mathrm{ID}}(\_)$ and $\tau_3 \not\prec_\tau \mathrm{FU}_{\mathrm{ID}}(\_)$. Then $\underline{\tau_3} = \_, \mathrm{FU}_{\nu_\tau(\mathrm{ID})}(\_)$, $\underline{\tau_3} <_{\underline{\tau}} \underline{\tau_1}$, $\underline{\tau_3} \not<_{\underline{\tau}} \mathrm{NS}_{\nu_\tau(\mathrm{ID})}(\_)$ and $\underline{\tau_3} \not<_{\underline{\tau}} \mathrm{FU}_{\nu_\tau(\mathrm{ID})}(\_)$.

Assume that $j_0 = j$, then we know that $\tau \prec_\tau \tau_3$, which is absurd. Therefore $j_0 \neq j$. Using the validity of τ, we know that τ_3 cannot occur between $\tau_2 = \_, \mathrm{TU}_{\mathrm{ID}}(j, 0)$ and $\tau = \_, \mathrm{TU}_{\mathrm{ID}}(j, 0)$. Hence $\tau_3 \prec_\tau \tau_2$.

Let τ_{NS} be the latest $\text{NS}_{\text{ID}}(\_)$, if it exists, or ϵ otherwise: $\tau_{\text{NS}} = \_, \text{NS}_{\text{ID}}(\_)$ or ϵ and $\tau_{\text{NS}} \not\prec_\tau \text{NS}_{\text{ID}}(\_)$. Let τ_x be $\_, \text{TU}_{\text{ID}}(j_0, 0)$ or $\_, \text{PU}_{\text{ID}}(j_0, 1)$ be the beginning of the *UE* session associated to τ_3. We know that $\tau_{\text{NS}} \prec_\tau \tau_x \prec_\tau \tau_3$.

We know that $\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \rightarrow \sigma^{\mathsf{in}}_{\tau_2}(\mathsf{valid\text{-}guti}^{\text{ID}}_{\text{U}})$. As $\tau_3 \not\prec_\tau \text{FU}_{\text{ID}}(\_)$, we know that there are no $\text{FU}_{\text{ID}}(\_)$ action between τ_3 and τ_2. If there exists an action by user ID between τ_3 and τ_2, then we have either $\tau_3 \prec_\tau \text{PU}_{\text{ID}}(\_, 1) \prec_\tau \tau_2$ or $\tau_3 \prec_\tau \text{TU}_{\text{ID}}(\_, 0) \prec_\tau \tau_2$. In both case, $\mathsf{valid\text{-}guti}^{\text{ID}}_{\text{U}}$ is set to false, and cannot be set back to something else without a $\text{FU}_{\text{ID}}(\_)$ action. It follows that if there exists a user action between τ_3 and τ_2 then $\neg\sigma^{\mathsf{in}}_{\tau_2}(\mathsf{valid\text{-}guti}^{\text{ID}}_{\text{U}})$. Using the same reasoning we have $\neg\sigma^{\mathsf{in}}_{\underline{\tau_2}}(\mathsf{valid\text{-}guti}^{\underline{\text{ID}}}_{\text{U}})$ if there exists a user action between τ_3 and τ_2. Hence in that case the derivation (4.97) works.

By consequence we now assume that:

$$\{\_, \text{TU}_{\text{ID}}(\_), \_, \text{PU}_{\text{ID}}(\_, \_), \text{FU}_{\text{ID}}(\_)\} \cap \{\tau' \mid \tau_3 \prec_\tau \tau' \prec_\tau \tau_2\} = \emptyset$$

It follows that $\neg\mathsf{accept}^{\text{ID}}_{\tau_3} \rightarrow \neg\sigma^{\mathsf{in}}_{\tau_2}(\mathsf{valid\text{-}guti}^{\text{ID}}_{\text{U}})$, hence $\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \rightarrow \mathsf{accept}^{\text{ID}}_{\tau_3}$. Also, we deduce that $\sigma_{\tau_3}(\text{GUTI}^{\text{ID}}_{\text{U}}) \equiv \sigma^{\mathsf{in}}_{\tau_2}(\text{GUTI}^{\text{ID}}_{\text{U}})$. Applying **(StrEqu1)**, we know that:

$$\mathsf{accept}^{\text{ID}}_{\tau_3} \leftrightarrow \bigvee_{\tau_x \prec_\tau \tau_a = \_, \text{FN}(j_a) \prec_\tau \tau_3} \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3}$$

Therefore:

$$\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \leftrightarrow \bigvee_{\tau_x \prec_\tau \tau_a = \_, \text{FN}(j_a) \prec_\tau \tau_3} \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \wedge \mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau}$$

Similarly, we show that $\sigma_{\underline{\tau_3}}(\text{GUTI}^{\underline{\text{ID}}}_{\text{U}}) \equiv \sigma^{\mathsf{in}}_{\underline{\tau_2}}(\text{GUTI}^{\underline{\text{ID}}}_{\text{U}})$ and that:

$$\mathsf{part\text{-}tr}^{\mathsf{n}:\underline{\tau_1}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}} \leftrightarrow \bigvee_{\tau_x \prec_\tau \tau_a = \_, \text{FN}(j_a) \prec_\tau \tau_3} \mathsf{fu\text{-}tr}^{\mathsf{n}:\underline{\tau_a}}_{\mathsf{u}:\underline{\tau_3}} \wedge \mathsf{part\text{-}tr}^{\mathsf{n}:\underline{\tau_1}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}}$$

We can start building the wanted derivation:

$$\dfrac{\begin{array}{c} \phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \left(\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \wedge \mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau}\right)_{\tau_x \prec_\tau \tau_a = \_, \text{FN}(j_a) \prec_\tau \tau_3} \\ \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \left(\mathsf{fu\text{-}tr}^{\mathsf{n}:\underline{\tau_a}}_{\mathsf{u}:\underline{\tau_3}} \wedge \mathsf{part\text{-}tr}^{\mathsf{n}:\underline{\tau_1}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}}\right)_{\tau_x \prec_\tau \tau_a = \_, \text{FN}(j_a) \prec_\tau \tau_3} \end{array}}{\begin{array}{c} \phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \bigvee_{\tau_x \prec_\tau \tau_a = \_, \text{FN}(j_a) \prec_\tau \tau_3} \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \wedge \mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \\ \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \bigvee_{\tau_x \prec_\tau \tau_a = \_, \text{FN}(j_a) \prec_\tau \tau_3} \mathsf{fu\text{-}tr}^{\mathsf{n}:\underline{\tau_a}}_{\mathsf{u}:\underline{\tau_3}} \wedge \mathsf{part\text{-}tr}^{\mathsf{n}:\underline{\tau_1}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}} \end{array}} \mathsf{FA}^*$$

$$\dfrac{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{part\text{-}tr}^{\mathsf{n}:\underline{\tau_1}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}}}{\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \wedge \mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{fu\text{-}tr}^{\mathsf{n}:\underline{\tau_a}}_{\mathsf{u}:\underline{\tau_3}} \wedge \mathsf{part\text{-}tr}^{\mathsf{n}:\underline{\tau_1}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}}} R$$

Let $\tau_a = \_, \text{FN}(j_a)$ be such that $\tau_x \prec_\tau \tau_a \prec_\tau \tau_3$. Let τ_b be $\_, \text{TN}(j_a, 1)$ or $\_, \text{PN}(j_a, 1)$ such that $\tau_b \prec_\tau \tau_a$. To conclude, we just need to build a derivation of:

$$\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \wedge \mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{fu\text{-}tr}^{\mathsf{n}:\underline{\tau_a}}_{\mathsf{u}:\underline{\tau_3}} \wedge \mathsf{part\text{-}tr}^{\mathsf{n}:\underline{\tau_1}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}}$$

The proof consist in rewriting $\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \wedge \mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau}$ and $\mathsf{fu\text{-}tr}^{\mathsf{n}:\underline{\tau_a}}_{\mathsf{u}:\underline{\tau_3}} \wedge \mathsf{part\text{-}tr}^{\mathsf{n}:\underline{\tau_1}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}}$ such that they can be decomposed (using FA) into corresponding parts appearing in reveal_{τ_0}. We do this piece by piece: the waved underlined part first, the dotted underlined and the dashed underlined part. We represent graphically the protocols executions in Figure 4.31.

Part 1 (Waves) We are going to give a derivation of:

$$\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \wedge \sigma^{\mathsf{in}}_{\tau_2}(\text{GUTI}^{\text{ID}}_{\text{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\text{GUTI}^{\text{ID}}_{\text{N}}) \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{fu\text{-}tr}^{\mathsf{n}:\underline{\tau_a}}_{\mathsf{u}:\underline{\tau_3}} \wedge \sigma^{\mathsf{in}}_{\underline{\tau_2}}(\text{GUTI}^{\underline{\text{ID}}}_{\text{U}}) = \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\text{GUTI}^{\underline{\text{ID}}}_{\text{N}})$$

Recall that $\sigma_{\tau_3}(\text{GUTI}^{\text{ID}}_{\text{U}}) \equiv \sigma^{\mathsf{in}}_{\tau_2}(\text{GUTI}^{\text{ID}}_{\text{U}})$ and $\sigma_{\underline{\tau_3}}(\text{GUTI}^{\underline{\text{ID}}}_{\text{U}}) \equiv \sigma^{\mathsf{in}}_{\underline{\tau_2}}(\text{GUTI}^{\underline{\text{ID}}}_{\text{U}})$. Hence it is sufficient to prove that:

$$\phi^{\mathsf{in}}_\tau, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \wedge \sigma_{\tau_3}(\text{GUTI}^{\text{ID}}_{\text{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\text{GUTI}^{\text{ID}}_{\text{N}}) \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{fu\text{-}tr}^{\mathsf{n}:\underline{\tau_a}}_{\mathsf{u}:\underline{\tau_3}} \wedge \sigma_{\underline{\tau_3}}(\text{GUTI}^{\underline{\text{ID}}}_{\text{U}}) = \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\text{GUTI}^{\underline{\text{ID}}}_{\text{N}})$$

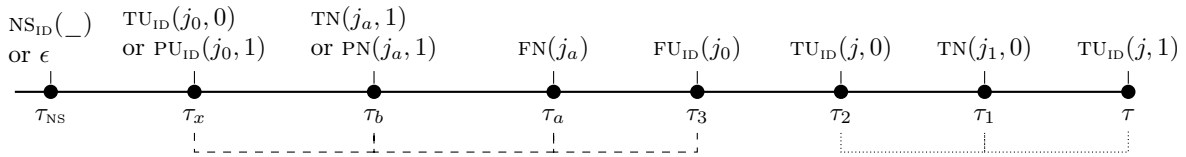


Figure 4.31: Graphical Representation of the Protocol Executions

We know that:

$$[\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3}]\sigma_{\tau_3}(\text{GUTI}^{\text{ID}}_{\text{U}}) = [\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3}]\text{GUTI}^{j_a}$$

Hence:

$$\left(\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \wedge \sigma_{\tau_3}(\text{GUTI}^{\text{ID}}_{\text{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\text{GUTI}^{\text{ID}}_{\text{N}})\right) \leftrightarrow \left(\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \wedge \sigma^{\mathsf{in}}_{\tau_1}(\text{GUTI}^{\text{ID}}_{\text{N}}) = \text{GUTI}^{j_a}\right)$$

Intuitively, the only way we can have $\sigma^{\mathsf{in}}_{\tau_1}(\text{GUTI}^{\text{ID}}_{\text{N}}) = \text{GUTI}^{j_a}$ is:

- if the SUPI or GUTI network session j_a accepts with the increasing sequence number condition.
- and if $\sigma^{\mathsf{in}}_{\tau_1}(\text{GUTI}^{\text{ID}}_{\text{N}})$ was not over-written between τ_b and τ_1.

It is actually straightforward to show by induction that:

$$\sigma^{\mathsf{in}}_{\tau_1}(\text{GUTI}^{\text{ID}}_{\text{N}}) \neq \text{GUTI}^{j_a} \leftrightarrow \left(\neg\mathsf{inc\text{-}accept}^{\text{ID}}_{\tau_b} \vee \bigvee_{\substack{\tau'=\_,\text{TN}(j',1) \\ \text{or } \tau'=\_,\text{PN}(j',1) \\ \tau_b \prec_\tau \tau' \prec_\tau \tau_1}} \mathsf{inc\text{-}accept}^{\text{ID}}_{\tau'} \vee \bigvee_{\substack{\tau'=\_,\text{TN}(j',0) \\ \tau_b \prec_\tau \tau' \prec_\tau \tau_1}} \mathsf{accept}^{\text{ID}}_{\tau'}\right)$$

Hence:

$$\begin{aligned}
& \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \wedge \sigma_{\tau_3}(\text{GUTI}^{\text{ID}}_{\text{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\text{GUTI}^{\text{ID}}_{\text{N}}) \\
\leftrightarrow\ & \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \wedge \mathsf{inc\text{-}accept}^{\text{ID}}_{\tau_b} \wedge \bigwedge_{\substack{\tau'=_,\text{TN}(j',1) \\ \text{or } \tau'=_,\text{PN}(j',1) \\ \tau_b \prec_\tau \tau' \prec_\tau \tau_1}} \neg\mathsf{inc\text{-}accept}^{\text{ID}}_{\tau'} \wedge \bigwedge_{\substack{\tau'=_,\text{TN}(j',0) \\ \tau_b \prec_\tau \tau' \prec_\tau \tau_1}} \neg\mathsf{accept}^{\text{ID}}_{\tau'} \\
\leftrightarrow\ & \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \wedge \mathsf{inc\text{-}accept}^{\text{ID}}_{\tau_b} \wedge \bigwedge_{\substack{\tau'=_,\text{TN}(j',1) \\ \text{or } \tau'=_,\text{PN}(j',1) \\ \tau_b \prec_\tau \tau' \prec_\tau \tau_1}} \neg\mathsf{inc\text{-}accept}^{\text{ID}}_{\tau'} \wedge \bigwedge_{\substack{\tau'=_,\text{TN}(j',0) \\ \tau_b \prec_\tau \tau' \prec_\tau \tau_1}} g(\phi^{\mathsf{in}}_{\tau'}) \neq \text{GUTI}^{j_a}
\end{aligned}$$

For every $\tau_n = \_,\text{TN}(\_,1)$ or $\_,\text{PN}(\_,1)$, we know that $\text{SQN}^{\text{ID}}_{\text{N}}$ is incremented at τ_n if and only if $\mathsf{inc\text{-}accept}^{\text{ID}}_{\tau_n}$ is true. Therefore:

$$\mathsf{inc\text{-}accept}^{\text{ID}}_{\tau_n} \leftrightarrow \sigma^{\mathsf{in}}_{\tau_n}(\text{SQN}^{\text{ID}}_{\text{N}}) < \sigma_{\tau_n}(\text{SQN}^{\text{ID}}_{\text{N}})$$

Using the fact that $\sigma^{\mathsf{in}}_{\tau_n}(\text{SQN}^{\text{ID}}_{\text{U}}) = \sigma_{\tau_n}(\text{SQN}^{\text{ID}}_{\text{U}})$, we can rewrite this as:

$$\mathsf{inc\text{-}accept}^{\text{ID}}_{\tau_n} \leftrightarrow \sigma^{\mathsf{in}}_{\tau_n}(\text{SQN}^{\text{ID}}_{\text{N}}) - \sigma^{\mathsf{in}}_{\tau_n}(\text{SQN}^{\text{ID}}_{\text{U}}) < \sigma_{\tau_n}(\text{SQN}^{\text{ID}}_{\text{N}}) - \sigma_{\tau_n}(\text{SQN}^{\text{ID}}_{\text{U}})$$

Using this remark we can show that:

$$\begin{aligned}
& \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \wedge \sigma_{\tau_3}(\text{GUTI}^{\text{ID}}_{\text{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\text{GUTI}^{\text{ID}}_{\text{N}}) \\
\leftrightarrow\ & \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \wedge \begin{pmatrix} \sigma^{\mathsf{in}}_{\tau_b}(\text{SQN}^{\text{ID}}_{\text{N}}) - \sigma^{\mathsf{in}}_{\tau_b}(\text{SQN}^{\text{ID}}_{\text{U}}) \\ < \sigma_{\tau_b}(\text{SQN}^{\text{ID}}_{\text{N}}) - \sigma_{\tau_b}(\text{SQN}^{\text{ID}}_{\text{U}}) \end{pmatrix} \wedge \begin{pmatrix} \sigma_{\tau_b}(\text{SQN}^{\text{ID}}_{\text{N}}) - \sigma_{\tau_b}(\text{SQN}^{\text{ID}}_{\text{U}}) \\ = \sigma^{\mathsf{in}}_{\tau_1}(\text{SQN}^{\text{ID}}_{\text{N}}) - \sigma^{\mathsf{in}}_{\tau_1}(\text{SQN}^{\text{ID}}_{\text{U}}) \end{pmatrix} \wedge \bigwedge_{\substack{\tau'=_,\text{TN}(j',0) \\ \tau_b \prec_\tau \tau' \prec_\tau \tau_1}} g(\phi^{\mathsf{in}}_{\tau'}) \neq \text{GUTI}^{j_a}
\end{aligned}$$

Doing exactly the same reasoning, we show that:

$$\begin{aligned}
& \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\underline{\tau_3}} \wedge \sigma_{\underline{\tau_3}}(\text{GUTI}^{\text{ID}}_{\text{U}}) = \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\text{GUTI}^{\text{ID}}_{\text{N}}) \\
\leftrightarrow\ & \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\underline{\tau_3}} \wedge \begin{pmatrix} \sigma^{\mathsf{in}}_{\underline{\tau_b}}(\text{SQN}^{\text{ID}}_{\text{N}}) - \sigma^{\mathsf{in}}_{\underline{\tau_b}}(\text{SQN}^{\text{ID}}_{\text{U}}) \\ < \sigma_{\underline{\tau_b}}(\text{SQN}^{\text{ID}}_{\text{N}}) - \sigma_{\underline{\tau_b}}(\text{SQN}^{\text{ID}}_{\text{U}}) \end{pmatrix} \wedge \begin{pmatrix} \sigma_{\underline{\tau_b}}(\text{SQN}^{\text{ID}}_{\text{N}}) - \sigma_{\underline{\tau_b}}(\text{SQN}^{\text{ID}}_{\text{U}}) \\ = \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\text{SQN}^{\text{ID}}_{\text{N}}) - \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\text{SQN}^{\text{ID}}_{\text{U}}) \end{pmatrix} \wedge \bigwedge_{\substack{\tau'=_,\text{TN}(j',0) \\ \tau_b \prec_\tau \tau' \prec_\tau \tau_1}} g(\phi^{\mathsf{in}}_{\underline{\tau'}}) \neq \text{GUTI}^{j_a}
\end{aligned}$$

We introduce some notation that will be used later: for every action trace $\tau = \tau_0, \mathsf{ai}$ and identity ID, we let $\mathsf{sync\text{-}diff\text{-}in}^{\mathrm{ID}}_{\tau} \equiv \mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau_0}$.

We now split the proof in two, depending on whether $\sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})$ is true or false. Let $\psi \equiv \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \wedge \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}})$ and $\underline{\psi} \equiv \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\underline{\tau_3}} \wedge \sigma^{\mathsf{in}}_{\underline{\tau_2}}(\mathrm{GUTI}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{GUTI}^{\underline{\mathrm{ID}}}_{\mathrm{N}})$. Using the fact that:

$$\left(\sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}), \sigma^{\mathsf{in}}_{\underline{\tau_b}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}})\right) \in \mathsf{reveal}_{\tau_0}$$

We can build the derivation:

$$\dfrac{\dfrac{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \psi, \neg\sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \psi \;\sim\; \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_{\underline{\tau_b}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \underline{\psi}, \neg\sigma^{\mathsf{in}}_{\underline{\tau_b}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \underline{\psi}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}), \sigma^{\mathsf{in}}_{\underline{\tau_b}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \underline{\psi}, \neg\sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \psi \;\sim\; \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_{\underline{\tau_b}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}), \sigma^{\mathsf{in}}_{\underline{\tau_b}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \underline{\psi}, \neg\sigma^{\mathsf{in}}_{\underline{\tau_b}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \underline{\psi}}\,\mathsf{Dup}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \psi \;\sim\; \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \underline{\psi}}\,\mathsf{Simp}$$

We now build a derivation of $\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \psi$ and of $\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \neg\sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \psi$:

- Using the fact that we have $\sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})$,we know that:

$$\begin{aligned}
&\sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \wedge \sigma_{\tau_3}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \\
\leftrightarrow\; &\sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \wedge \begin{pmatrix} & \mathsf{sync\text{-}diff\text{-}in}^{\mathrm{ID}}_{\tau_b} \\ < & \mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau_b} \end{pmatrix} \wedge \begin{pmatrix} & \mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau_b} \\ = & \mathsf{sync\text{-}diff\text{-}in}^{\mathrm{ID}}_{\tau_1} \end{pmatrix} \wedge \bigwedge_{\substack{\tau' = _, \mathrm{TN}(j',0) \\ \tau_b \prec_\tau \tau' \prec_\tau \tau_1}} g(\phi^{\mathsf{in}}_{\tau'}) \neq \mathrm{GUTI}^{j_a}
\end{aligned}$$

  Similarly we get that:

$$\begin{aligned}
&\sigma^{\mathsf{in}}_{\underline{\tau_b}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\underline{\tau_3}} \wedge \sigma_{\underline{\tau_3}}(\mathrm{GUTI}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{GUTI}^{\underline{\mathrm{ID}}}_{\mathrm{N}}) \\
\leftrightarrow\; &\sigma^{\mathsf{in}}_{\underline{\tau_b}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\underline{\tau_3}} \wedge \begin{pmatrix} & \mathsf{sync\text{-}diff\text{-}in}^{\underline{\mathrm{ID}}}_{\underline{\tau_b}} \\ < & \mathsf{sync\text{-}diff}^{\underline{\mathrm{ID}}}_{\underline{\tau_b}} \end{pmatrix} \wedge \begin{pmatrix} & \mathsf{sync\text{-}diff}^{\underline{\mathrm{ID}}}_{\underline{\tau_b}} \\ = & \mathsf{sync\text{-}diff\text{-}in}^{\underline{\mathrm{ID}}}_{\underline{\tau_1}} \end{pmatrix} \wedge \bigwedge_{\substack{\tau' = _, \mathrm{TN}(j',0) \\ \tau_b \prec_\tau \tau' \prec_\tau \tau_1}} g(\phi^{\mathsf{in}}_{\tau'}) \neq \mathrm{GUTI}^{j_a}
\end{aligned}$$

  Moreover, we know that:

$$\left(\left(\mathrm{GUTI}^{j_a}, \mathrm{GUTI}^{j_a}\right) \in \mathsf{reveal}_{\tau_0}\right)_{\substack{\tau' = \_, \mathrm{TN}(j',0) \\ \tau_b \prec_\tau \tau' \prec_\tau \tau_1}} \qquad \left(\mathsf{sync\text{-}diff\text{-}in}^{\mathrm{ID}}_{\tau_1}, \mathsf{sync\text{-}diff\text{-}in}^{\underline{\mathrm{ID}}}_{\underline{\tau_1}}\right) \in \mathsf{reveal}_{\tau_0}$$

$$\left(\mathsf{sync\text{-}diff\text{-}in}^{\mathrm{ID}}_{\tau_b}, \mathsf{sync\text{-}diff\text{-}in}^{\underline{\mathrm{ID}}}_{\underline{\tau_b}}\right) \in \mathsf{reveal}_{\tau_0} \qquad \left(\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau_b}, \mathsf{sync\text{-}diff}^{\underline{\mathrm{ID}}}_{\underline{\tau_b}}\right) \in \mathsf{reveal}_{\tau_0}$$

$$\left(\sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}), \sigma^{\mathsf{in}}_{\underline{\tau_b}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}})\right) \in \mathsf{reveal}_{\tau_0}$$

  And using **(Der2)**, we know that we have a derivation of:

$$\dfrac{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3}}\,\mathsf{Simp}$$

  Using this, we can rewrite $\sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \psi$ and $\sigma^{\mathsf{in}}_{\underline{\tau_b}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \underline{\psi}$ as two terms that decompose, using FA, into matching part of reveal_{τ_0}. By consequence we can build the following derivation:

$$\dfrac{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}\psi \;\sim\; \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \psi \;\sim\; \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_{\underline{\tau_b}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \underline{\psi}}\,\mathsf{Simp} \tag{4.98}$$

- We now focus on the case where we have $\neg\sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})$.
  First, assume that $\tau_b = \_, \mathrm{TN}(j_a, 1)$. In that case, we know that $\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \rightarrow \mathsf{accept}^{\mathrm{ID}}_{\tau_b}$. Since $\mathsf{accept}^{\mathrm{ID}}_{\tau_b} \rightarrow \sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})$, we get that $(\neg\sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \psi) \leftrightarrow \mathsf{false}$. Similarly we have $(\neg\sigma^{\mathsf{in}}_{\underline{\tau_b}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \underline{\psi}) \leftrightarrow \mathsf{false}$. By consequence, we have a trivial derivation:

$$\dfrac{\dfrac{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0} \;\sim\; \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{false} \;\sim\; \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{false}}\,\mathsf{FA}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \neg\sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \psi \;\sim\; \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \neg\sigma^{\mathsf{in}}_{\underline{\tau_b}}(\mathsf{sync}^{\underline{\mathrm{ID}}}_{\mathrm{U}}) \wedge \underline{\psi}}\,\mathsf{Simp}$$

Now assume that $\tau_b = \_, \mathrm{PN}(j_a, 1)$. Since $\tau_3 = \_, \mathrm{FU}_{\mathrm{ID}}(j_0) \prec \tau$, we know by validity of τ there there exists $\tau' = \_, \mathrm{PU}_{\mathrm{ID}}(j_0, 2)$ or $\_, \mathrm{TU}_{\mathrm{ID}}(j_0, 1)$ such that $\tau' \prec_\tau \tau_3$. It is straightforward to check that if $\tau' = \_, \mathrm{TU}_{\mathrm{ID}}(j_0, 1)$ then since $\tau_b = \_, \mathrm{PN}(j_a, 1)$ we have $\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \leftrightarrow \mathsf{false}$ and $\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\underline{\tau_3}} \leftrightarrow \mathsf{false}$. Building the wanted derivation is then trivial.

Therefore assume that $\tau' = \_, \mathrm{PU}_{\mathrm{ID}}(j_0, 2)$. Observe that $\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \rightarrow \mathsf{accept}^{\mathrm{ID}}_{\tau'}$. We have two cases:

– Assume $\tau' \prec_\tau \tau_b$. Using **(Equ2)**, we know that:

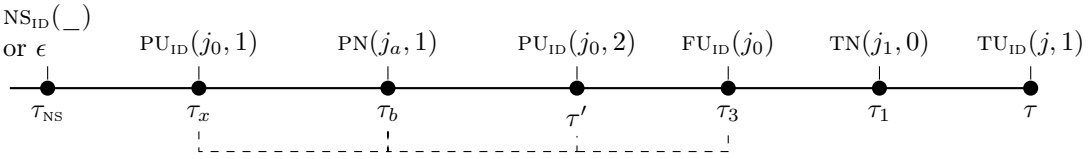
$$\begin{aligned} \mathsf{accept}^{\mathrm{ID}}_{\tau'} \;&\rightarrow\; \bigvee_{\substack{\tau_n = \_, \mathrm{PN}(j_n, 1) \\ \tau_x \prec_\tau \tau_n \prec_\tau \tau'}} \sigma^{\mathrm{ID}}_{\tau_x}(\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^{j_n} \\ &\rightarrow\; \sigma^{\mathrm{ID}}_{\tau_x}(\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) \neq \mathsf{n}^{j_a} \qquad \text{(Since } \tau' \prec_\tau \tau_b\text{)} \end{aligned}$$

Moreover:

$$\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \rightarrow \sigma^{\mathrm{ID}}_{\tau'}(\mathsf{e\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^{j_a} \rightarrow \sigma^{\mathrm{ID}}_{\tau_x}(\mathsf{b\text{-}auth}^{\mathrm{ID}}_{\mathrm{U}}) = \mathsf{n}^{j_a}$$

Therefore $\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \rightarrow \mathsf{false}$. Similarly $\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\underline{\tau_3}} \rightarrow \mathsf{false}$. Hence we have a trivial derivation.

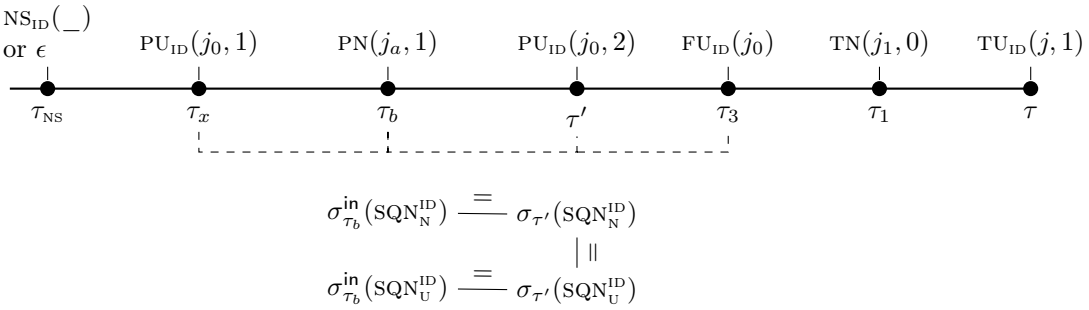
– Assume $\tau_b \prec_\tau \tau'$. We summarize graphically the situation below:



First, since there are no ID actions between τ_b and τ', we know that $\neg\sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \rightarrow \neg\sigma^{\mathsf{in}}_{\tau'}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})$. Recall that $\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \rightarrow \mathsf{accept}^{\mathrm{ID}}_{\tau'}$. Using **(Equ2)**, it is simple to check that $\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \wedge \mathsf{accept}^{\mathrm{ID}}_{\tau'} \rightarrow \mathsf{supi\text{-}tr}^{\mathsf{n}:\tau_b}_{\mathsf{u}:\tau_x,\tau'}$. Therefore:

$$\begin{aligned} \neg\sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \;&\rightarrow\; \neg\sigma^{\mathsf{in}}_{\tau'}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{accept}^{\mathrm{ID}}_{\tau'} \\ &\rightarrow\; \mathsf{inc\text{-}accept}^{\mathrm{ID}}_{\tau_b} \begin{array}{l} \wedge\, \sigma^{\mathsf{in}}_{\tau'}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) = \sigma_{\tau_b}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \\ \wedge\, \sigma_{\tau'}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma_{\tau'}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \end{array} \qquad \text{(By } \textbf{(StrEqu4)}\text{)} \end{aligned}$$

Using again the fact that there are no ID actions between τ_b and τ', we know that $\sigma^{\mathsf{in}}_{\tau_b}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) \equiv \sigma^{\mathsf{in}}_{\tau'}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})$. Moreover $\sigma^{\mathsf{in}}_{\tau'}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \equiv \sigma_{\tau'}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})$, therefore $\sigma^{\mathsf{in}}_{\tau_b}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma_{\tau'}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})$. Similarly, we know that $\sigma_{\tau'}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \equiv \sigma^{\mathsf{in}}_{\tau'}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})$. Summarizing:



Therefore we get that:

$$\begin{aligned} &\neg\sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \wedge \sigma_{\tau_3}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \\ \leftrightarrow\; &\neg\sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \wedge \left(\begin{array}{r} \sigma_{\tau'}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) - \sigma_{\tau'}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) \\ = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) - \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) \end{array} \right) \wedge \bigwedge_{\substack{\tau' = \_, \mathrm{TN}(j', 0) \\ \tau_b \prec_\tau \tau' \prec_\tau \tau_1}} g(\phi^{\mathsf{in}}_{\tau'}) \neq \mathrm{GUTI}^{j_a} \end{aligned}$$

Besides, $\mathsf{accept}^{\mathrm{ID}}_{\tau'} \rightarrow \sigma_{\tau'}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})$. Since $\tau' \prec_\tau \tau_1$ we know that $\sigma_{\tau'}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \rightarrow \sigma^{\mathsf{in}}_{\tau_1}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})$. Hence:

$$\begin{aligned} & \neg\sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \wedge \sigma_{\tau_3}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \\ \leftrightarrow\ & \neg\sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \wedge \mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau'} = \mathsf{sync\text{-}diff\text{-}in}^{\mathrm{ID}}_{\tau_1} \wedge \bigwedge_{\substack{\tau'=\_,\mathrm{TN}(j',0)\\ \tau_b \prec_\tau \tau' \prec_\tau \tau_1}} g(\phi^{\mathsf{in}}_{\tau'}) \neq \mathrm{GUTI}^{j_a} \end{aligned}$$

Similarly:

$$\begin{aligned} & \neg\sigma^{\mathsf{in}}_{\underline{\tau_b}}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{fu\text{-}tr}^{\mathsf{n}:\underline{\tau_a}}_{\mathsf{u}:\underline{\tau_3}} \wedge \sigma_{\underline{\tau_3}}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \\ \leftrightarrow\ & \neg\sigma^{\mathsf{in}}_{\underline{\tau_b}}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \mathsf{fu\text{-}tr}^{\mathsf{n}:\underline{\tau_a}}_{\mathsf{u}:\underline{\tau_3}} \wedge \mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\underline{\tau'}} = \mathsf{sync\text{-}diff\text{-}in}^{\mathrm{ID}}_{\underline{\tau_1}} \wedge \bigwedge_{\substack{\tau'=\_,\mathrm{TN}(j',0)\\ \tau_b \prec_\tau \tau' \prec_\tau \tau_1}} g(\phi^{\mathsf{in}}_{\underline{\tau'}}) \neq \mathrm{GUTI}^{j_a} \end{aligned}$$

And using **(Der2)**, we know that we have a derivation of:

$$\frac{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_a}_{\mathsf{u}:\tau_3}}\ \mathsf{Simp}$$

Moreover, we know that:

$$\left((\mathrm{GUTI}^{j_a}, \mathrm{GUTI}^{j_a}) \in \mathsf{reveal}_{\tau_0}\right)_{\substack{\tau'=\_,\mathrm{TN}(j',0)\\ \tau_b \prec_\tau \tau' \prec_\tau \tau_1}} \qquad \left(\mathsf{sync\text{-}diff\text{-}in}^{\mathrm{ID}}_{\tau_1}, \mathsf{sync\text{-}diff\text{-}in}^{\mathrm{ID}}_{\underline{\tau_1}}\right) \in \mathsf{reveal}_{\tau_0}$$

$$\left(\mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\tau}, \mathsf{sync\text{-}diff}^{\mathrm{ID}}_{\underline{\tau'}}\right) \in \mathsf{reveal}_{\tau_0} \qquad \left(\sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}), \sigma^{\mathsf{in}}_{\underline{\tau_b}}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})\right) \in \mathsf{reveal}_{\tau_0}$$

Similarly to what we did in (4.98), we can rewrite $\neg\sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \psi$ and $\neg\sigma^{\mathsf{in}}_{\underline{\tau_b}}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \underline{\psi}$ as two terms that decompose, using FA, into matching part of reveal_{τ_0}. By consequence we can build the following derivation:

$$\frac{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}\psi \ \sim\ \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \neg\sigma^{\mathsf{in}}_{\tau_b}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \psi \ \sim\ \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \neg\sigma^{\mathsf{in}}_{\underline{\tau_b}}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \underline{\psi}}\ \mathsf{Simp}$$

Part 2 (Dots) Using **(StrEqu2)** we know that $\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \rightarrow \mathsf{accept}^{\mathrm{ID}}_{\tau_1}$. Therefore, using **(A6)**, $\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \rightarrow \neg\mathsf{accept}^{\mathrm{ID}'}_{\tau_1}$ for every $\mathrm{ID}' \neq \mathrm{ID}$. It follows that $\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \rightarrow t_{\tau_1} = \mathsf{msg}^{\mathrm{ID}}_{\tau_1}$, and therefore:

$$\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \ \rightarrow\ \pi_2(t_{\tau_1}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \oplus \mathsf{f}_{\mathsf{k}^{\mathrm{ID}}}(\mathsf{n}^{j_1})$$

And:

$$\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \ \rightarrow\ \pi_3(t_{\tau_1}) = \mathsf{Mac}^3_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle \mathsf{n}^{j_1}, \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}), \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}})\rangle)$$

Since no action from agent ID occurs between τ_2 and τ_1, we know that $\sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}})$. Hence:

$$\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \ \rightarrow\ \pi_3(t_{\tau_1}) = \mathsf{Mac}^3_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\langle \mathsf{n}^{j_1}, \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}), \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}})\rangle)$$

Hence we can rewrite $\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau}$ as follows:

$$\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} = \left(\begin{array}{l} \pi_1(g(\phi^{\mathsf{in}}_{\tau})) = \mathsf{n}^{j_1} \ \wedge\ \pi_2(g(\phi^{\mathsf{in}}_{\tau})) = \pi_2(t_{\tau_1}) \ \wedge\ \pi_3(g(\phi^{\mathsf{in}}_{\tau})) = \pi_3(t_{\tau_1}) \\ \wedge\ g(\phi^{\mathsf{in}}_{\tau_1}) = \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) \ \wedge\ \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \ \wedge\ \sigma^{\mathsf{in}}_{\tau_2}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \\ \wedge\ \mathsf{range}(\sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}), \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})) \end{array}\right)$$

By a similar reasoning we rewrite $\mathsf{part\text{-}tr}^{\mathsf{n}:\underline{\tau_1}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}}$ as follows:

$$\mathsf{part\text{-}tr}^{\mathsf{n}:\underline{\tau_1}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}} = \left(\begin{array}{l} \pi_1(g(\phi^{\mathsf{in}}_{\underline{\tau}})) = \mathsf{n}^{j_1} \ \wedge\ \pi_2(g(\phi^{\mathsf{in}}_{\underline{\tau}})) = \pi_2(t_{\tau_1}) \ \wedge\ \pi_3(g(\phi^{\mathsf{in}}_{\underline{\tau}})) = \pi_3(t_{\tau_1}) \\ \wedge\ g(\phi^{\mathsf{in}}_{\underline{\tau_1}}) = \sigma^{\mathsf{in}}_{\underline{\tau_2}}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) \ \wedge\ \sigma^{\mathsf{in}}_{\underline{\tau_2}}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{N}}) \ \wedge\ \sigma^{\mathsf{in}}_{\underline{\tau_2}}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \\ \wedge\ \mathsf{range}(\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}), \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})) \end{array}\right)$$

Part 3 (Dash) Since $\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \to \sigma^{\mathsf{in}}_{\tau_2}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}})$ we know that:

$$\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \;\to\; \sigma^{\mathsf{in}}_{\tau_2}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) \;=\; \mathsf{m\text{-}suci}^{\,\mathrm{ID}}_{\tau}$$

Besides, as $\sigma^{\mathsf{in}}_{\tau_2}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \to \sigma^{\mathsf{in}}_{\tau_2}(\mathsf{sync}^{\mathrm{ID}}_{\mathrm{U}})$, and since $\sigma^{\mathsf{in}}_{\tau_2}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \to \sigma^{\mathsf{in}}_{\tau_1}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}})$ (because $\tau_2 \prec_\tau \tau_1$ and $\tau_2 \not\prec_\tau \mathrm{NS}_{\mathrm{ID}}(\_)$), we know that:

$$\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \to \big(\mathsf{range}(\sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}), \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})) \leftrightarrow \big(\sigma^{\mathsf{in}}_{\tau_1}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})\big)\big)$$

Similarly we have:

$$\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}} \;\to\; \sigma^{\mathsf{in}}_{\underline{\tau_2}}(\mathrm{GUTI}^{\mathrm{ID}}_{\mathrm{U}}) \;=\; \mathsf{m\text{-}suci}^{\,\mathrm{ID}}_{\underline{\tau}}$$

$$\mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}} \to \Big(\mathsf{range}(\sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}), \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})) \leftrightarrow \Big(\sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})\Big)\Big)$$

Moreover:

$$\big(\mathsf{m\text{-}suci}^{\,\mathrm{ID}}_{\tau} \;\sim\; \mathsf{m\text{-}suci}^{\,\mathrm{ID}}_{\underline{\tau}}\big) \in \mathsf{reveal}_{\tau_0} \qquad \big(\sigma^{\mathsf{in}}_{\tau_2}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \;\sim\; \sigma^{\mathsf{in}}_{\underline{\tau_2}}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}})\big) \in \mathsf{reveal}_{\tau_0}$$

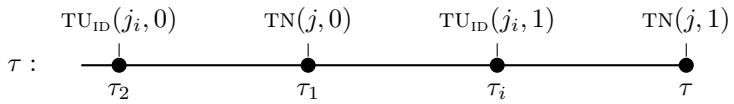
Finally, using **(Der1)**, we know that we have a derivation of:

$$\frac{\mathsf{l\text{-}reveal}_{\tau_0} \;\sim\; \mathsf{r\text{-}reveal}_{\tau_0}}{\begin{array}{l}\mathsf{l\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_{\tau_1}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \sigma^{\mathsf{in}}_{\tau}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\tau_1}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}}) \\ \sim\; \mathsf{r\text{-}reveal}_{\tau_0}, \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathsf{valid\text{-}guti}^{\mathrm{ID}}_{\mathrm{U}}) \wedge \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{U}}) = \sigma^{\mathsf{in}}_{\underline{\tau_1}}(\mathrm{SQN}^{\mathrm{ID}}_{\mathrm{N}})\end{array}}\ \mathsf{Simp}$$

Part 4 (conclusion) To conclude, we combine the derivations of Part 1, Part 2 and Part 3. ■

*Proof of **(Der4)**.* Recall that:

$$\mathsf{full\text{-}tr}^{\mathsf{n}:\tau_1,\tau}_{\mathsf{u}:\tau_2,\tau_i} \;\equiv\; \mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau_i} \wedge g(\phi^{\mathsf{in}}_{\tau}) = \mathsf{Mac}^4_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\mathsf{n}^j)$$



The fact that $\underline{\tau_2} = \_, \mathrm{TU}_{\nu_{\tau_1}(\mathrm{ID})}(j_i,0)$, $\underline{\tau_i} = \_, \mathrm{TU}_{\nu_{\tau_1}(\mathrm{ID})}(j_i,1)$ and $\underline{\tau_2} <_{\underline{\tau}} \underline{\tau_1} <_{\underline{\tau}} \underline{\tau_i}$ is straightforward from **(Der3)**. It is easy to check that:

$$\mathsf{full\text{-}tr}^{\mathsf{n}:\underline{\tau_1},\underline{\tau}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau_i}} \;\equiv\; \mathsf{part\text{-}tr}^{\mathsf{n}:\underline{\tau_1}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau_i}} \wedge g(\phi^{\mathsf{in}}_{\underline{\tau}}) = \mathsf{Mac}^4_{\mathsf{k}^{\nu_{\tau_1}(\mathrm{ID})}_{\mathsf{m}}}(\mathsf{n}^j)$$

Moreover, $(\mathsf{Mac}^4_{\mathsf{k}^{\mathrm{ID}}_{\mathsf{m}}}(\mathsf{n}^j), \mathsf{Mac}^4_{\mathsf{k}^{\nu_{\tau_1}(\mathrm{ID})}_{\mathsf{m}}}(\mathsf{n}^j)) \in \mathsf{reveal}_{\tau_0}$. By **(Der3)**, there exists a derivation using FA and Dup of:

$$\frac{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi^{\mathsf{in}}_{\tau}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{part\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau_2,\tau} \sim \;\; \phi^{\mathsf{in}}_{\underline{\tau}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{part\text{-}tr}^{\mathsf{n}:\underline{\tau_1}}_{\mathsf{u}:\underline{\tau_2},\underline{\tau}}}$$

It is therefore easy to built the wanted derivation using only FA and Dup. ■

*Proof of **(Der2)**.* We recall that:

$$\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\tau} \;\equiv\; \begin{pmatrix} \mathsf{inj\text{-}auth}_{\tau}(\mathrm{ID}, j_0) \wedge \sigma^{\mathsf{in}}_{\tau}(\mathsf{e\text{-}auth}^{j_0}_{\mathrm{N}}) \neq \mathsf{UnknownId} \\ \wedge\, \pi_1(g(\phi^{\mathsf{in}}_{\tau})) = \mathrm{GUTI}^{j_0} \oplus \mathsf{f}^{\mathsf{r}}_{\mathsf{k}}(\mathsf{n}^{j_0}) \wedge\;\; \pi_2(g(\phi^{\mathsf{in}}_{\tau})) = \mathsf{Mac}^5_{\mathsf{k}_{\mathsf{m}}}(\langle \mathrm{GUTI}^{j_0}\,,\, \mathsf{n}^{j_0}\rangle) \end{pmatrix}$$

$$\mathsf{fu\text{-}tr}^{\mathsf{n}:\tau_1}_{\mathsf{u}:\underline{\tau}} \;\equiv\; \begin{pmatrix} \mathsf{inj\text{-}auth}_{\underline{\tau}}(\nu_{\tau}(\mathrm{ID}), j_0) \wedge \sigma^{\mathsf{in}}_{\underline{\tau}}(\mathsf{e\text{-}auth}^{j_0}_{\mathrm{N}}) \neq \mathsf{UnknownId} \\ \wedge\, \pi_1(g(\phi^{\mathsf{in}}_{\underline{\tau}})) = \mathrm{GUTI}^{j_0} \oplus \mathsf{f}^{\mathsf{r}}_{\mathsf{k}}(\mathsf{n}^{j_0}) \wedge\;\; \pi_2(g(\phi^{\mathsf{in}}_{\underline{\tau}})) = \mathsf{Mac}^5_{\mathsf{k}_{\mathsf{m}}}(\langle \mathrm{GUTI}^{j_0}\,,\, \mathsf{n}^{j_0}\rangle) \end{pmatrix}$$

Let $j_0 \in \mathbb{N}$, and τ_0 be such that $\tau = \tau_0, \mathsf{ai}$. It is straightforward to check that for any $n \in \mathbb{N}$:

$$\underbrace{\sigma_{\tau_0}(\mathsf{e\text{-}auth}_{\mathrm{N}}^{j_0}) = \mathsf{UnknownId}}_{\mathsf{unk}} \leftrightarrow \bigwedge_{1 \le i \le B} \neg \mathsf{net\text{-}e\text{-}auth}_{\tau}(\mathsf{A}_i, j_0)$$
$$\underbrace{\sigma_{\underline{\tau_0}}(\mathsf{e\text{-}auth}_{\mathrm{N}}^{j_0}) = \mathsf{UnknownId}}_{\underline{\mathsf{unk}}} \leftrightarrow \bigwedge_{1 \le i \le B} \neg \underline{\mathsf{net\text{-}e\text{-}auth}_{\tau}}(\mathsf{A}_i, j_0)$$

Since for all $1 \le i \le B$:

$$(\mathsf{net\text{-}e\text{-}auth}_{\tau}(\mathsf{A}_i, j_0) \sim \underline{\mathsf{net\text{-}e\text{-}auth}_{\tau}}(\mathsf{A}_i, j_0)) \in \mathsf{reveal}_{\tau_0}$$

and since $\mathsf{fu\text{-}tr}_{\mathsf{u}:\tau}^{\mathsf{n}:\tau_1} \wedge \mathsf{unk} \rightarrow \mathsf{false}$ and $\mathsf{fu\text{-}tr}_{\mathsf{u}:\underline{\tau}}^{\mathsf{n}:\tau_1} \wedge \underline{\mathsf{unk}} \rightarrow \mathsf{false}$, we deduce that:

$$\dfrac{\dfrac{\dfrac{\phi_{\tau}^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0}, b_{j_i} \wedge \neg\mathsf{unk} \sim \phi_{\underline{\tau}}^{\mathsf{in}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{fu\text{-}tr}_{\mathsf{u}:\underline{\tau}}^{\mathsf{n}:\tau_1} \wedge \neg\underline{\mathsf{unk}}}{\phi_{\tau}^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{unk}, \mathsf{false}, \mathsf{fu\text{-}tr}_{\mathsf{u}:\tau}^{\mathsf{n}:\tau_1} \wedge \neg\mathsf{unk} \sim \phi_{\underline{\tau}}^{\mathsf{in}}, \mathsf{r\text{-}reveal}_{\tau_0}, \underline{\mathsf{unk}}, \mathsf{false}, \mathsf{fu\text{-}tr}_{\mathsf{u}:\underline{\tau}}^{\mathsf{n}:\tau_1} \wedge \neg\underline{\mathsf{unk}}}\,\mathsf{Dup}^*}{\phi_{\tau}^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{unk}, \mathsf{fu\text{-}tr}_{\mathsf{u}:\tau}^{\mathsf{n}:\tau_1} \wedge \mathsf{unk}, \mathsf{fu\text{-}tr}_{\mathsf{u}:\tau}^{\mathsf{n}:\tau_1} \wedge \neg\mathsf{unk}}\,R \quad \sim\phi_{\underline{\tau}}^{\mathsf{in}}, \mathsf{r\text{-}reveal}_{\tau_0}, \underline{\mathsf{unk}}, \mathsf{fu\text{-}tr}_{\mathsf{u}:\underline{\tau}}^{\mathsf{n}:\tau_1} \wedge \underline{\mathsf{unk}}, \mathsf{fu\text{-}tr}_{\mathsf{u}:\underline{\tau}}^{\mathsf{n}:\tau_1} \wedge \neg\underline{\mathsf{unk}}}{\phi_{\tau}^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{fu\text{-}tr}_{\mathsf{u}:\tau}^{\mathsf{n}:\tau_1} \sim \phi_{\underline{\tau}}^{\mathsf{in}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{fu\text{-}tr}_{\mathsf{u}:\underline{\tau}}^{\mathsf{n}:\tau_1}}\,R + \mathsf{FA}^*$$

From the definitions, we get that:

$$\sigma_{\tau}^{\mathsf{in}}(\mathsf{b\text{-}auth}_{\mathrm{N}}^{j_0}) = \mathrm{ID} \rightarrow \left(\sigma_{\tau}^{\mathsf{in}}(\mathsf{e\text{-}auth}_{\mathrm{N}}^{j_0}) = \mathrm{ID} \vee \sigma_{\tau}^{\mathsf{in}}(\mathsf{e\text{-}auth}_{\mathrm{N}}^{j_0}) = \mathsf{UnknownId}\right)$$

Therefore:

$$\mathsf{fu\text{-}tr}_{\mathsf{u}:\tau}^{\mathsf{n}:\tau_1} \wedge \neg\mathsf{unk} \rightarrow \sigma_{\tau}^{\mathsf{in}}(\mathsf{e\text{-}auth}_{\mathrm{N}}^{j_0}) = \mathrm{ID} \rightarrow \mathsf{net\text{-}e\text{-}auth}_{\tau}(\mathrm{ID}, j_0)$$

Moreover:

$$\mathsf{net\text{-}e\text{-}auth}_{\tau}(\mathrm{ID}, j_0) \rightarrow \begin{pmatrix} \mathrm{GUTI}^{j_0} \oplus \mathsf{f}_{\mathsf{k}}^{\mathsf{r}}(\mathsf{n}^{j_0}) = [\mathsf{net\text{-}e\text{-}auth}_{\tau}(\mathrm{ID}, j_0)]\mathsf{t\text{-}suci\text{-}}\oplus_{\tau}(\mathrm{ID}, j_0) \\ \wedge\, \mathsf{Mac}_{\mathsf{k_m}}^{5}(\langle \mathrm{GUTI}^{j_0}, \mathsf{n}^{j_0} \rangle) = [\mathsf{net\text{-}e\text{-}auth}_{\tau}(\mathrm{ID}, j_0)]\mathsf{t\text{-}mac}_{\tau}(\mathrm{ID}, j_0) \end{pmatrix}$$

Using Proposition 4.15 on τ:

$$\mathsf{inj\text{-}auth}_{\tau}(\mathrm{ID}, j_0) \leftrightarrow \mathsf{n}^{j_0} = \sigma_{\tau}^{\mathsf{in}}(\mathsf{e\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}})$$

Using the observations above, we can rewrite $\mathsf{fu\text{-}tr}_{\mathsf{u}:\tau}^{\mathsf{n}:\tau_1} \wedge \neg\mathsf{unk}$ as follows:

$$\mathsf{fu\text{-}tr}_{\mathsf{u}:\tau}^{\mathsf{n}:\tau_1} \wedge \neg\mathsf{unk} = \begin{pmatrix} \mathsf{n}^{j_0} = \sigma_{\tau}^{\mathsf{in}}(\mathsf{e\text{-}auth}_{\mathrm{U}}^{\mathrm{ID}}) \wedge \neg\mathsf{unk} \\ \wedge\, \pi_1(g(\phi_{\tau}^{\mathsf{in}})) = [\mathsf{net\text{-}e\text{-}auth}_{\tau}(\mathrm{ID}, j_0)]\mathsf{t\text{-}suci\text{-}}\oplus_{\tau}(\mathrm{ID}, j_0) \\ \wedge\, \pi_2(g(\phi_{\tau}^{\mathsf{in}})) = [\mathsf{net\text{-}e\text{-}auth}_{\tau}(\mathrm{ID}, j_0)]\mathsf{t\text{-}mac}_{\tau}(\mathrm{ID}, j_0) \end{pmatrix}$$

Similarly, we can rewrite $\mathsf{fu\text{-}tr}_{\mathsf{u}:\underline{\tau}}^{\mathsf{n}:\tau_1} \wedge \neg\underline{\mathsf{unk}}$ as follows:

$$\mathsf{fu\text{-}tr}_{\mathsf{u}:\underline{\tau}}^{\mathsf{n}:\tau_1} \wedge \neg\underline{\mathsf{unk}} = \begin{pmatrix} \mathsf{n}^{j_0} = \sigma_{\underline{\tau}}^{\mathsf{in}}(\mathsf{e\text{-}auth}_{\mathrm{U}}^{\nu_{\tau}(\mathrm{ID})}) \wedge \neg\underline{\mathsf{unk}} \\ \wedge\, \pi_1(g(\phi_{\underline{\tau}}^{\mathsf{in}})) = [\underline{\mathsf{net\text{-}e\text{-}auth}_{\tau}}(\mathrm{ID}, j_0)]\underline{\mathsf{t\text{-}suci\text{-}}\oplus_{\tau}}(\mathrm{ID}, j_0) \\ \wedge\, \pi_2(g(\phi_{\underline{\tau}}^{\mathsf{in}})) = [\underline{\mathsf{net\text{-}e\text{-}auth}_{\tau}}(\mathrm{ID}, j_0)]\underline{\mathsf{t\text{-}mac}_{\tau}}(\mathrm{ID}, j_0) \end{pmatrix}$$

We can now conclude the proof:

$$\dfrac{\phi_{\tau}^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0} \sim \phi_{\underline{\tau}}^{\mathsf{in}}, \mathsf{r\text{-}reveal}_{\tau_0}}{\phi_{\tau}^{\mathsf{in}}, \mathsf{l\text{-}reveal}_{\tau_0}, \mathsf{fu\text{-}tr}_{\mathsf{u}:\tau}^{\mathsf{n}:\tau_1} \wedge \neg\mathsf{unk} \sim \phi_{\underline{\tau}}^{\mathsf{in}}, \mathsf{r\text{-}reveal}_{\tau_0}, \mathsf{fu\text{-}tr}_{\mathsf{u}:\underline{\tau}}^{\mathsf{n}:\tau_1} \wedge \neg\underline{\mathsf{unk}}}\, R + \mathsf{FA}^* + \mathsf{Dup}^*$$

■

4.14 Conclusion

We studied the privacy provided by the 5G-AKA authentication protocol. While this protocol is not vulnerable to IMSI catchers, we showed that several privacy attacks from the literature apply to it. We also discovered a novel desynchronization attack against PRIV-AKA, a modified version of AKA, even though it had been claimed secure.

We then proposed the AKA$^+$ protocol. This is a fixed version of 5G-AKA, which is both efficient and has improved privacy guarantees. To study AKA$^+$'s privacy, we defined the σ-unlinkability property. This is a new parametric privacy property, which requires the prover to establish privacy only for a subset of the standard unlinkability game scenarios. Finally, we formally proved that AKA$^+$ provides mutual authentication and σ_{ul}-unlinkability for any number of agents and sessions. Our proof is carried out in the Bana-Comon model, which is well-suited to the formal analysis of stateful protocols.

Chapter 5

Deciding Indistinguishability

The security proofs in the case studies of Chapter 3 and Chapter 4 are hand-made. Of course, this poses the question of their validity. In particular, the security analysis of the AKA$^+$ protocol in Chapter 4 is very long and tedious, and would strongly benefit from some tool support. In this chapter, we try to remedy to this problem by studying the problem of proof automation in the Bana-Comon equivalence model. Our main result is the decidability of a subset of the axioms presented in Chapter 2, which are a computationally sound, though incomplete, axiomatization of computational indistinguishability for protocols, with a bounded number of sessions, whose security is based on an IND-CCA$_2$ encryption scheme. Alternatively, our result can be viewed as the decidability of a family of cryptographic game transformations. Our proof relies on term rewriting and automated deduction techniques.

5.1 Introduction

When trying to prove a protocol, there are three possible outcomes: either we find a proof, which gives security guarantees corresponding to the attacker model; or we find an attack, meaning that the protocol is insecure; or the tool or the user (for interactive provers) could not carry out the proof and failed to find an attack. The latter case may happen for two different reasons. First, we could neither find a proof nor an attack because the proof method used is incomplete. In that case, we need either to make new assumptions and try again, or to use another proof technique. Second, the tool may not terminate on the protocol considered. This is problematic, as we do not know if we should continue waiting, and consume more resources and memory, or try another method.

This can be avoided for decidable classes of protocols and properties. Of course, such classes depend on both the attacker model and the security properties considered. We give here a non-exhaustive survey of such results. In the symbolic model, [CCZ10] shows decidability of secrecy (a reachability property) for a bounded number of sessions. In [DOT17], the authors show the decidability of a secrecy property for *depth-bounded* protocols, with an unbounded number of sessions, using Well-Structured Transition Systems [FS01]. Chrétien et al [CCD15] show the decidability of indistinguishability properties for a restricted class of protocols. E.g., they consider processes communicating on distinct channels and without else branches. The authors of [CCD17] show the decidability of symbolic equivalence for a bounded number of sessions, but with conditional branching.

In the computational model, we are aware of only one direct result. In [CCS13], the authors show the decidability of the security of a formula in the BC model, for *reachability properties*, for a bounded number of sessions. But there is an indirect way of getting decidability in the computational model, through a *computational soundness* theorem (e.g. [AR02]). A computational soundness theorem states that, for some given classes of protocols and properties, symbolic security implies computational security. These results usually make strong implementation assumptions (e.g. parsing assumptions, or the absence of dishonest keys), and require that the security primitives satisfy strong cryptographic hypothesis. By combining a decidability result in the symbolic model with a computational soundness theorem, which applies to the considered classes of protocols and properties (e.g. [BMU12] for reachability properties, or [BMR14] for indistinguishability properties), we obtain a decidability result in the computational model.

Contributions We tackle this problem in the Bana-Comon model. More precisely, we identify a subset Ax of the axioms presented in Chapter 2 which is both decidable and expressive enough to carry out proofs of security protocols. For this, we design a alternative set of axioms for the IND-CCA$_2$ cryptographic assumption [BDPR98], which are more amenable to automated deduction than the axioms presented in Chapter 2 (for IND-CCA$_1$). Our main result is the decidability of the problem:

Input: A ground formula $\vec{u} \sim \vec{v}$.
Question: Is $\mathsf{Ax} \wedge \vec{u} \not\sim \vec{v}$ unsatisfiable?

The main difficulty lies in dealing with equalities (defined through a term rewriting system R). First we show the completeness of an ordered strategy by commuting rule applications. This allows us to have only one rewriting modulo R at the beginning of the proof. We then bound the size of the terms after this rewriting as follows: we identify a class of proof cuts introducing arbitrary subterms; we give proof cut eliminations to remove them; and finally, we show that cut-free proofs are of bounded size w.r.t. the size of the conclusion.

Game Transformations Our result can be reinterpreted as the decidability of the problem of determining whether there exists a sequence of game transformations [Sho04, BR06] that allows to prove the security of a protocol. Indeed, one can associate to every axiom in Ax either a cryptographic assumption or a game transformation.

Each unitary axiom in Ax (i.e. each atomic formula) corresponds to an instantiation of the IND-CCA$_2$ game. For instance, in the simpler case of IND-CPA security of an encryption $\{\_\}^{\mathsf{n}}_{\mathsf{pk}}$, no polynomial-time adversary can distinguish between two cipher-texts, even if it chooses the two corresponding plain-texts. Initially, the public key pk is given to the adversary, who computes a pair of plain-texts $g(\mathsf{pk})$: g is interpreted as the adversary's computation. Then the two cipher-texts, corresponding to the encryptions of the first and second components of $g(\mathsf{pk})$, should be indistinguishable. This yields the unitary axiom:

$$\{\pi_1(g(\mathsf{pk}))\}^{\mathsf{n}}_{\mathsf{pk}} \sim \{\pi_2(g(\mathsf{pk}))\}^{\mathsf{n}}_{\mathsf{pk}}$$

Similarly, non-unitary axioms correspond to cryptographic game transformations. E.g., the FA axiom:

$$\frac{\vec{u} \sim \vec{v}}{f(\vec{u}) \sim f(\vec{v})}\ \mathsf{FA}$$

states that if no adversary can distinguish between the arguments of a function call, then no adversary can distinguish between the images. As for a cryptographic game transformation, the soundness of this axiom is shown by reduction. Given a winning adversary $\mathcal{A}$ against the conclusion $f(\vec{u}) \sim f(\vec{v})$, we build a winning adversary $\mathcal{B}$ against $\vec{u} \sim \vec{v}$: the adversary $\mathcal{B}$, on input $\vec{w}$ (which was sampled from $\vec{u}$ or $\vec{v}$), computes $f(\vec{w})$ and then gives the result to the distinguisher $\mathcal{A}$. The advantage of $\mathcal{B}$ against $\vec{u} \sim \vec{v}$ is then the advantage of $\mathcal{A}$ against $f(\vec{u}) \sim f(\vec{v})$, which is (by hypothesis) non negligible.

By interpreting every axiom in Ax as a cryptographic assumption or a game transformation, and the goal formula $\vec{u} \sim \vec{v}$ as the initial game, our result can be reformulated as showing the decidability of the following problem:

Input: An initial game $\vec{u} \sim \vec{v}$.
Question: Is there a sequence of game transformations in Ax showing that $\vec{u} \sim \vec{v}$ is secure?

From this point of view, our result guarantees a kind of sub-formula property for the intermediate games appearing in the game transformation proof. We may only consider intermediate games that are in a finite set computable from the original protocol: the other games are provably unnecessary detours. To our knowledge, our result is the first showing the decidability of a class of game transformations.

Scope and Limitations To achieve decidability, we had to remove or restrict some axioms. The most important restriction is arguably that we do not include the transitivity axiom. The transitivity axiom states that to show that $\vec{u} \sim \vec{v}$, it is sufficient to find a $\vec{w}$ such that $\vec{u} \sim \vec{w}$ and $\vec{w} \sim \vec{v}$. Obviously, this axiom is problematic for decidability, as the vector of term $\vec{w}$ must be guessed, and may be arbitrarily large. Therefore, instead of directly including transitivity, we push it inside the CCA$_2$ axiom schema, by allowing instances of the CCA$_2$ axiom to deal simultaneously with multiple keys and interleaved encryptions. Of course, this is at the cost of a more complex axiom. We do not know if our problem remains decidable when we include the transitivity axiom.

Applications The Bana-Comon indistinguishability model has been used to analyse RFID protocols, in Chapter 3, and a variant of the AKA protocol in Chapter 4. Moreover, it has also been used by Scerri and Stanley-Oakes to analyse a key-wrapping API [SS16], and by Bana, Chadha and Eeralla to prove an e-voting protocol [BCE18]. Ideally, we would like future case studies to be carried out automatically and machine checked. Because our procedure has a high complexity, it is unclear whether it can be used directly for this. Still, our procedure could be a building block in a tool doing an incomplete but faster heuristic exploration of the proof space.

CryptoVerif and EasyCrypt are based on game transformations, directly in the former and through the pRHL logic in the latter. Therefore, our result could be used to bring automation to these tools. Of course, both tools allow for more rules. Still, we could identify which game transformations or rules correspond to our axioms, and apply our result to obtain decidability for this subset of game transformations.

Related Works In [BCG+13], the authors design a set of inference rules to prove CPA and CCA security of asymmetric encryption schemes in the Random Oracle Model. The paper also presents an attack finding algorithm. The authors of [BCG+13] do not provide a decision algorithm for the designed inference rules. However, they designed proof search heuristics and implemented an automated tool, called ZooCrypt, to synthesize new CCA encryption schemes. For small schemes, this procedure can show CCA security or find an attack in more than 80% of the cases. In 20% of the cases, security remains undecided. Additionally, ZooCrypt automatically generates concrete security bounds.

In [JR12], the authors study proof automation in the UC framework [Can01]. They design a complete procedure for deciding the existence of a simulator, for ideal and real functionalities using if-then-else, equality, random samplings and xor. Therefore their algorithm cannot be used to analyse functionalities relying on more complex functions (e.g., public key encryption), or stateful functionalities. This restricts the protocols that can be checked. Still, their method is *semantically* complete (while we are complete w.r.t. a fixed set of inference rules): if there exists a simulator, they will find it.

In [BDK+10], the authors show the decidability of the problem of the equality of two distributions, for a *specific* equational theory (concatenation, projection and xor). Then, for *arbitrary* equational theories, they design a proof system for proving the equality of two distributions. This second contribution has similarities with our work, but differ in two ways.

First, the proof system of [BDK+10] shares some rules with ours, e.g. the R, Dup and FA rules. But it does not allow for reasoning on terms using if_then_else_. E.g., they do not have a counterpart to the CS rule. This is a major difference, as most of the difficulties encountered in the design of our decision procedure result from the if_then_else_ conditionals. Moreover, there are no rules corresponding to cryptographic assumptions, as our CCA_2 rules. Because of this and the lack of support for reasoning on branching terms, the analysis of security protocols is out of the scope of [BDK+10].

Second, the authors do not provide a decision procedure for their inference rules, but instead rely on heuristics.

Outline We introduce the axioms in Section 5.2, which include, in particular, the equality axioms R and the cryptographic axioms CCA_2. In Section 5.3, we prove that the set of equality axioms R can be defined using a convergent term rewriting system $\to_R$. In Section 5.4, we define the cryptographic axioms CCA_2, and prove some properties of these axioms. We state the main result in Section 5.5, and depict the difficulties of the proof. We prove several rule commutations in Section 5.6, which allow us to obtain a complete ordered strategy for our fragment. In Section 5.7, we prove, through a cut elimination procedure, that we can use an eager reduction strategy for some rules of R. We then define a normal form for derivations, and prove that we can assume w.l.o.g. that derivations are in normal form in Section 5.8. We prove key properties of terms appearing in derivation in normal form in Section 5.9, and in Section 5.10 we characterize subterms that corresponds to detours in proof. We use this characterization in Section 5.11 to show a first main proof cut elimination lemma. We prove a second main proof cut elimination lemma in Section 5.12, and show that the resulting derivations contain only subterms of bounded size. Finally, we conclude in Section 5.13.

$$\frac{u_{\pi(1)},\dots,u_{\pi(n)} \sim v_{\pi(1)},\dots,v_{\pi(n)}}{u_1,\dots,u_n \sim v_1,\dots,v_n}\ \mathsf{Perm} \qquad \frac{\vec{u},t \sim \vec{v},t'}{\vec{u} \sim \vec{v}}\ \mathsf{Restr} \qquad \text{for any } s =_R t,\ \frac{\vec{u},t \sim \vec{v}}{\vec{u},s \sim \vec{v}}\ R$$

$$\frac{\vec{u}_1,\vec{v}_1 \sim \vec{u}_2,\vec{v}_2}{f(\vec{u}_1),\vec{v}_1 \sim f(\vec{u}_2),\vec{v}_2}\ \mathsf{FA}_{\setminus \mathbf{0}} \quad \text{where } f \in \mathcal{F}_{\setminus \mathbf{0}} \qquad \frac{\vec{u},t \sim \vec{v},t'}{\vec{u},t,t \sim \vec{v},t',t'}\ \mathsf{Dup} \qquad \frac{\vec{v} \sim \vec{u}}{\vec{u} \sim \vec{v}}\ \mathsf{Sym}$$

$$\text{for any } b,b' \in \mathcal{T}(\mathcal{F}_{\setminus \mathsf{if}},\mathcal{N}),\ \frac{\vec{w},b,(u_i)_i \sim \vec{w}',b',(u'_i)_i \qquad \vec{w},b,(v_i)_i \sim \vec{w}',b',(v'_i)_i}{\vec{w},(\mathsf{if}\ b\ \mathsf{then}\ u_i\ \mathsf{else}\ v_i)_i \sim \vec{w}',(\mathsf{if}\ b'\ \mathsf{then}\ u'_i\ \mathsf{else}\ v'_i)_i}\ \mathsf{CS}^{\mathsf{no}}_{\mathsf{if}}$$

Conventions: π is a permutation of $\{1,\dots,n\}$.

Figure 5.1: The Axioms Struct-Ax.

5.2 Axioms

For the strategy, we use only a subset of the axioms presented in Figure 2.2 of Chapter 2, with some restrictions. Arguably, the most important restriction is the interdiction of the transitivity rule Trans. Indeed, this rule requires to guess a intermediate term, which is, a priori, arbitrarily large. Before discussing the restrictions on the axioms, we define some subsets of the set of function symbols $\mathcal{F}$:

Definition 5.1. We let $\mathcal{F}_{\setminus \mathbf{0}}$, $\mathcal{F}_{\setminus \mathsf{if}}$ and $\mathcal{F}_{\setminus \mathsf{if},\mathbf{0}}$ be the subsets of $\mathcal{F}$ defined by:

$$\mathcal{F}_{\setminus \mathbf{0}} = \mathcal{F} \setminus \{\mathbf{0}(\_)\} \qquad \mathcal{F}_{\setminus \mathsf{if}} = \mathcal{F} \setminus \{\mathsf{if\_then\_else\_}\} \qquad \mathcal{F}_{\setminus \mathsf{if},\mathbf{0}} = \mathcal{F} \setminus \{\mathbf{0}(\_), \mathsf{if\_then\_else\_}\}$$

Restrictions We give in Figure 5.1 the structural axioms used in this chapter. We comment on some of the restrictions:

- We restrict the case study rule, by only considering instances of the rule where the conditionals b and b' are if-free. This restriction is used in the decidability proof, but might be unnecessary. We let $\mathsf{CS}^{\mathsf{no}}_{\mathsf{if}}$ be the rule:

$$\frac{\vec{w},b,(u_i)_i \sim \vec{w}',b',(u'_i)_i \qquad \vec{w},b,(v_i)_i \sim \vec{w}',b',(v'_i)_i}{\vec{w},(\mathsf{if}\ b\ \mathsf{then}\ u_i\ \mathsf{else}\ v_i)_i \sim \vec{w}',(\mathsf{if}\ b'\ \mathsf{then}\ u'_i\ \mathsf{else}\ v'_i)_i}\ \mathsf{CS}^{\mathsf{no}}_{\mathsf{if}} \quad \text{when } b,b' \in \mathcal{T}(\mathcal{F}_{\setminus \mathsf{if}},\mathcal{N}),$$

- We replace the equality rule Equ by a weaker rule R. Basically, instead of having the formula $u = v$ as premise (like in Equ), we require that u can be rewritten into v using a set equalities R (given in Figure 5.2). We give details about this change later in this section.
- We use a modified version of the CCA_2 axioms, which already includes transitivity (as we did not include it in the set of axioms). We give some high-level details later in this section, and present the full axioms in Section 5.4.
- We reserve the function symbol $\mathbf{0}_{/1}$ for the CCA_2 axioms. In particular, we forbid to apply the Function Application rule $\mathsf{FA}_{\setminus \mathbf{0}}$ to $\mathbf{0}(\_)$. This is necessary for technical reasons, but may be unnecessary for decidability. We let $\mathsf{FA}_{\setminus \mathbf{0}}$ be the rule:

$$\frac{\vec{u}_1,\vec{v}_1 \sim \vec{u}_2,\vec{v}_2}{f(\vec{u}_1),\vec{v}_1 \sim f(\vec{u}_2),\vec{v}_2}\ \mathsf{FA}_{\setminus \mathbf{0}} \quad \text{where } f \in \mathcal{F}_{\setminus \mathbf{0}}$$

Equality Axioms To handle equalities automatically, we are going to replace the equality axioms given in Figure 2.1 by rewrite rules: we introduce a set of equalities R (given in Figure 5.2) and its congruence closure $=_R$. We split R in four sub-parts: R_1 contains the functional correctness assumptions on the pair and encryption; R_2 and R_3 contain, respectively, the homomorphism properties and simplification rules of the if_then_else_; and R_4 allows to change the order in which conditional tests are performed.

We then introduce a recursive set of rules to replace the equality rule Equ:

$$\frac{\vec{u},t \sim \vec{v}}{\vec{u},s \sim \vec{v}}\ R \qquad (s,t \text{ ground terms with } s =_R t)$$

$$R_1 \begin{cases} \pi_i(\langle x_1, x_2\rangle) = x_i \\ \mathsf{dec}(\{x\}^z_{\mathsf{pk}(y)}, \mathsf{sk}(y)) = x \\ \mathsf{eq}(x,x) = \mathsf{true} \end{cases}$$

$$R_2 \begin{cases} f(\vec{u}, \mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ y, \vec{v}) = \mathsf{if}\ b\ \mathsf{then}\ f(\vec{u},x,\vec{v})\ \mathsf{else}\ f(\vec{u},y,\vec{v}) & (f \in \mathcal{F}_{\setminus \mathsf{if}}) \\ \mathsf{if}\ (\mathsf{if}\ b\ \mathsf{then}\ a\ \mathsf{else}\ c)\ \mathsf{then}\ x\ \mathsf{else}\ y = \mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ x\ \mathsf{else}\ y)\ \mathsf{else}\ (\mathsf{if}\ c\ \mathsf{then}\ x\ \mathsf{else}\ y) \end{cases}$$

$$R_3 \begin{cases} \mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ x = x \\ \mathsf{if}\ \mathsf{true}\ \mathsf{then}\ x\ \mathsf{else}\ y = x \\ \mathsf{if}\ \mathsf{false}\ \mathsf{then}\ x\ \mathsf{else}\ y = y \\ \mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ y)\ \mathsf{else}\ z = \mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ z \\ \mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ (\mathsf{if}\ b\ \mathsf{then}\ y\ \mathsf{else}\ z) = \mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ z \end{cases}$$

$$R_4 \begin{cases} \mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ x\ \mathsf{else}\ y)\ \mathsf{else}\ z = \mathsf{if}\ a\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ z)\ \mathsf{else}\ (\mathsf{if}\ b\ \mathsf{then}\ y\ \mathsf{else}\ z) \\ \mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ (\mathsf{if}\ a\ \mathsf{then}\ y\ \mathsf{else}\ z) = \mathsf{if}\ a\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ y)\ \mathsf{else}\ (\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ z) \end{cases}$$

Figure 5.2: $R = R_1 \cup R_2 \cup R_3 \cup R_4$

It turns out that there exists a convergent orientation $\to_R$ of $=_R$.[1] We describe how we orient equalities of R, and prove that the resulting term rewriting system is convergent, later, in Section 5.3. Still, we anticipate and give the outlines of the orientation now.

We let $R_{\leq 3}$ be $R_1 \cup R_2 \cup R_3$. By orienting $R_{\leq 3}$ from left to right, and carefully choosing an orientation for the ground instances of R_4, we can build a recursive convergent term rewriting system $\to_R$:

- First, we choose the orientation of the rules in R_4. This is done by using a Lexicographic Path Ordering [DJ90] on the conditionals, modified using a user-chosen total order $\succ_u$ on if-free $R_{\leq 3}$-irreducible conditionals. We show that the resulting term rewriting system is locally confluent.
- Then, we show local confluence and termination of our term rewriting system. We deduce that it is convergent using Newman's lemma.

Theorem 5.1. *There exists an orientation $\to_{R_4}$ of R_4 such that the resulting term rewriting system $\to_R = \to_{R_{\leq 3}} \cup \to_{R_4}$ is convergent on ground terms.*

The CCA$_2$ Axioms Before giving the CCA$_2$ axioms, we recall the CCA$_1^s$ axioms from Section 2.6.1:

$$\frac{\mathsf{len}(u) = \mathsf{len}(v)}{\vec{w}, \{u\}^{\mathsf{n_e}}_{\mathsf{pk(n)}} \sim \vec{w}, \{v\}^{\mathsf{n_e}}_{\mathsf{pk(n)}}}\ \text{CCA}_1^s \qquad \text{when } \begin{cases} \mathsf{fresh}(\mathsf{n_e}; \vec{w}, u, v) \\ \mathsf{n} \sqsubseteq_{\mathsf{pk}(\cdot),\mathsf{sk}(\cdot)} \vec{w}, u, v \ \wedge\ \mathsf{sk}(\mathsf{n}) \sqsubseteq_{\mathsf{dec}(\_,\cdot)} \vec{w}, u, v \end{cases} \tag{5.1}$$

Remark 5.1. We do not use the stronger CCA$_1$ axioms. The CCA$_1$ axiom schema allows to have a different vectors of terms $\vec{w}$ and $\vec{w}'$ on the left and the right, but must be provided with a proof of $\vec{w} \sim \vec{w}'$. The CCA$_1^s$ axioms are simpler and easier to handle. □

To extend this axiom to the IND-CCA$_2$ game, we need to deal with calls to the decryption oracle performed after some calls to the left-right oracle. For example, consider the case where one call (u, v) was made. Let $\alpha \equiv \{u\}^{\mathsf{n_e}}_{\mathsf{pk(n)}}$ and $\alpha' \equiv \{v\}^{\mathsf{n_e}}_{\mathsf{pk(n)}}$ be the result of the call on, respectively, the left and the right. A naive first try could be to state that decryptions are indistinguishable. That is, if we let $s \equiv t[\alpha]$ and $s' \equiv t[\alpha']$, then $\mathsf{dec}(s, \mathsf{sk}(\mathsf{n})) \sim \mathsf{dec}(s', \mathsf{sk}(\mathsf{n}))$. But this is not valid: for example, take $u \equiv 0, v \equiv 1$, $t \equiv g([])$ (where $[]$ is a hole variable). Then the adversary can, by interpreting g as the identity function, obtain a term semantically equal to 0 on the left and 1 on the right. This allows him to distinguish between the left and right cases.

[1] Actually, there are many such orientations, as we will see later.

We prevent this by adding a guard checking that we are not decrypting α on the left (resp. α' on the right): if not, we return the decryption $\mathsf{dec}(t[\alpha], \mathsf{sk}(\mathsf{n}))$ (resp. $\mathsf{dec}(t[\alpha'], \mathsf{sk}(\mathsf{n}))$) asked for, otherwise we return a dummy message $\mathbf{0}(\mathsf{dec}(t[\alpha], \mathsf{sk}(\mathsf{n})))$ (resp. $\mathbf{0}(\mathsf{dec}(t[\alpha'], \mathsf{sk}(\mathsf{n})))$). $\mathsf{CCA2}^{\mathrm{s}}$ is the (recursive) set of unitary axioms:

$$\frac{}{\begin{array}{l}\vec{w}, \alpha, \mathsf{if}\ \mathsf{eq}(t[\alpha], \alpha)\ \mathsf{then}\ \mathbf{0}(\mathsf{dec}(t[\alpha], \mathsf{sk}(\mathsf{n})))\\ \qquad\qquad \mathsf{else}\ \mathsf{dec}(t[\alpha], \mathsf{sk}(\mathsf{n}))\end{array} \sim \begin{array}{l}\vec{w}, \alpha', \mathsf{if}\ \mathsf{eq}(t[\alpha'], \alpha')\ \mathsf{then}\ \mathbf{0}(\mathsf{dec}(t[\alpha'], \mathsf{sk}(\mathsf{n})))\\ \qquad\qquad \mathsf{else}\ \mathsf{dec}(t[\alpha'], \mathsf{sk}(\mathsf{n}))\end{array}}\ \mathsf{CCA2}^{\mathrm{s}}$$

under the side-conditions of the CCA_1^s axioms in (5.1), plus a side-condition on length (that we omit here), to account for the fact that we removed the premise $\mathsf{len}(u) = \mathsf{len}(v)$. We do not prove validity of these axioms yet, as we are going to use a modified version CCA_2 of this axiom schema:

- We are going to allow for any number of calls to the left-right oracle, by adding a guard for each call. We use extra syntactic side-conditions to remove superfluous guards.
- In the axioms Struct-Ax given in Figure 5.1, we did not include the alpha renaming axiom α-equ. Instead, our CCA_2 axiom schema is closed under α-renaming.
- We restrict t to be without $\mathsf{if\_then\_else\_}$ and $\mathbf{0}(\_)$. This is needed in the completeness proof.
- Finally, the axioms allow for an arbitrary number of public/private key pairs to be used simultaneously, and an instance of the axiom can contain any number of interleaved left-right and decryption oracles calls.

Remark 5.2. The last point is what allows us to avoid transitivity in proofs. For example, consider four encryptions, two of them (α and γ) using the public key $\mathsf{pk}(\mathsf{n})$, and the other two (β and δ) using the public key $\mathsf{pk}(\mathsf{n}')$:

$$\alpha \equiv \{A\}^{\mathsf{n}_0}_{\mathsf{pk}(\mathsf{n})} \quad \beta \equiv \{B\}^{\mathsf{n}_1}_{\mathsf{pk}(\mathsf{n}')} \quad \gamma \equiv \{C\}^{\mathsf{n}_0}_{\mathsf{pk}(\mathsf{n})} \quad \delta \equiv \{D\}^{\mathsf{n}_1}_{\mathsf{pk}(\mathsf{n}')}$$

Then the following formula is a valid instance of the CCA_2 axioms on, simultaneously, $\mathsf{pk}(\mathsf{n})$ and $\mathsf{pk}(\mathsf{n}')$:

$$\frac{}{\alpha, \beta \sim \gamma, \delta}\ \mathrm{CCA}_2(\mathsf{pk}(\mathsf{n}), \mathsf{pk}(\mathsf{n}'))$$

However, proving the above formula using CCA_2 only on one key at a time, as in [BCL14], uses a hybrid argument, which requires transitivity:

$$\frac{\dfrac{}{\alpha, \beta \sim \alpha, \delta}\ \mathrm{CCA}_2(\mathsf{pk}(\mathsf{n}'))\qquad \dfrac{}{\alpha, \delta \sim \gamma, \delta}\ \mathrm{CCA}_2(\mathsf{pk}(\mathsf{n}))}{\alpha, \beta \sim \gamma, \delta}$$

□

5.2.1 Comments and Examples

Our set of axioms is not complete w.r.t. the computational interpretation semantics. Indeed, being so would mean axiomatizing exactly which distributions (computable in polynomial time) can be distinguished by PPTM*s*, which is unrealistic and would lead to undecidability. E.g., if we completely axiomatized IND-CCA_2, then showing the satisfiability of our set of axioms would show the existence of IND-CCA_2 functions, which is an open problem.

Still, our axioms are expressive enough to complete concrete proofs of security. We illustrate this on two examples: a proof of the simple formula from Example 2.3 and a proof of the security of one round of the NSL protocol [Low95]. Of course, such proofs can be found automatically using our decision procedure.

Example 5.1. We give a proof of the formula below:

$$\mathsf{if}\ g()\ \mathsf{then}\ \mathsf{n}_0\ \mathsf{else}\ \mathsf{n}_1 \sim \mathsf{n}$$

First, we introduce a conditional $g()$ on the right to match the structure of the left side using R. Then, we split the proof in two using the $\mathsf{CS}^{\mathsf{no}}_{\mathsf{if}}$ axiom. We conclude using the reflexivity modulo α-renaming axiom (this axiom is subsumed by CCA_2, therefore we do not include it in Ax).

$$\dfrac{\dfrac{\dfrac{}{g(), \mathsf{n}_0 \sim g(), \mathsf{n}}\ \mathsf{Refl}\qquad \dfrac{}{g(), \mathsf{n}_1 \sim g(), \mathsf{n}}\ \mathsf{Refl}}{\mathsf{if}\ g()\ \mathsf{then}\ \mathsf{n}_0\ \mathsf{else}\ \mathsf{n}_1 \sim \mathsf{if}\ g()\ \mathsf{then}\ \mathsf{n}\ \mathsf{else}\ \mathsf{n}}\ \mathsf{CS}^{\mathsf{no}}_{\mathsf{if}}}{\mathsf{if}\ g()\ \mathsf{then}\ \mathsf{n}_0\ \mathsf{else}\ \mathsf{n}_1 \sim \mathsf{n}}\ R$$

□

Example 5.2 (Proof of NSL)**.** We consider a simple setting with one initiator A, one respondent B and no key server. An execution of the NSL protocol is given in Figure 5.3.

We write this in the logic. First, we let $\mathsf{pk_A} \equiv \mathsf{pk}(\mathsf{n_A})$ and $\mathsf{sk_A} \equiv \mathsf{sk}(\mathsf{n_A})$ be the public/private key pair of agent A (we define similarly $(\mathsf{pk_B}, \mathsf{sk_B})$). Since A does not wait for any input before sending its first message, we put it into the initial frame:

$$\phi_0 \equiv \mathsf{pk_A}, \mathsf{pk_B}, \{\langle \mathsf{n_A}\,,\, \mathsf{A}\rangle\}^{\mathsf{n}_0}_{\mathsf{pk}_B}$$

Then, both agents wait for a message before sending a single reply. When receiving $\mathbf{x_A}$ (resp. $\mathbf{x_B}$), the answer of agent A (resp. B) is expressed in the logic as follows:

$$\begin{aligned} t_{\mathsf{A}}[\mathbf{x_A}] \;\equiv\; & \mathsf{if}\ \mathsf{eq}(\pi_1(\mathsf{dec}(\mathbf{x_A}, \mathsf{sk_A})), \mathsf{n_A}) \quad \mathsf{then} \\ & \quad \mathsf{if}\ \mathsf{eq}(\pi_2(\pi_2(\mathsf{dec}(\mathbf{x_A}, \mathsf{sk_A}))), \mathsf{B})\ \mathsf{then} \\ & \quad\quad \{\pi_1(\pi_2(\mathsf{dec}(\mathbf{x_A}, \mathsf{sk_A})))\}^{\mathsf{n}_2}_{\mathsf{pk_B}} \\ t_{\mathsf{B}}[\mathbf{x_B}] \;\equiv\; & \mathsf{if}\ \mathsf{eq}(\pi_2(\mathsf{dec}(\mathbf{x_B}, \mathsf{sk_B})), \mathsf{A}) \quad\quad \mathsf{then} \\ & \quad\quad \{\langle \pi_1(\mathsf{dec}(\mathbf{x_B}, \mathsf{sk_B}))\,,\, \langle \mathsf{n_B}\,,\, \mathsf{B}\rangle\rangle\}^{\mathsf{n}_1}_{\mathsf{pk_A}} \end{aligned}$$

During an execution of the protocol, the adversary has several choices. First, it decides whether to interact with A or B first. We focus on the case where it first sends a message to B (the other case is similar). Then, it can honestly forward the messages or forge new ones. E.g., when sending the first message to B, it can either forward A's message $\{\langle \mathsf{n_A}\,,\, \mathsf{A}\rangle\}^{\mathsf{n}_0}_{\mathsf{pk}_B}$ or forge a new message. We are going to prove the security of the protocol in the following case (the other cases are similar):

- the first message, sent to B, is honest. Therefore we take $\mathbf{x_B} \equiv \{\langle \mathsf{n_A}\,,\, \mathsf{A}\rangle\}^{\mathsf{n}_0}_{\mathsf{pk}_B}$, and B answers:

$$t_{\mathsf{B}}[\mathbf{x_B}] =_R \{\langle \mathsf{n_A}\,,\, \langle \mathsf{n_B}\,,\, \mathsf{B}\rangle\rangle\}^{\mathsf{n}_1}_{\mathsf{pk_A}}$$

- the second message, sent to A, is forged. Therefore we take $\mathbf{x_A} \equiv g(\phi_1)$, where $\phi_1 \equiv \phi_0, t_{\mathsf{B}}[\mathbf{x_B}]$. As, a priori, nothing prevents $g(\phi_1)$ from being equal to $t_{\mathsf{B}}[\mathbf{x_B}]$, we use the conditional $\mathsf{eq}(g(\phi_1), t_{\mathsf{B}}[\mathbf{x_B}])$ to ensure that this message is forged. The answer from A is then:

$$s \equiv \mathsf{if}\ \mathsf{eq}(g(\phi_1), t_{\mathsf{B}}[\mathbf{x_B}])\ \mathsf{then}\ 0\ \mathsf{else}\ t_{\mathsf{A}}[g(\phi_1)] \tag{5.2}$$

We show the secrecy of the nonce $\mathsf{n_B}$: we let $t'_{\mathsf{B}}[\mathbf{x_B}]$ (resp. s') be the term $t_{\mathsf{B}}[\mathbf{x_B}]$ (resp. s) where we replaced all occurrences of $\mathsf{n_B}$ by 0. For example, $t'_{\mathsf{B}}[\mathbf{x_B}] =_R \{\langle \mathsf{n_A}\,,\, \langle 0\,,\, \mathsf{B}\rangle\rangle\}^{\mathsf{n}_1}_{\mathsf{pk_A}}$. This yields the goal:

$$\phi_0, t_{\mathsf{B}}[\mathbf{x_B}], s \sim \phi_0, t'_{\mathsf{B}}[\mathbf{x_B}], s' \tag{5.3}$$

We let δ be the guarded decryption that will be used in the CCA_2 axiom:

$$\begin{aligned} \delta \;\equiv\; & \mathsf{if}\ \mathsf{eq}(g(\phi_1), t_{\mathsf{B}}[\mathbf{x_B}])\ \mathsf{then}\ \mathbf{0}(\mathsf{dec}(g(\phi_1), \mathsf{sk_A})) \\ & \quad\quad\quad\quad\quad\quad\quad\quad\quad\quad \mathsf{else}\ \mathsf{dec}(g(\phi_1), \mathsf{sk_A}) \end{aligned} \tag{5.4}$$

and s_δ be the term s where all occurrences of $\mathsf{dec}(g(\phi_1), \mathsf{sk_A})$ have been replaced by δ. We have $s =_R s_\delta$. We also introduce shorthands for some subterms of s_δ: we let a_δ, b_δ and e_δ be the terms $\mathsf{eq}(\pi_1(\delta), \mathsf{n_A})$, $\mathsf{eq}(\pi_2(\pi_2(\delta))), \mathsf{B})$ and $\{\pi_1(\pi_2(\delta))\}^{\mathsf{n}_2}_{\mathsf{pk_B}}$. We define $\delta', s'_{\delta'}, a'_{\delta'}, b'_{\delta'}$ and $e'_{\delta'}$ similarly.

We then rewrite s and s' into s_δ and $s'_{\delta'}$ using R. Then we apply $\mathsf{FA}_{\setminus \mathbf{0}}$ several times, first to deconstruct s_δ and $s'_{\delta'}$, and then to deconstruct a_δ, b_δ and $a'_{\delta'}, b'_{\delta'}$. Finally, we use Dup to remove duplicates, and we apply CCA_2 simultaneously on key pairs $(\mathsf{pk_A}, \mathsf{sk_A})$ and $(\mathsf{pk_B}, \mathsf{sk_B})$ (we omit here the details of the syntactic side-conditions that have to be checked):

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\phi_0, t_{\mathsf{B}}[\mathbf{x_B}], \mathsf{n_A}, \delta, e_\delta \sim \phi_0, t'_{\mathsf{B}}[\mathbf{x_B}], \mathsf{n_A}, \delta', e'_{\delta'}}\ \mathrm{CCA}_2}{\phi_0, t_{\mathsf{B}}[\mathbf{x_B}], a_\delta, b_\delta, e_\delta \sim \phi_0, t'_{\mathsf{B}}[\mathbf{x_B}], a'_{\delta'}, b'_{\delta'}, e'_{\delta'}}\ (\mathsf{FA}_{\setminus \mathbf{0}}, \mathsf{Dup})^*}{\phi_0, t_{\mathsf{B}}[\mathbf{x_B}], s_\delta \sim \phi_0, t'_{\mathsf{B}}[\mathbf{x_B}], s'_{\delta'}}\ (\mathsf{FA}_{\setminus \mathbf{0}}, \mathsf{Dup})^*}{\phi_0, t_{\mathsf{B}}[\mathbf{x_B}], s \sim \phi_0, t'_{\mathsf{B}}[\mathbf{x_B}], s'}\ R$$

□

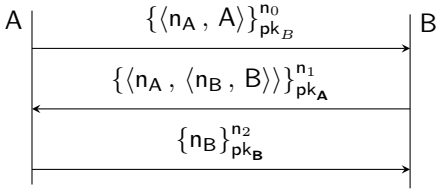


Figure 5.3: The NSL protocol.

Remark 5.3. The process of computing the formula in (5.3) from the protocol description can be done automatically, using a simple procedure similar to the folding procedure from [BCL14]. The formula in (5.3) has already been split between the honest and dishonest cases using the case study axiom $\mathsf{CS}^{\mathsf{no}}_{\mathsf{if}}$ (we omit the $\mathsf{CS}^{\mathsf{no}}_{\mathsf{if}}$ applications to keep the proof readable). For example, the term in (5.2) is the "else" branch of a $\mathsf{CS}^{\mathsf{no}}_{\mathsf{if}}$ application on conditional $\mathsf{eq}(g(\phi_1), t_{\mathsf{B}}[\mathbf{x}_{\mathsf{B}}])$ (which does not contain nested conditionals, as required by the $\mathsf{CS}^{\mathsf{no}}_{\mathsf{if}}$ side-condition). □

5.3 The Term Rewriting System R

In this section, we orient the equalities in $=_R$, and show that the resulting Term Rewriting System is convergent. First, we recall the definition of a Lexicographic Path Ordering [DJ90].

Definition 5.2. Let $\succ_f$ be a total precedence over function symbols. The lexicographic path ordering associated with $\succ_f$ is the total order on ground terms defined by:

$$s = f(s_1, \ldots, s_n) \succ t = g(t_1, \ldots, t_m) \text{ iff } \begin{cases} \exists i \in [\![1, n]\!] \text{ s.t. } s_i \succeq t \\ \text{or } f = g \wedge \forall j \in [\![1, m]\!], s \succ t_j \wedge s_1, \ldots, s_n \succ_{lex} t_1, \ldots, t_n \\ \text{or } f \succ_f g \wedge \forall j \in [\![1, m]\!], s \succ t_j \end{cases}$$

Let $\succ_f$ be a total precedence on $\mathcal{F}, \mathcal{N}$ such that $\mathsf{if\_then\_else\_}$ is the smallest element (elements of $\mathcal{N}$ are treated as function symbols of arity zero). We define the lexicographic path ordering $\succ$ on ground terms using $\succ_f$.

Definition 5.3. Let $\succ$ be the lexicographic path ordering on $\mathcal{T}(\mathcal{F}, \mathcal{N})$ using precedence $\succ_f$.

Now, we want to have some leeway in the ordering of terms. We do this by letting $\succ_u$ be an arbitrary total order on if-free conditionals that are $R_{\leq 3}$-irreducible. We define the extension $\succ_u^{\mathsf{lpo}}$ of $\succ_u$ to arbitrary ground conditionals. Basically, $\succ_u^{\mathsf{lpo}}$ compares if-free $R_{\leq 3}$-irreducible conditionals using $\succ_u$; conditionals that are *not* if-free or *not* $R_{\leq 3}$-irreducible are compared using $\succ$; and we choose the behavior of $\succ_u^{\mathsf{lpo}}$ on cross-cases (i.e. one if-free $R_{\leq 3}$-irreducible conditional and one *not* if-free or *not* $R_{\leq 3}$-irreducible) so as to have a pre-order.

Definition 5.4. For any total ordering $\succ_u$ on ground if-free $R_{\leq 3}$-irreducible terms, we let $\succ_u^{\mathsf{lpo}}$ be the relation defined by:

$$b \succ_u^{\mathsf{lpo}} a = \begin{cases} b \succ_u a & \text{if } a \text{ and } b \text{ are if-free and } R_{\leq 3}\text{-irreducible} \\ b \succ a & \text{if } a \text{ and } b \text{ are not if-free or not } R_{\leq 3}\text{-irreducible} \\ \mathsf{true} & \text{if } a \text{ is if-free and } R_{\leq 3}\text{-irreducible, and } b \text{ is not} \\ \mathsf{false} & \text{if } b \text{ is if-free and } R_{\leq 3}\text{-irreducible, and } a \text{ is not} \end{cases}$$

We then order R_4 using $\succ_u^{\mathsf{lpo}}$.

Definition 5.5. For any total ordering $\succ_u$ on ground if-free $R_{\leq 3}$-irreducible terms, we let $\to_{R_4^{\succ_u}}$ be the ordering of R_4 defined by:

$\mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ x\ \mathsf{else}\ y)\ \mathsf{else}\ z \to \mathsf{if}\ a\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ z)\ \mathsf{else}\ (\mathsf{if}\ b\ \mathsf{then}\ y\ \mathsf{else}\ z)$ (when $b \succ_u^{\mathsf{lpo}} a$)

$\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ (\mathsf{if}\ a\ \mathsf{then}\ y\ \mathsf{else}\ z) \to \mathsf{if}\ a\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ y)\ \mathsf{else}\ (\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ z)$ (when $b \succ_u^{\mathsf{lpo}} a$)

Moreover, we let $\to_{R^{\succ_u}} = \to_{R_1} \cup \to_{R_2} \cup \to_{R_3} \cup \to_{R_4^{\succ_u}}$.

The term rewriting system $\to_{R^{\succ_u}}$ is an orientation of the rules given in Figure 5.2. When the ordering $\succ_u$ is irrelevant, we write $\to_R$ instead of $\to_{R^{\succ_u}}$. We state the convergence theorem.

Theorem 5.2. *For all $\succ_u$, the term rewriting system $\to_{R^{\succ_u}}$ is convergent on ground terms.*

Observe that this result subsumes Theorem 5.1.

Proof. Using Newman's lemma, we only need to prove that $\to_{R^{\succ_u}}$ is locally confluent and terminating.

Local Confluence We show that all critical pairs are joinable. Normally, we would rely on some automated checker for local confluence. Unfortunately, as we rely on a side-condition to orient R_4 (using a LPO), writing down the rules in a tool is not straightforward. By consequence we believe it is simpler to manually check that every critical pair is joinable. We give below the most interesting critical pairs, and show how we join them. For every critical pair, we underline the starting term.

- **Critical Pairs $R_1/(R_1 \cup R_2 \cup R_3 \cup R_4)$:** we only show the critical pairs involving $\pi_1(\_)$ (the critical pairs with $\pi_2(\_)$ are similar), and for $\mathsf{eq}(\_,\_)$. The critical pairs involving $\mathsf{dec}(\_,\_)$ are similar to the critical pairs involving $\pi_1(\_)$, and the critical pairs for $\mathbf{0}(\_)$ are trivial.

$$\begin{aligned} \mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v \ &\leftarrow^2\ \mathsf{if}\ b\ \mathsf{then}\ \pi_1(\langle u\,,\,w\rangle)\ \mathsf{else}\ \pi_1(\langle v\,,\,w\rangle) \ \leftarrow \\ &\qquad \underline{\pi_1(\langle \mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v\,,\,w\rangle)} \ \to\ \mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v \end{aligned}$$

$$\begin{aligned} w \leftarrow \mathsf{if}\ b\ \mathsf{then}\ w\ \mathsf{else}\ w \leftarrow^2 \mathsf{if}\ b\ \mathsf{then}\ \pi_1(\langle w\,,\,u\rangle)\ \mathsf{else}\ \pi_2(\langle w\,,\,v\rangle) \ &\leftarrow \\ \underline{\pi_1(\langle w\,,\,\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v\rangle)} \ &\to\ w \end{aligned}$$

$$\begin{aligned} \mathsf{true} \ &\leftarrow\ \underline{\mathsf{eq}(\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v, \mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v)} \\ &\to\ \mathsf{if}\ b\ \mathsf{then}\ \mathsf{eq}(u, \mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v)\ \mathsf{else}\ \mathsf{eq}(v, \mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v) \\ &\to\ \mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ \mathsf{eq}(u,u)\ \mathsf{else}\ \mathsf{eq}(u,v))\ \mathsf{else}\ \mathsf{eq}(v, \mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v) \\ &\to\ \mathsf{if}\ b\ \mathsf{then}\ \mathsf{eq}(u,u)\ \mathsf{else}\ \mathsf{eq}(v, \mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v) \\ &\to\ \mathsf{if}\ b\ \mathsf{then}\ \mathsf{true}\ \mathsf{else}\ \mathsf{eq}(v, \mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v) \\ &\to^*\ \mathsf{if}\ b\ \mathsf{then}\ \mathsf{true}\ \mathsf{else}\ \mathsf{true} \\ &\to\ \mathsf{true} \end{aligned}$$

- **Critical Pairs R_2/R_2:** we assume that $b \succ_u^{\mathsf{lpo}} c$. The other possible orderings are handled in the same fashion.

$$\begin{aligned} \mathsf{if}\ c\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ f(u,s)\ \mathsf{else}\ f(v,s))\ \mathsf{else}\ (\mathsf{if}\ b\ \mathsf{then}\ f(u,t)\ \mathsf{else}\ f(v,t)) \ &\leftarrow^2 \\ \mathsf{if}\ c\ \mathsf{then}\ f(\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v, s)\ \mathsf{else}\ f(\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v, t) \ &\leftarrow \\ \underline{f(\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v, \mathsf{if}\ c\ \mathsf{then}\ s\ \mathsf{else}\ t)} & \\ \to\ \mathsf{if}\ b\ \mathsf{then}\ f(u, \mathsf{if}\ c\ \mathsf{then}\ s\ \mathsf{else}\ t)\ \mathsf{else}\ f(v, \mathsf{if}\ c\ \mathsf{then}\ s\ \mathsf{else}\ t) & \\ \to^2\ \mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ c\ \mathsf{then}\ f(u,s)\ \mathsf{else}\ f(u,t))\ \mathsf{else}\ (\mathsf{if}\ c\ \mathsf{then}\ f(v,s)\ \mathsf{else}\ f(v,t)) & \\ \to^*\ \mathsf{if}\ c\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ f(u,s)\ \mathsf{else}\ f(v,s))\ \mathsf{else}\ (\mathsf{if}\ b\ \mathsf{then}\ f(u,t)\ \mathsf{else}\ f(v,t)) & \end{aligned}$$

- **Critical Pairs R_2/R_3:**

$$f(u,w) \leftarrow \underline{f(\mathsf{if}\ \mathsf{true}\ \mathsf{then}\ u\ \mathsf{else}\ v, w)} \to \mathsf{if}\ \mathsf{true}\ \mathsf{then}\ f(u,w)\ \mathsf{else}\ f(v,w) \to f(u,w)$$

$$f(u,v) \leftarrow \underline{f(\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ u, v)} \to \mathsf{if}\ b\ \mathsf{then}\ f(u,v)\ \mathsf{else}\ f(u,v) \to f(u,v)$$

$$\begin{aligned} &\mathsf{if}\ b\ \mathsf{then}\ f(u,s)\ \mathsf{else}\ f(w,s) && \leftarrow \\ &f(\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ w, s) && \leftarrow \\ &\underline{f(\mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v)\ \mathsf{else}\ w, s)} & \end{aligned}$$

$$\begin{aligned}
&\to \mathsf{if}\ b\ \mathsf{then}\ f(\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v, s)\ \mathsf{else}\ f(w, s)\\
&\to \mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ f(u, s)\ \mathsf{else}\ f(v, s))\ \mathsf{else}\ f(w, s)\\
&\to \mathsf{if}\ b\ \mathsf{then}\ f(u, s)\ \mathsf{else}\ f(w, s)
\end{aligned}$$

- **Critical Pairs R_2/R_4:** we assume that $a \succ_u^{\mathsf{lpo}} b \succ_u^{\mathsf{lpo}} c \succ_u^{\mathsf{lpo}} d$. The other possible orderings are handled in the same fashion.

$$\begin{aligned}
&\mathsf{if}\ d\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ u\ \mathsf{else}\ v)\ \mathsf{else}\ w)\ \mathsf{else}\ (\mathsf{if}\ c\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ u\ \mathsf{else}\ v)\ \mathsf{else}\ w) && \leftarrow^*\\
&\mathsf{if}\ a\ \mathsf{then}\ \mathsf{if}\ d\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ w)\ \mathsf{else}\ (\mathsf{if}\ c\ \mathsf{then}\ u\ \mathsf{else}\ w) && \leftarrow^2\\
&\quad \mathsf{else}\ \mathsf{if}\ d\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ v\ \mathsf{else}\ w)\ \mathsf{else}\ (\mathsf{if}\ c\ \mathsf{then}\ v\ \mathsf{else}\ w)\\
&\mathsf{if}\ a\ \mathsf{then}\ (\mathsf{if}\ (\mathsf{if}\ d\ \mathsf{then}\ b\ \mathsf{else}\ c)\ \mathsf{then}\ u\ \mathsf{else}\ w)\ \mathsf{else}\ (\mathsf{if}\ (\mathsf{if}\ d\ \mathsf{then}\ b\ \mathsf{else}\ c)\ \mathsf{then}\ v\ \mathsf{else}\ w) && \leftarrow\\
&\underline{\mathsf{if}\ (\mathsf{if}\ d\ \mathsf{then}\ b\ \mathsf{else}\ c)\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ u\ \mathsf{else}\ v)\ \mathsf{else}\ w}\\
\to\ &\mathsf{if}\ d\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ u\ \mathsf{else}\ v)\ \mathsf{else}\ w)\ \mathsf{else}\ (\mathsf{if}\ c\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ u\ \mathsf{else}\ v)\ \mathsf{else}\ w)
\end{aligned}$$

- **Critical Pairs R_3/R_3:**

$$u \leftarrow \underline{\mathsf{if}\ \mathsf{true}\ \mathsf{then}\ u\ \mathsf{else}\ u} \to u$$

$$\begin{aligned}
u \leftarrow \mathsf{if}\ \mathsf{true}\ \mathsf{then}\ u\ \mathsf{else}\ v\ &\leftarrow \underline{\mathsf{if}\ \mathsf{true}\ \mathsf{then}\ (\mathsf{if}\ \mathsf{true}\ \mathsf{then}\ u\ \mathsf{else}\ v)\ \mathsf{else}\ w}\\
&\to \mathsf{if}\ \mathsf{true}\ \mathsf{then}\ u\ \mathsf{else}\ w \to u
\end{aligned}$$

$$\begin{aligned}
\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v\ &\leftarrow \underline{\mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v)\ \mathsf{else}\ (\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v)}\\
&\to \mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ (\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v) \to \mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v
\end{aligned}$$

- **Critical Pairs R_3/R_4:**

$$\begin{aligned}
&\mathsf{if}\ a\ \mathsf{then}\ u\ \mathsf{else}\ v && \leftarrow\\
&\underline{\mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ u\ \mathsf{else}\ v)\ \mathsf{else}\ (\mathsf{if}\ a\ \mathsf{then}\ u\ \mathsf{else}\ v)}\\
\to\ &\mathsf{if}\ a\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ (\mathsf{if}\ a\ \mathsf{then}\ u\ \mathsf{else}\ v))\ \mathsf{else}\ (\mathsf{if}\ b\ \mathsf{then}\ v\ \mathsf{else}\ (\mathsf{if}\ a\ \mathsf{then}\ u\ \mathsf{else}\ v))\\
\to^2\ &\mathsf{if}\ a\ \mathsf{then}\ \mathsf{if}\ a\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ u)\ \mathsf{else}\ (\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v)\\
&\quad \mathsf{else}\ \mathsf{if}\ a\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ v\ \mathsf{else}\ u)\ \mathsf{else}\ (\mathsf{if}\ b\ \mathsf{then}\ v\ \mathsf{else}\ v)\\
\to^2\ &\mathsf{if}\ a\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ u)\ \mathsf{else}\ (\mathsf{if}\ b\ \mathsf{then}\ v\ \mathsf{else}\ v)\\
\to^2\ &\mathsf{if}\ a\ \mathsf{then}\ u\ \mathsf{else}\ v
\end{aligned}$$

- **Critical Pairs R_4/R_4:** we assume that $a \succ_u^{\mathsf{lpo}} b \succ_u^{\mathsf{lpo}} c$. The other possible orderings are handled in the same fashion.

$$\begin{aligned}
&\mathsf{if}\ c\ \mathsf{then}\ \mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ u\ \mathsf{else}\ s)\ \mathsf{else}\ (\mathsf{if}\ a\ \mathsf{then}\ v\ \mathsf{else}\ s) && \leftarrow^2\\
&\quad \mathsf{else}\ \mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ u\ \mathsf{else}\ t)\ \mathsf{else}\ (\mathsf{if}\ a\ \mathsf{then}\ v\ \mathsf{else}\ t)\\
&\mathsf{if}\ c\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v)\ \mathsf{else}\ s)\ \mathsf{else}\ (\mathsf{if}\ a\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ v\ \mathsf{else}\ u)\ \mathsf{else}\ t) && \leftarrow\\
&\underline{\mathsf{if}\ a\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v)\ \mathsf{else}\ (\mathsf{if}\ c\ \mathsf{then}\ s\ \mathsf{else}\ t)}\\
\to\ &\mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ u\ \mathsf{else}\ (\mathsf{if}\ c\ \mathsf{then}\ s\ \mathsf{else}\ t))\ \mathsf{else}\ (\mathsf{if}\ a\ \mathsf{then}\ v\ \mathsf{else}\ (\mathsf{if}\ c\ \mathsf{then}\ s\ \mathsf{else}\ t))\\
\to^2\ &\mathsf{if}\ b\ \mathsf{then}\ \mathsf{if}\ c\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ u\ \mathsf{else}\ s)\ \mathsf{else}\ (\mathsf{if}\ a\ \mathsf{then}\ u\ \mathsf{else}\ t)\\
&\quad \mathsf{else}\ \mathsf{if}\ c\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ v\ \mathsf{else}\ s)\ \mathsf{else}\ (\mathsf{if}\ a\ \mathsf{then}\ v\ \mathsf{else}\ t)\\
\to^*\ &\mathsf{if}\ c\ \mathsf{then}\ \mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ u\ \mathsf{else}\ s)\ \mathsf{else}\ (\mathsf{if}\ a\ \mathsf{then}\ v\ \mathsf{else}\ s)\\
&\quad \mathsf{else}\ \mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ u\ \mathsf{else}\ t)\ \mathsf{else}\ (\mathsf{if}\ a\ \mathsf{then}\ v\ \mathsf{else}\ t)
\end{aligned}$$

Termination To prove termination, we let $\mathcal{F}_{\mathsf{term}}$ be the signature $\mathcal{F}$ to which we added a symbol $\mathsf{if}_b(\ ,\ )$ for every if-free $R_{\leq 3}$-irreducible conditional b:

$$\mathcal{F}_{\mathsf{term}} = \mathcal{F} \cup \left\{\mathsf{if}_b(\ ,\ ) \mid b \in \mathcal{T}(\mathcal{F}_{\setminus \mathsf{if}}, \mathcal{N}), b \text{ is a } R_{\leq 3}\text{-irreducible conditional}\right\}$$

$$\rightarrow_{R'_2} \left\{ f(\vec{u}, \mathsf{if}_b(x\,,\,y), \vec{v}) \rightarrow \mathsf{if}_b(f(\vec{u}, x, \vec{v})\,,\, f(\vec{u}, y, \vec{v})) \quad (f \in \mathcal{F}_{\setminus \mathsf{if}}) \right.$$

$$\rightarrow_{R'_3} \begin{cases} \mathsf{if}_{\mathsf{true}}(x\,,\,y) \rightarrow x \\ \mathsf{if}_{\mathsf{false}}(x\,,\,y) \rightarrow y \\ \mathsf{if}_b(x\,,\,x) \rightarrow x \\ \mathsf{if}_b((\mathsf{if}_b(x\,,\,y))\,,\,z) \rightarrow \mathsf{if}_b(x\,,\,z) \\ \mathsf{if}_b(x\,,\,(\mathsf{if}_b(y\,,\,z))) \rightarrow \mathsf{if}_b(x\,,\,z) \end{cases}$$

$$\rightarrow_{R_4^0} \begin{cases} \mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ x\ \mathsf{else}\ y)\ \mathsf{else}\ z \rightarrow \mathsf{if}\ a\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ z)\ \mathsf{else}\ (\mathsf{if}\ b\ \mathsf{then}\ y\ \mathsf{else}\ z) \\ \qquad\qquad (b \succ a,\ a,b \text{ not if-free or not } R_{\leq 3}\text{-irreducible}) \\ \mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ (\mathsf{if}\ a\ \mathsf{then}\ y\ \mathsf{else}\ z) \rightarrow \mathsf{if}\ a\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ y)\ \mathsf{else}\ (\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ z) \\ \qquad\qquad (b \succ a,\ a,b \text{ not if-free or not } R_{\leq 3}\text{-irreducible}) \end{cases}$$

$$\rightarrow_{R_4^1} \begin{cases} \mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}_a(x\,,\,y))\ \mathsf{else}\ z \rightarrow \mathsf{if}_a((\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ z)\,,\,(\mathsf{if}\ b\ \mathsf{then}\ y\ \mathsf{else}\ z)) \\ \qquad\qquad (b \text{ not if-free or not } R_{\leq 3}\text{-irreducible}) \\ \mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ (\mathsf{if}_a(y\,,\,z)) \rightarrow \mathsf{if}_a((\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ y)\,,\,(\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ z)) \\ \qquad\qquad (b \text{ not if-free or not } R_{\leq 3}\text{-irreducible}) \end{cases}$$

$$\rightarrow_{R_4^2} \begin{cases} \mathsf{if}_b((\mathsf{if}_a(x\,,\,y))\,,\,z) \rightarrow \mathsf{if}_a((\mathsf{if}_b(x\,,\,z))\,,\,(\mathsf{if}_b(y\,,\,z))) \quad (b \succ_u a) \\ \mathsf{if}_b(x\,,\,(\mathsf{if}_a(y\,,\,z))) \rightarrow \mathsf{if}_a((\mathsf{if}_b(x\,,\,y))\,,\,(\mathsf{if}_b(x\,,\,z))) \quad (b \succ_u a) \end{cases}$$

$$\rightarrow_{R^i} \left\{ \mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v \rightarrow \mathsf{if}_b(u\,,\,v) \quad (b \text{ if-free and } R_{\leq 3}\text{-irreducible}) \right.$$

Figure 5.4: The Relations $\rightarrow_{R'_2}, \rightarrow_{R'_3}, \rightarrow_{R_4^0}, \rightarrow_{R_4^1}, \rightarrow_{R_4^2}$ and $\rightarrow_{R^i}$ used for termination

This yields an infinite countable signature. We extend the precedence $\succ_f$ to $\mathcal{F}_{\mathsf{term}}$ by having the function symbols $\{\mathsf{if}_b(\,,\,)\}$ be smaller than all the other function symbols, and $\mathsf{if}_b(\,,\,) \succ_f \mathsf{if}_a(\,,\,)$ if and only if $b \succ_u a$. Observe that the extended precedence is still a total order.

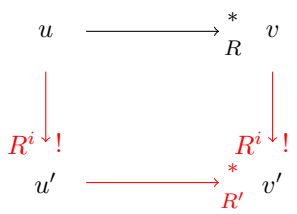
We then consider the term rewriting system $\rightarrow_{R'}$ on $\mathcal{T}(\mathcal{F}_{\mathsf{term}}, \mathcal{N})$, defined by removing $\rightarrow_{R_4}$ from $\rightarrow_R$ and adding all the rules in Figure 5.4:

$$\rightarrow_{R'} = \rightarrow_{R_1} \cup \rightarrow_{R_2} \cup \rightarrow_{R'_2} \cup \rightarrow_{R_3} \cup \rightarrow_{R'_3} \cup \rightarrow_{R_4^0} \cup \rightarrow_{R_4^1} \cup \rightarrow_{R_4^2} \cup \rightarrow_{R^i}$$

One can easily (but tediously) check that $\succ$ is compatible with $\rightarrow_{R'}$: the only non-trivial cases are the cases in $\rightarrow_{R_2}$ (the first rule is decreasing because $f \succ_f \mathsf{if\_then\_else\_}$, the second rule using the lexicographic order), in $\rightarrow_{R'_2}$ (same arguments than for R_2) and the cases in $\rightarrow_{R_4^0}, \rightarrow_{R_4^1}, \rightarrow_{R_4^2}$ (where we use the side conditions $b \succ a$, $b \succ_u a \ldots$).

Since $\succ$ is a lexicographic path ordering we know that it is total and well-founded on ground-terms. Therefore $\rightarrow_{R'}$ is a terminating term rewriting systems on ground terms.

To conclude, one just has to observe that for every ground terms u, v and integer n, if $u \rightarrow_R^{(n)} v$ then there exist u', v' such that $u \rightarrow_{R^i}^! u'$, $v \rightarrow_{R^i}^! v'$ and $u' \rightarrow_{R'}^{(\geq n)} v'$. That is, we have the following diagram (black edges stand for universal quantifications, red edges for existentials):



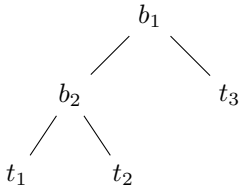
This result can be proved by induction on n. Since $\to_{R'}$ is terminating on ground terms, and since any infinite sequence for $\to_R$ can be translated into an infinite sequence for $\to_{R'}$, it follows that $\to_R$ is terminating on ground terms. ■

The normal form of term t by $\to_{R^{\succ_u}}$ is of the form $C[\vec{b} \diamond \vec{u}]$, where $\vec{b}, \vec{u}$ are if-free terms in R-normal form. We are going to call $\vec{b}$ the conditionals of $t \downarrow_{R^{\succ_u}}$, and $\vec{u}$ its leaves.

Definition 5.6. An if-free term is a term that does not use the if_then_else_ function symbols. For every if-free terms $\vec{b}, \vec{u}$, if t is the term $C[\vec{b} \diamond \vec{u}]$ then we let $\mathsf{cond\text{-}st}(t)$ be the set of conditionals $\vec{b}$, and $\mathsf{leave\text{-}st}(t)$ be the set of terms $\vec{u}$.

Example 5.3. Let b_1, b_2, t_1, t_2, t_3 be if-free terms, and let s be the following term (we give the labelled tree representation of s on the right):

$$\mathsf{if}\ b_1\ \mathsf{then}\ \mathsf{if}\ b_2\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2\ \mathsf{else}\ t_3$$



Then $\mathsf{cond\text{-}st}(s) = \{b_1, b_2\}$ and $\mathsf{leave\text{-}st}(s) = \{t_1, t_2, t_3\}$. □

Interestingly, the leaves and conditionals of $t \downarrow_{R^{\succ_u}}$ do not depend on the order $\succ_u$ on ground conditionals. Formally:

Proposition 5.1. *Let $\succ_u$ and $\succ'_u$ be two total orderings on if-free $R_{\leq 3}$-irreducible conditionals. Then for every ground term t we have:*

$$\mathsf{leave\text{-}st}(t \downarrow_{R^{\succ_u}}) = \mathsf{leave\text{-}st}(t \downarrow_{R^{\succ'_u}}) \qquad \textit{and} \qquad \mathsf{cond\text{-}st}(t \downarrow_{R^{\succ_u}}) = \mathsf{cond\text{-}st}(t \downarrow_{R^{\succ'_u}})$$

Proof. Let C, C' be two if-contexts such that $t \downarrow_{R^{\succ_u}} \equiv C[\vec{b} \diamond \vec{u}]$ and $t \downarrow_{R^{\succ'_u}} \equiv C'[\vec{b}' \diamond \vec{u}']$ where:

$$\vec{b} = \mathsf{leave\text{-}st}(t \downarrow_{R^{\succ_u}}) \qquad \vec{u} = \mathsf{cond\text{-}st}(t \downarrow_{R^{\succ_u}}) \qquad \vec{b}' = \mathsf{leave\text{-}st}(t \downarrow_{R^{\succ'_u}}) \qquad \vec{u}' = \mathsf{cond\text{-}st}(t \downarrow_{R^{\succ'_u}})$$

We know that $C[\vec{b} \diamond \vec{u}] \to^*_{R^{\succ'_u}} C'[\vec{b}' \diamond \vec{u}']$. Since the terms $\vec{b}$, $\vec{u}$, $\vec{b}'$ and $\vec{u}'$ are if-free and in R-normal form, we can only apply the rules:

$$\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ x \to x$$
$$\mathsf{if}\ \mathsf{true}\ \mathsf{then}\ x\ \mathsf{else}\ y \to x$$
$$\mathsf{if}\ \mathsf{false}\ \mathsf{then}\ x\ \mathsf{else}\ y \to y$$
$$\mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ y)\ \mathsf{else}\ z \to \mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ z$$
$$\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ (\mathsf{if}\ b\ \mathsf{then}\ y\ \mathsf{else}\ z) \to \mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ z$$
$$\mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ x\ \mathsf{else}\ y)\ \mathsf{else}\ z \to \mathsf{if}\ a\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ z)\ \mathsf{else}\ (\mathsf{if}\ b\ \mathsf{then}\ y\ \mathsf{else}\ z) \quad (\text{when } b \succ_u^{\mathsf{lpo}} a)$$
$$\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ (\mathsf{if}\ a\ \mathsf{then}\ y\ \mathsf{else}\ z) \to \mathsf{if}\ a\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ y)\ \mathsf{else}\ (\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ z) \quad (\text{when } b \succ_u^{\mathsf{lpo}} a)$$

Moreover, if a term $C_1[\vec{a}_1 \diamond \vec{v}_1]$ can be rewritten in one step into $C_2[\vec{a}_2 \diamond \vec{v}_2]$ using one of the rules above then $\vec{a}_2 \subseteq \vec{a}_1$ and $\vec{v}_2 \subseteq \vec{v}_1$. Hence, by induction, $\vec{b}' \subseteq \vec{b}$ and $\vec{u}' \subseteq \vec{u}$. Similarly, since $C'[\vec{b}' \diamond \vec{u}'] \to^*_{R^{\succ_u}} C[\vec{b} \diamond \vec{u}]$, we get that $\vec{b} \subseteq \vec{b}'$ and $\vec{u} \subseteq \vec{u}'$. We deduce that $\vec{b} \equiv \vec{b}'$ and $\vec{u} \equiv \vec{u}'$. ■

By consequence, for any term u, the sets $\mathsf{leave\text{-}st}(t \downarrow_R)$ and $\mathsf{cond\text{-}st}(t \downarrow_R)$ are always well-defined, by taking an arbitrary ordering of if-free $R_{\leq 3}$-irreducible conditionals.

5.4 The CCA$_2$ Axioms

We define and prove correct a recursive set of axioms for an IND-CCA$_2$ encryption scheme. For the sake of simplicity, we first ignore all length constraints. We explain how length constraints are added and handled to the logic in Section 5.4.2.

Multi-Users IND-CCA$_2$ Game Consider the following multi-users IND-CCA$_2$ game: the adversary receives n public-keys. For each key pk_i, he has access to a left-right oracle $\mathcal{O}_{\mathsf{LR}}(\mathsf{pk}_i, b)$ that takes two messages m_0, m_1 as input and returns $\{m_b\}^{\mathsf{n}_r}_{\mathsf{pk}_i}$, where b is an internal random bit uniformly drawn at the beginning by the challenger (the same b is used for all left-right oracles) and n_r is a fresh nonce. Moreover, for all key pairs $(\mathsf{pk}_i, \mathsf{sk}_i)$, the adversary has access to an sk_i decryption oracle $\mathcal{O}_{\mathsf{dec}}(\mathsf{sk}_i)$, but cannot call $\mathcal{O}_{\mathsf{dec}}(\mathsf{sk}_i)$ on a cipher-text returned by $\mathcal{O}_{\mathsf{LR}}(\mathsf{pk}_i, b)$ (to do this, the two oracles use a shared memory where all encryption requests are logged). The advantage of an adversary against this game and the multi-user IND-CCA$_2$ security are defined as usual.

It is known that if an encryption scheme is IND-CCA$_2$ then it is also multi-users IND-CCA$_2$ (see [BBM00]). Therefore, we allow multiple key pairs to appear in the CCA$_2$ axioms, and multiple encryptions over different terms using the same public key (each encryption corresponds to one call to a left-right oracle).

Decryption Guards If we want the following to hold in any computational model

$$\mathsf{dec}\big(\underbrace{t\big[\{u_1\}^{\mathsf{n}_1}_{\mathsf{pk}}, \ldots, \{u_n\}^{\mathsf{n}_n}_{\mathsf{pk}}\big]}_{s}, \mathsf{sk}\big) \sim \mathsf{dec}\big(\underbrace{t\big[\{v_1\}^{\mathsf{n}_1}_{\mathsf{pk}}, \ldots, \{v_n\}^{\mathsf{n}_n}_{\mathsf{pk}}\big]}_{s'}, \mathsf{sk}\big)$$

then we need to make sure that s is different from all $\{u_i\}^{\mathsf{n}_i}_{\mathsf{pk}}$ and that s' is different from all $\{v_i\}^{\mathsf{n}_i}_{\mathsf{pk}}$. This is done by introducing all the unwanted equalities in if_then_else_ tests and making sure that we are in the else branch of all these tests, so as to have a "safe call" to the decryption oracle. Moreover, the adversary is allowed to use values obtained from previous calls to the decryption oracle in future calls.

To do this, we use the following function:

Definition 5.7. We define the function else^* by induction:

$$\begin{aligned}
&\mathsf{else}^*(\emptyset, x) \equiv x\\
&\mathsf{else}^*\left((\mathsf{eq}(a,b)) :: \Gamma, x\right) \equiv \mathsf{if}\ \mathsf{eq}(a,b)\ \mathsf{then}\ \mathbf{0}(x)\ \mathsf{else}\ \mathsf{else}^*(\Gamma, x)
\end{aligned}$$

Example 5.4. Let $u \equiv t[\{v_1\}^{\mathsf{n}^1_r}_{\mathsf{pk}}, \{v_2\}^{\mathsf{n}^2_r}_{\mathsf{pk}}]$. Then:

$$\begin{aligned}
&\mathsf{else}^*\big(\big(\mathsf{eq}(u, \{v_1\}^{\mathsf{n}^1_r}_{\mathsf{pk}}), \mathsf{eq}(u, \{v_2\}^{\mathsf{n}^2_r}_{\mathsf{pk}})\big), \mathsf{dec}(u, \mathsf{sk})\big) \equiv\\
&\qquad \mathsf{if}\ \mathsf{eq}(u, \{v_1\}^{\mathsf{n}^1_r}_{\mathsf{pk}})\ \mathsf{then}\ \mathbf{0}(\mathsf{dec}(u, \mathsf{sk}))\ \mathsf{else}\ \mathsf{if}\ \mathsf{eq}(u, \{v_2\}^{\mathsf{n}^2_r}_{\mathsf{pk}})\ \mathsf{then}\ \mathbf{0}(\mathsf{dec}(u, \mathsf{sk}))\ \mathsf{else}\ \mathsf{dec}(u, \mathsf{sk})
\end{aligned}$$

Morally, this represents a safe call to the decryption oracle. □

Definition of CCA$_2$ We use the following notations: for any finite set $\mathcal{K}$ of valid private keys, $\mathcal{K} \sqsubseteq_d \vec{u}$ holds if for all $\mathsf{sk} \in \mathcal{K}$, the secret key sk appears only in decryption position in $\vec{u}$; $\mathsf{nodec}(\mathcal{K}, \vec{u})$ denotes that for all $\mathsf{sk}(\mathsf{n}) \in \mathcal{K}$, the only occurrences of n are in subterms $\mathsf{pk}(\mathsf{n})$; $\mathsf{hidden\text{-}rand}(\vec{r}; \vec{u})$ denotes that for all $\mathsf{n}_r \in \vec{r}$, n_r appears only in encryption randomness position and is not used with two distinct plaintexts.

We are now going to define by induction the CCA$_2$ axiom. In order to do this we define by induction a binary relation $R^{\mathcal{K}}_{\mathrm{CCA}^a_2}$ on CCA$_2$ executions, where $\mathcal{K}$ is the finite set of private keys used in the terms (corresponding to the public keys sent by the challenger).

Definition 5.8. Let $\mathcal{K}$ be a set of private keys. $(\phi, \mathcal{X}_{\mathsf{enc}}, \mathcal{X}_{\mathsf{dec}}, \sigma_{\mathsf{rand}}, \theta_{\mathsf{enc}}, \lambda_{\mathsf{dec}})$ is a CCA$_2$ execution if:

- ϕ is a vector of ground terms in $\mathcal{T}(\mathcal{F}, \mathcal{N})$.
- $\mathcal{X}_{\mathsf{enc}}$ and $\mathcal{X}_{\mathsf{dec}}$ are two disjoint sets of variables used as handles for, respectively, encryptions and decryptions.
- σ_{rand} is a substitution from $\mathcal{X}_{\mathsf{enc}}$ to $\mathcal{N}$.
- θ_{enc} and λ_{dec} are substitutions from, respectively, $\mathcal{X}_{\mathsf{enc}}$ and $\mathcal{X}_{\mathsf{dec}}$, to ground terms in $\mathcal{T}(\mathcal{F}, \mathcal{N})$.

σ_{rand}, θ_{enc} and λ_{dec} co-domains are the sets of, respectively, encryption randomness, encryption oracle calls and decryption oracle calls in ϕ. Intuitively, we have:

$$(\phi, \mathcal{X}_{\mathsf{enc}}, \mathcal{X}_{\mathsf{dec}}, \sigma_{\mathsf{rand}}, \theta_{\mathsf{enc}}, \lambda_{\mathsf{dec}}) R^{\mathcal{K}}_{\mathrm{CCA}^a_2} (\psi, \mathcal{X}_{\mathsf{enc}}, \mathcal{X}_{\mathsf{dec}}, \sigma'_{\mathsf{rand}}, \theta'_{\mathsf{enc}}, \lambda'_{\mathsf{dec}})$$

when we can build ϕ and ψ using function symbols, matching encryption oracle calls and matching decryption oracle calls.

Definition 5.9. Let $\mathcal{K}$ be a finite set of private keys. We define the binary relation $R^{\mathcal{K}}_{\mathrm{CCA}_2^a}$ by induction:

1. **No Call to the Oracles:** if $\mathcal{K} \sqsubseteq_d \phi$ then $(\phi, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset) R^{\mathcal{K}}_{\mathrm{CCA}_2^a} (\phi, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$ for every sequence ϕ of ground terms in $\mathcal{T}(\mathcal{F}, \mathcal{N})$ such that $\mathsf{nodec}(\mathcal{K}; \phi)$.
2. **Encryption Case:** Let x a fresh variable that does not appear in $\mathcal{X}_{\mathsf{enc}} \cup \mathcal{X}_{\mathsf{dec}}$, sk be a secret key in $\mathcal{K}$ and pk the corresponding public key. Then:

$$\big((\phi, \{u\}^{\mathsf{n}_r}_{\mathsf{pk}}), \mathcal{X}_{\mathsf{enc}} \cup \{x\}, \mathcal{X}_{\mathsf{dec}}, \sigma_{\mathsf{rand}} \cup \{x \mapsto \mathsf{n}_r\}, \theta_{\mathsf{enc}} \cup \{x \mapsto \{u\}^{\mathsf{n}_r}_{\mathsf{pk}}\}, \lambda_{\mathsf{dec}}\big)$$
$$R^{\mathcal{K}}_{\mathrm{CCA}_2^a}\big((\psi, \{v\}^{\mathsf{n}'_r}_{\mathsf{pk}}), \mathcal{X}_{\mathsf{enc}} \cup \{x\}, \mathcal{X}_{\mathsf{dec}}, \sigma'_{\mathsf{rand}} \cup \{x \mapsto \mathsf{n}'_r\}, \theta'_{\mathsf{enc}} \cup \{x \mapsto \{v\}^{\mathsf{n}'_r}_{\mathsf{pk}}\}, \lambda'_{\mathsf{dec}}\big)$$

   if there exist $t, t' \in \mathcal{T}(\mathcal{F}_{\setminus \mathbf{0}}, \mathcal{N}, \mathcal{X}_{\mathsf{enc}})$ such that:
   - $(\phi, \mathcal{X}_{\mathsf{enc}}, \mathcal{X}_{\mathsf{dec}}, \sigma_{\mathsf{rand}}, \theta_{\mathsf{enc}}, \lambda_{\mathsf{dec}}) R^{\mathcal{K}}_{\mathrm{CCA}_2^a} (\psi, \mathcal{X}_{\mathsf{enc}}, \mathcal{X}_{\mathsf{dec}}, \sigma'_{\mathsf{rand}}, \theta'_{\mathsf{enc}}, \lambda'_{\mathsf{dec}})$
   - $u \equiv t\lambda_{\mathsf{dec}}$, $v \equiv t'\lambda'_{\mathsf{dec}}$
   - $\mathsf{nodec}(\mathcal{K}; t, t')$, which ensures that the only decryptions are calls to the oracle.
   - $\mathsf{fresh}(\mathsf{n}_r, \mathsf{n}'_r; \phi, u, \psi, v)$ and $\mathsf{hidden\text{-}rand}(\mathcal{X}_{\mathsf{enc}}\sigma_{\mathsf{rand}} \cup \mathcal{X}_{\mathsf{enc}}\sigma'_{\mathsf{rand}}; \phi, u, \psi, v)$
3. **Decryption Case:** Let $\mathsf{sk} \in \mathcal{K}$, pk the corresponding public key and z be a fresh variable. Then:

$$\big(\,(\phi, \mathsf{else}^*(l, \mathsf{dec}(u, \mathsf{sk}))), \mathcal{X}_{\mathsf{enc}}, \mathcal{X}_{\mathsf{dec}} \cup \{z\}, \sigma_{\mathsf{rand}}, \theta_{\mathsf{enc}}, \lambda_{\mathsf{dec}} \cup \{z \mapsto \mathsf{else}^*(l, \mathsf{dec}(u, \mathsf{sk}))\}\big)$$
$$R^{\mathcal{K}}_{\mathrm{CCA}_2^a}\big(\,(\psi, \mathsf{else}^*(l', \mathsf{dec}(v, \mathsf{sk}))), \mathcal{X}_{\mathsf{enc}}, \mathcal{X}_{\mathsf{dec}} \cup \{z\}, \sigma'_{\mathsf{rand}}, \theta'_{\mathsf{enc}}, \lambda'_{\mathsf{dec}} \cup \{z \mapsto \mathsf{else}^*(l', \mathsf{dec}(v, \mathsf{sk}))\}\big)$$

   if there exists $t \in \mathcal{T}(\mathcal{F}_{\setminus \mathsf{if}, \mathbf{0}}, \mathcal{N}, \mathcal{X}_{\mathsf{enc}}, \mathcal{X}_{\mathsf{dec}})$ such that:
   - $(\phi, \mathcal{X}_{\mathsf{enc}}, \mathcal{X}_{\mathsf{dec}}, \sigma_{\mathsf{rand}}, \theta_{\mathsf{enc}}, \lambda_{\mathsf{dec}}) R^{\mathcal{K}}_{\mathrm{CCA}_2^a} (\psi, \mathcal{X}_{\mathsf{enc}}, \mathcal{X}_{\mathsf{dec}}, \sigma'_{\mathsf{rand}}, \theta'_{\mathsf{enc}}, \lambda'_{\mathsf{dec}})$
   - $u \equiv t\theta_{\mathsf{enc}}\lambda_{\mathsf{dec}}$ and $v \equiv t\theta'_{\mathsf{enc}}\lambda'_{\mathsf{dec}}$.
   - Consider the set $\mathcal{Y}_u$ of variables $x \in \mathcal{X}_{\mathsf{enc}}$ such that the encryption binded to x directly appears in u, i.e. appears outside of another encryption. That is, x must appear in the term u where we substituted every encryption $\{\_\}^{\mathsf{n}_x}_{\mathsf{pk}} \in \mathsf{codom}(\theta_{\mathsf{enc}})$ by $\{0\}^{\mathsf{n}_x}_{\mathsf{pk}}$:

$$x\sigma_{\mathsf{rand}} \in u\big\{\{0\}^{\mathsf{n}_x}_{\mathsf{pk}} / \{\_\}^{\mathsf{n}_x}_{\mathsf{pk}} \mid \{\_\}^{\mathsf{n}_x}_{\mathsf{pk}} \in \mathsf{codom}(\theta_{\mathsf{enc}})\big\} \downarrow_R$$

     Then l is the sequence of guards $l \equiv (\mathsf{eq}(u, y_1), \ldots, \mathsf{eq}(u, y_m))$ where $(y_1, \ldots, y_m) = \mathsf{sort}(\mathcal{Y}_u\theta_{\mathsf{enc}})$. Similarly, $l' \equiv (\mathsf{eq}(v, y'_1), \ldots, \mathsf{eq}(v, y'_m))$ where $(y'_1, \ldots, y'_m) = \mathsf{sort}(\mathcal{Y}_u\theta'_{\mathsf{enc}})$[2].
   - $\mathsf{nodec}(\mathcal{K}; t)$ and $\mathsf{hidden\text{-}rand}(\mathcal{X}_{\mathsf{enc}}\sigma_{\mathsf{rand}} \cup \mathcal{X}_{\mathsf{enc}}\sigma'_{\mathsf{rand}}; \phi, u, \psi, v)$

where sort is a deterministic function sorting terms according to an arbitrary linear order.

Remark 5.4. In the decryption case, we add a guard only for encryption that appear directly in u. Without this restriction, we would add one guard $\mathsf{eq}(u, x\theta_{\mathsf{enc}})$ for every $x \in \mathcal{X}_{\mathsf{enc}}$ such that $x\theta_{\mathsf{enc}}$ is an encryption using public-key pk.

For example, if $\mathcal{X}_{\mathsf{enc}} = \{x_0, x_1, x_2\}$ and $\theta_{\mathsf{enc}} = \{x_0 \mapsto \alpha_0, x_1 \mapsto \alpha_1, x_2 \mapsto \alpha_2\}$ where:

$$\alpha_0 \mapsto \{m_0\}^{\mathsf{n}_0}_{\mathsf{pk}} \qquad \alpha_1 \mapsto \{m_1\}^{\mathsf{n}_1}_{\mathsf{pk}} \qquad \alpha_2 \mapsto \{\alpha_1\}^{\mathsf{n}_2}_{\mathsf{pk}}$$

then to guard $\mathsf{dec}(g(\alpha_2), \mathsf{sk})$, we need to add three guards, $\mathsf{eq}(g(\alpha_2), \alpha_0)$, $\mathsf{eq}(g(\alpha_2), \alpha_1)$ and $\mathsf{eq}(g(\alpha_2), \alpha_2)$. This yields the term:

$$\begin{array}{ll}
\mathsf{if} & \mathsf{eq}(g(\alpha_2), \alpha_0) \ \mathsf{then}\ \mathbf{0}(\mathsf{dec}(g(\alpha_2), \mathsf{sk})) \\
\mathsf{else\ if} & \mathsf{eq}(g(\alpha_2), \alpha_1) \ \mathsf{then}\ \mathbf{0}(\mathsf{dec}(g(\alpha_2), \mathsf{sk})) \\
\mathsf{else\ if} & \mathsf{eq}(g(\alpha_2), \alpha_2) \ \mathsf{then}\ \mathbf{0}(\mathsf{dec}(g(\alpha_2), \mathsf{sk})) \\
\mathsf{else} & \mathsf{dec}(g(\alpha_2), \mathsf{sk})
\end{array}$$

But here, the adversary, represented by the adversarial function g, is computing the query to the decryption oracle using only α_2. Hence, it cannot use α_1, which is hidden by the encryption, nor α_0 which does

[2]Remark that we use, for v, the set $\mathcal{Y}_u$ defined using u. As we will see later, this is not a problem because $\mathcal{Y}_u = \mathcal{Y}_v$.

not appear at all. Therefore, there is no need to add the guards $\mathsf{eq}(g(\alpha_2), \alpha_0)$ and $\mathsf{eq}(g(\alpha_2), \alpha_1)$, since g has a negligible probability of returning α_0 or α_1.

To remove unnecessary guards when building the decryption oracle call $\mathsf{dec}(u, \mathsf{sk})$, we require that $\mathsf{eq}(u, \alpha)$ is added to the list of guards if and only if $\alpha \equiv \{\_\}^{\mathsf{n}}_{\mathsf{pk}}$ appears directly in u. This yields smaller axioms, e.g. the term $\mathsf{dec}(g(\alpha_2), \mathsf{sk})$ is guarded by:

$$\begin{array}{l}\mathsf{if}\ \mathsf{eq}(g(\alpha_2), \alpha_2)\ \mathsf{then}\ \mathbf{0}(\mathsf{dec}(g(\alpha_2), \mathsf{sk}))\\ \qquad\qquad\qquad\qquad\mathsf{else}\ \mathsf{dec}(g(\alpha_2), \mathsf{sk})\end{array}$$

Finally, the sort function is used to ensure that guards are always in the same order, which guarantees that two calls with the same terms are guarded in the same way. □

We can now define the recursive set of axioms CCA_2^a and show their validity. We also state and prove a key property of these axioms.

Definition 5.10. CCA_2^a is the set of unitary axioms $\phi \sim \psi\mu$, where μ is a renaming of names in $\mathcal{N}$ and there exist two CCA_2 executions $\mathcal{Y}, \mathcal{Y}'$ such that:

$$\mathcal{Y} = (\phi, \mathcal{X}_{\mathsf{enc}}, \mathcal{X}_{\mathsf{dec}}, \sigma_{\mathsf{rand}}, \theta_{\mathsf{enc}}, \lambda_{\mathsf{dec}}) \qquad \mathcal{Y}' = (\psi, \mathcal{X}_{\mathsf{enc}}, \mathcal{X}_{\mathsf{dec}}, \sigma'_{\mathsf{rand}}, \theta'_{\mathsf{enc}}, \lambda'_{\mathsf{dec}}) \qquad \mathcal{Y}\ R^{\mathcal{K}}_{\text{CCA}_2^a}\ \mathcal{Y}'$$

In that case, we say that $(\mathcal{Y}, \mathcal{Y}')$ is a valid CCA_2^a application, and $\phi \sim \psi\mu$ is a valid CCA_2^a instance.

Proposition 5.2. *All formulas in* CCA_2^a *are computationally valid if the encryption scheme is* IND-CCA_2.

Proof. First, $\phi \sim \psi\mu$ is computationally valid if and only if $\phi \sim \psi$ is computationally valid. Hence, w.l.o.g. we consider μ empty. Let $\mathcal{M}_{\mathsf{c}}$ be a computational model where the encryption and decryption symbol are interpreted as an IND-CCA_2 encryption scheme. Let $\phi \sim \psi$ be a valid instance of CCA_2^a such that $[\![\phi]\!] \not\approx_{\mathcal{M}_{\mathsf{c}}} [\![\psi]\!]$ i.e. there is a PPTM $\mathcal{A}$ that has a non-negligible advantage of distinguishing these two distributions.

Since $\phi \sim \psi$ is an instance of CCA_2 we know that there exist two CCA_2 executions such that:

$$(\phi, \mathcal{X}_{\mathsf{enc}}, \mathcal{X}_{\mathsf{dec}}, \sigma_{\mathsf{rand}}, \theta_{\mathsf{enc}}, \lambda_{\mathsf{dec}}) R^{\mathcal{K}}_{\text{CCA}_2^a} (\psi, \mathcal{X}_{\mathsf{enc}}, \mathcal{X}_{\mathsf{dec}}, \sigma'_{\mathsf{rand}}, \theta'_{\mathsf{enc}}, \lambda'_{\mathsf{dec}})$$

We are going to build from ϕ and ψ a winning attacker against the multi-user IND-CCA_2 game. This attacker has access to a LR oracle and a decryption oracle for all keys in $\mathcal{K}$. We are going to build by induction on $R^{\mathcal{K}}_{\text{CCA}_2^a}$ a algorithm $\mathcal{B}$ that samples from $[\![\phi]\!]$ or $[\![\psi]\!]$ (depending on the oracles internal bit). The algorithm $\mathcal{B}$ uses a memoisation technique: it builds a store whose keys are subterms of ϕ, ψ already encountered and variable in $\mathcal{X}_{\mathsf{enc}} \cup \mathcal{X}_{\mathsf{dec}}$, and values are elements of the $\mathcal{M}_{\mathsf{c}}$ domain.

1. $(\phi, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset) R^{\mathcal{K}}_{\text{CCA}_2^a} (\phi, \emptyset, \emptyset, \emptyset, \emptyset, \emptyset)$: for every term t in the vector ϕ, $\mathcal{B}$ samples from $[\![t]\!]$ by induction as follows:
   - if t is in the store then $\mathcal{B}$ returns its value.
   - nonce n: $\mathcal{B}$ draws n uniformly at random and stores the drawn value.
     Remark that $\mathsf{nodec}(\mathcal{K}, \phi)$ ensures that n is not used in a secret key sk appearing in $\mathcal{K}$, which we could not compute. If it is a public key pk, either the corresponding secret key sk is such that $\mathsf{sk} \in \mathcal{K}$ and the challenger sent us a random sample from $[\![\mathsf{pk}]\!]$, or sk does not appear in $\mathcal{K}$ and then $\mathcal{B}$ can draw the corresponding key pair itself.
   - $f(t_1, \ldots, t_n)$, then $\mathcal{B}$ inductively samples the function arguments $([\![t_1]\!], \ldots, [\![t_1]\!])$ and then samples from $[\![f]\!]\,([\![t_1]\!], \ldots, [\![t_1]\!])$. $\mathcal{B}$ stores the value at the key $f(t_1, \ldots, t_n)$.
2. **Encryption Case**:

$$\begin{array}{l}\big((\phi, \{u\}^{\mathsf{n}_r}_{\mathsf{pk}}), \mathcal{X}_{\mathsf{enc}} \cup \{x\}, \mathcal{X}_{\mathsf{dec}}, \sigma_{\mathsf{rand}} \cup \{x \mapsto \mathsf{n}_r\}, \theta_{\mathsf{enc}} \cup \{x \mapsto \{u\}^{\mathsf{n}_r}_{\mathsf{pk}}\}, \lambda_{\mathsf{dec}}\big)\\ \qquad R^{\mathcal{K}}_{\text{CCA}_2^a}\big((\psi, \{v\}^{\mathsf{n}'_r}_{\mathsf{pk}}), \mathcal{X}_{\mathsf{enc}} \cup \{x\}, \mathcal{X}_{\mathsf{dec}}, \sigma'_{\mathsf{rand}} \cup \{x \mapsto \mathsf{n}'_r\}, \theta'_{\mathsf{enc}} \cup \{x \mapsto \{v\}^{\mathsf{n}'_r}_{\mathsf{pk}}\}, \lambda'_{\mathsf{dec}}\big)\end{array}$$

   Since we have $\mathsf{fresh}(\mathsf{n}_r, \mathsf{n}'_r; \phi, u, \psi, v)$ we know that the top-level terms do not appear in the store. It is easy to check that $\mathcal{B}$ inductive definition is such that $\mathcal{B}$ store has a value associated with every variable in $\mathcal{X}_{\mathsf{enc}} \cup \mathcal{X}_{\mathsf{dec}}$ and that, if $x \in \mathcal{X}_{\mathsf{enc}}$, then the store value of x is either sampled from $[\![x\theta_{\mathsf{enc}}]\!]$

or from $[\![x\theta'_{\mathsf{enc}}]\!]$ (depending on the challenger internal bit), and that if $x \in \mathcal{X}_{\mathsf{dec}}$ then the store value of x is either sampled from $[\![x\lambda_{\mathsf{dec}}]\!]$ or from $[\![x\lambda'_{\mathsf{dec}}]\!]$ (depending on the challenger internal bit). We also observe that if the challenger internal bit is 0 then for all w:

$$\mathcal{O}_{\mathsf{LR}}(\mathsf{pk}, b)([\![u]\!], [\![v]\!]) = \mathcal{O}_{\mathsf{LR}}(\mathsf{pk}, b)([\![u]\!], w)$$

Similarly if the challenger internal bit is 1 then for all w:

$$\mathcal{O}_{\mathsf{LR}}(\mathsf{pk}, b)([\![u]\!], [\![v]\!]) = \mathcal{O}_{\mathsf{LR}}(\mathsf{pk}, b)(w, [\![v]\!])$$

$\mathcal{B}$ samples two values α, β such that if the challenger internal bit is 0 then α is sampled from $[\![u]\!]$ and if the challenger internal bit is 1 then β is sampled from $[\![v]\!]$. Therefore whatever the challenger internal is bit, $\mathcal{O}_{\mathsf{LR}}(\mathsf{pk}, b)(\alpha, \beta)$ is sampled from $\mathcal{O}_{\mathsf{LR}}(\mathsf{pk}, b)([\![u]\!], [\![v]\!])$:

- α is sampled from $[\![u]\!]$ using the case 1 algorithm. Remark that when we encounter a decryption under $\mathsf{sk}' \in \mathcal{K}$, we know that it was already sampled and can therefore retrieve it from the store.
- similarly, β is sampled from $[\![v]\!]$ using the case 1 algorithm.

The condition $\mathsf{nodec}(\mathcal{K}; t, t')$ ensures that no secret key from $\mathcal{K}$ appears in u, v anywhere else than in decryption positions for already queried oracle calls (which can therefore be retrieved from the store), and the two conditions $\mathsf{fresh}(\mathsf{n}_r, \mathsf{n}'_r; \phi, u, \psi, v)$ and $\mathsf{hidden\text{-}rand}(\mathcal{X}_{\mathsf{enc}}\sigma_{\mathsf{rand}} \cup \mathcal{X}_{\mathsf{enc}}\sigma'_{\mathsf{rand}}; \phi, u, \psi, v)$ ensure that all randomness used by the challenger left-right oracles do not appear anywhere else than in encryption randomness position for the corresponding left-right oracle calls.[3]

We store the result of the left-right oracle call at key x.

3. **Decryption Case**:

$$\begin{aligned}&((\phi, \mathsf{else}^*(l, \mathsf{dec}(u, \mathsf{sk}))), \mathcal{X}_{\mathsf{enc}}, \mathcal{X}_{\mathsf{dec}} \cup \{z\}, \sigma_{\mathsf{rand}}, \theta_{\mathsf{enc}}, \lambda_{\mathsf{dec}} \cup \{z \mapsto \mathsf{else}^*(l, \mathsf{dec}(u, \mathsf{sk}))\})\\ &\quad R^{\mathcal{K}}_{\mathrm{CCA}_2^a}((\psi, \mathsf{else}^*(l', \mathsf{dec}(v, \mathsf{sk}))), \mathcal{X}_{\mathsf{enc}}, \mathcal{X}_{\mathsf{dec}} \cup \{z\}, \sigma'_{\mathsf{rand}}, \theta'_{\mathsf{enc}}, \lambda'_{\mathsf{dec}} \cup \{z \mapsto \mathsf{else}^*(l', \mathsf{dec}(v, \mathsf{sk}))\})\end{aligned}$$

We know that $u \equiv t\theta_{\mathsf{enc}}\lambda_{\mathsf{dec}}$ and $v \equiv t\theta'_{\mathsf{enc}}\lambda'_{\mathsf{dec}}$. $\mathcal{B}$ uses the case 1 algorithm to sample γ from $[\![t\theta_{\mathsf{enc}}\lambda_{\mathsf{dec}}]\!]$ or $[\![t\theta'_{\mathsf{enc}}\lambda'_{\mathsf{dec}}]\!]$ depending on the challenger internal bit. $\mathsf{nodec}(\mathcal{K}; t)$ ensures that no call to the decryption oracles are needed and $\mathsf{hidden\text{-}rand}(\mathcal{X}_{\mathsf{enc}}\sigma_{\mathsf{rand}} \cup \mathcal{X}_{\mathsf{enc}}\sigma'_{\mathsf{rand}}; \phi, u, \psi, v)$ guarantee that the randomness drawn by the challenger for LR oracle encryptions do not appear in t.

Observe that all calls to $\mathcal{O}_{\mathsf{LR}}(\mathsf{pk}, b)$ have already been stored. Let $x_1\theta_{\mathsf{enc}}, \ldots, x_p\theta_{\mathsf{enc}}$ be the corresponding keys in the store. Hence if γ is equal to any of the values stored at keys $x_1\theta_{\mathsf{enc}}, \ldots, x_p\theta_{\mathsf{enc}}$ then $\mathcal{B}$ return $[\![\mathbf{0}]\!](\gamma)$, otherwise $\mathcal{B}$ can call the decryption oracle $\mathcal{O}_{\mathsf{dec}}(\mathsf{sk})$ on γ.

As we observed in Remark 5.4, if the challenger internal bit is 0, checking whether γ is different from the values sampled from $[\![x_1\theta_{\mathsf{enc}}]\!], \ldots, [\![x_p\theta_{\mathsf{enc}}]\!]$ amounts to checking whether γ is different from the values sampled from $[\![y_1]\!], \ldots, [\![y_m]\!]$, except for a negligible number of samplings. Therefore we are sampling from the correct distribution (up to a negligible number of samplings).

Moreover, the set of variables $x \in \mathcal{X}_{\mathsf{enc}}$ such that the encryption binded to x in θ_{enc} appears directly in the *left decryption* u:

$$x\sigma_{\mathsf{rand}} \in u\big\{\{0\}^{\mathsf{n}_x}_{\mathsf{pk}}/\{\_\}^{\mathsf{n}_x}_{\mathsf{pk}} \mid \{\_\}^{\mathsf{n}_x}_{\mathsf{pk}} \in \mathsf{codom}(\theta_{\mathsf{enc}})\big\} \downarrow_R$$

is exactly the set of variables x such that the encryption binded to x in θ'_{enc} appears directly in the *right decryption* v:

$$x\sigma_{\mathsf{rand}} \in v\big\{\{0\}^{\mathsf{n}_x}_{\mathsf{pk}}/\{\_\}^{\mathsf{n}_x}_{\mathsf{pk}} \mid \{\_\}^{\mathsf{n}_x}_{\mathsf{pk}} \in \mathsf{codom}(\theta'_{\mathsf{enc}})\big\} \downarrow_R$$

Hence, if the internal bit is 1 then checking whether γ is different from the values sampled from $[\![x_1\theta'_{\mathsf{enc}}]\!], \ldots, [\![x_p\theta'_{\mathsf{enc}}]\!]$ amounts to checking whether γ is different from the values sampled from $[\![y'_1]\!], \ldots, [\![y'_m]\!]$, except for a negligible number of samplings.

We store the result at key z.

The attacker against the multi-user IND-CCA$_2$ game simply returns $\mathcal{A}(\mathcal{B})$. Since $\mathcal{B}$ samples either from $[\![\phi]\!]$ if $b = 0$ or from $[\![\psi]\!]$ if $b = 1$ (up to a negligible number of samplings), and since $\mathcal{A}$ has a non-negligible advantage of distinguishing $[\![\phi]\!]$ from $[\![\psi]\!]$ we know that the attacker has a non-negligible advantage against the multi-user IND-CCA$_2$ game. ■

[3] We omit for now the length check, which is dealt with later.

5.4.1 Closure Under Restr

To close our logic under Restr, we need the unitary axioms to be closed. Therefore, we let CCA_2 be the closure of CCA_2^a under Restr.

Definition 5.11. CCA_2 is the set of formula $\phi \sim \psi$ such that we have the derivation:

$$\dfrac{\dfrac{}{\phi' \sim \psi'}\ \text{CCA}_2^a}{\phi \sim \psi}\ \mathsf{Restr}$$

The main contribution of this sub-section, given below, states that any instance $\vec{u} \sim \vec{v}$ of CCA_2 can be automatically extended into an instance $\vec{u}' \sim \vec{v}'$ of CCA_2^a of, at most, polynomial size.

Proposition 5.3. *For every instance $\vec{u} \sim \vec{v}$ of CCA_2, there exists $\vec{u}_1, \vec{v}_1$ such that $\vec{u}, \vec{u}_1 \sim \vec{v}, \vec{v}_1$ is an instance of CCA_2^a (modulo Perm) and $|\vec{u}_1|+|\vec{v}_1|$ is of polynomial size in $|\vec{u}|+|\vec{v}|$. We let $\mathsf{completion}(\vec{u} \sim \vec{v})$ be the formula $\vec{u}, \vec{u}_1 \sim \vec{v}, \vec{v}_1$.*

Proof. We first show how to extend an instance of CCA_2 into an instance of CCA_2^a. Let $(u_i)_{i\in I} \sim (v_i)_{i\in I}$ be an instance of CCA_2^a. Let $I' \subseteq I$, we want to extend $(u_i)_{i\in I'} \sim (v_i)_{i\in I'}$ into an instance of CCA_2^a. Let $\phi \equiv (u_i)_{i\in I}$, $\psi \equiv (v_i)_{i\in I}$, since $(u_i)_{i\in I} \sim (v_i)_{i\in I}$ is an instance of CCA_2^a we have:

$$(\phi, \mathcal{X}_{\mathsf{enc}}, \mathcal{X}_{\mathsf{dec}}, \sigma_{\mathsf{rand}}, \theta_{\mathsf{enc}}, \lambda_{\mathsf{dec}}) R^{\mathcal{K}}_{\text{CCA}_2^a} (\psi, \mathcal{X}_{\mathsf{enc}}, \mathcal{X}_{\mathsf{dec}}, \sigma'_{\mathsf{rand}}, \theta'_{\mathsf{enc}}, \lambda'_{\mathsf{dec}})$$

For all $x \in \mathcal{X}_{\mathsf{enc}} \cup \mathcal{X}_{\mathsf{dec}}$, we let $i_x \in I$ be the index corresponding to $x\theta_{\mathsf{enc}}\lambda_{\mathsf{dec}} \sim x\theta'_{\mathsf{enc}}\lambda'_{\mathsf{dec}}$. Moreover, for all $x \in \mathcal{X}_{\mathsf{dec}}$, we let t_{i_x} be the context used for the decryption in the definition of $R^{\mathcal{K}}_{\text{CCA}_2^a}$ (hence we have $x\lambda_{\mathsf{dec}} \equiv \mathsf{else}^*(l, \mathsf{dec}(t_{i_x}\theta_{\mathsf{enc}}\lambda_{\mathsf{dec}}), \mathsf{sk}))$.

Outline We are going to define $I^{lr}, I^l, I^r \subseteq I$ and $(\tilde{u}_i)_{i\in J}$, $(\tilde{v}_i)_{i\in J}$ (where $J = I^{lr} \cup I^l \cup I^r$) such that:

- I^{lr}, I^l, I^r are pair-wise disjoints and $I' \subseteq I^{lr}$.
- $(\tilde{u}_i)_{i\in J} \sim (\tilde{v}_i)_{i\in J}$ is an instance of CCA_2^a of polynomial size with respect to $\sum_{i\in I'} |u_i| + |v_i|$.

Intuitively, I^{lr} is the subset of indices of $I\backslash I'$ of the terms that are subterm of $(u_i)_{i\in I'} \sim (v_i)_{i\in I'}$ *on the left and on the right*, i.e. for all $i \in I^{lr}$, $u_i \in \mathsf{st}((u_i)_{i\in I'})$ *and* $v_i \in \mathsf{st}((v_i)_{i\in I'})$. The terms whose index is in I^{lr} are easy to handle, as they are immediately bounded by the terms whose indices is in I'.

Then, I^l is the subset of indices of $I\backslash I'$ of the terms that are subterms of $(u_i)_{i\in I'} \sim (v_i)_{i\in I'}$ *on the left only* (i.e. for every $i \in I^l$, we only know that $u_i \in \mathsf{st}((u_i)_{i\in I'})$). Terms with indices in I^l are easy to bound on the left, but not on the right. To bound the right terms, we introduce dummy messages (by replace encryptions by encryption of $g()$, where g is an adversarial function symbol in $\mathcal{G}$). Similarly I^r is the subset of indices of $I\backslash I'$ of the terms that are subterms of $(u_i)_{i\in I'} \sim (v_i)_{i\in I'}$ *on the right only.*

First, we define I^{lr}, I^l, I^r, and then we define the corresponding CCA_2^a instance $(\tilde{u}_i)_{i\in J} \sim (\tilde{v}_i)_{i\in J}$.

Inductive Definition of the Left and Right Appearance Sets We define by induction on $i \in I'$ the sets $I^l_i, I^r_i \subseteq I$. Intuitively, I^l_i is the set of indices of I needed so that u_i is well-defined (same for I^r_i and v_i). Let $i \in I'$, we do a case disjunction on the rule applied to u_i, v_i in $R^{\mathcal{K}}_{\text{CCA}_2^a}$:

- **No Call to the Oracles:** In that case we take $I^l_i = I^r_i = \{i\}$.
- **Encryption Case:** let $t, t' \in \mathcal{T}(\mathcal{F}_{\backslash \mathbf{0}}, \mathcal{N}, \mathcal{X}_{\mathsf{dec}})$ such that $u_i \equiv \{t\lambda_{\mathsf{dec}}\}^{-}_{-}$ and $v_i \equiv \{t'\lambda'_{\mathsf{dec}}\}^{-}_{-}$. To have u_i well-defined, we need all the decryptions in u_i to be well-defined (same for v_i). Hence let:

$$I^l_i = \{i\} \cup \bigcup_{x\in\mathcal{X}_{\mathsf{dec}}\cap\mathsf{st}(t)} I^l_{i_x} \qquad\qquad I^r_i = \{i\} \cup \bigcup_{x\in\mathcal{X}_{\mathsf{dec}}\cap\mathsf{st}(t')} I^r_{i_x}$$

- **Decryption Case:** recall that $u_i \equiv \mathsf{else}^*(l, \mathsf{dec}(u, \mathsf{sk}))$ where $u \equiv t_i\theta_{\mathsf{enc}}\lambda_{\mathsf{dec}}$. Therefore we need all encryption in $\mathcal{X}_{\mathsf{enc}} \cap \mathsf{st}(t_i)$ and decryption in $\mathcal{X}_{\mathsf{dec}} \cap \mathsf{st}(t_i)$ to be defined, on the left and on the right. Hence we let:

$$I^l_i = \{i\} \cup \bigcup_{x\in(\mathcal{X}_{\mathsf{dec}}\cup\mathcal{X}_{\mathsf{enc}})\cap\mathsf{st}(t_i)} I^l_{i_x} \qquad\qquad I^r_i = \{i\} \cup \bigcup_{x\in(\mathcal{X}_{\mathsf{dec}}\cup\mathcal{X}_{\mathsf{enc}})\cap\mathsf{st}(t_i)} I^r_{i_x}$$

We let:

$$I^{lr} = \bigcup_{i\in I'} I_i^l \cap \bigcup_{i\in I'} I_i^r \qquad I^{l} = \bigcup_{i\in I'} I_i^l \cap \overline{\bigcup_{i\in I'} I_i^r} \qquad I^{r} = \overline{\bigcup_{i\in I'} I_i^l} \cap \bigcup_{i\in I'} I_i^r$$

These three sets are disjoint and form a partition of $\bigcup_{i\in I'} I_i^l \cup I_i^r$. Remark that for every $i \in I_j^l$, u_i is a subterm of u_j. Hence, for every $i \in I^{lr} \cup I^l$, there exists $j \in I'$ such that u_i is a subterm of u_j.

Building the New Instance We define (by induction on i) the terms $(\tilde{u}_i)_{i\in J}$, by letting $\tilde{u}_i$ be:

- u_i when $i \in I^{lr} \cup I^l$.
- $\{g()\}^{\mathsf{n}}_{\mathsf{pk}}$ when $i \in I^r$ and u_i is an encryption, with $u_i \equiv \{\_\}^{\mathsf{n}}_{\mathsf{pk}}$.
- $\mathsf{else}^*(\tilde{l}, \mathsf{dec}(\tilde{u}, \mathsf{sk}))$ when $i \in I^r$ and u_i is a decryption, where $u_i \equiv \mathsf{else}^*(l, \mathsf{dec}(u, \mathsf{sk}))$, $u \equiv t_i\theta_{\mathsf{enc}}\lambda_{\mathsf{dec}}$, l is the sequence of guards $l \equiv (\mathsf{eq}(u, y_1), \ldots, \mathsf{eq}(u, y_m))$ where $(y_1, \ldots, y_m) = \mathsf{sort}(\mathcal{Y}_u\theta_{\mathsf{enc}})$. Then we take:
  - $\tilde{u} \equiv t_i\tilde{\theta}_{\mathsf{enc}}\tilde{\lambda}_{\mathsf{dec}}$, where $\tilde{\theta}_{\mathsf{enc}} = \{x \mapsto \tilde{u}_{i_x} \mid x \in \mathcal{X}_{\mathsf{enc}}\}$ and $\tilde{\lambda}_{\mathsf{dec}} = \{x \mapsto \tilde{u}_{i_x} \mid x \in \mathcal{X}_{\mathsf{dec}}\}$.
  - $\tilde{l} \equiv (\mathsf{eq}(\tilde{u}, \tilde{y}_1), \ldots, \mathsf{eq}(\tilde{u}, \tilde{y}_m))$ where $(\tilde{y}_1, \ldots, \tilde{y}_m) = \mathsf{sort}(\mathcal{Y}_u\tilde{\theta}_{\mathsf{enc}})$.

Similarly, we define $\tilde{v}_i$ for every $i \in J$.

Conclusion Let $J = I^{lr} \cup I^l \cup I^r$. To conclude, we check that $(\tilde{u}_i)_{i\in J} \sim (\tilde{v}_i)_{i\in J}$:

- is a CCA_2^a instance. This is done by induction on $i \in J$.
- is of polynomial size w.r.t. $(u_i)_{i\in I'} \sim (v_i)_{i\in I'}$.

We omit the details of the proof of the first point.

For the second point, we first show by induction on i that $|I_i^l| \leq |u_i|$ and $|I_i^r| \leq |v_i|$. We deduce that:

$$|J| = \Big|\bigcup_{i\in I'} I_i^r \cup I_i^l\Big| \leq \sum_{i\in I'} |I_i^r| + |I_i^l| \leq \sum_{i\in I'} |u_i| + |v_i|$$

Let $i \in I^{lr} \cup I^l$, we know that there exists $j \in I'$ such that u_i is a subterm of u_j. Since $\tilde{u}_i \equiv u_i$, we deduce that $|\tilde{u}_i| \leq |u_j| \leq \sum_{j\in I'} |u_j| + |v_j|$.

Let $i \in I^r$. If $\tilde{u}_i$ is an encryption then it is of constant size. Assume $\tilde{u}_i$ is a decryption. Then $\tilde{u}_i$ is the decryption v_i where any encryption whose index is in I^{lr} has been replaced by its left counterpart, and any encryption whose index is in I^r has been replaced by a dummy encryption (the case I^l cannot happen, since $i \in I^r$). Since there are at most $|v_i| - 1$ such encryptions (as v_i contain at least one occurrence of the dec function symbol), and since any encryption with index in I^{lr} or I^r is upper-bounded by $\sum_{j\in I'} |u_j| + |v_j|$, we get that:

$$|\tilde{u}_i| \leq |v_i| + (|v_i| - 1).\sum_{j\in I'} |u_j| + |v_j| \leq |v_i|.\sum_{j\in I'} |u_j| + |v_j| \leq \Big(\sum_{j\in I'} |u_j| + |v_j|\Big)^2$$

We deduce that $(\tilde{u}_i)_{i\in J} \sim (\tilde{v}_i)_{i\in J}$ is of polynomial size in $\sum_{j\in I'} |u_j| + |v_j|$. ■

5.4.2 Length in the CCA_2 Axioms

If we want the formula $\{t\}^r_{\mathsf{pk}} \sim \{t'\}^{r'}_{\mathsf{pk}'}$ to be a valid application of the CCA_2 axioms, we need to make sure that t and t' are of the same length. Since the length of terms depend on implementation details (e.g. how is the pair $\langle \_ , \_ \rangle$ implemented), we let the user supply implementation assumptions. We use a predicate symbol $\mathsf{EQL}(\_, \_)$ in the logic, together with some derivation rules $\mathcal{D}_{\mathsf{L}}$ (supplied by the user), and we require that they verify the following properties:

- **Complexity:** for every u, v, we can decide whether $\mathsf{EQL}(u, v)$ is a consequence of $\mathcal{D}_{\mathsf{L}}$ in polynomial time in $|u| + |v|$.
- **Branch Invariance:** for all terms b, u, v, t, if $\mathsf{EQL}(\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v,\ t)$ is derivable using $\mathcal{D}_{\mathsf{L}}$ then $\mathsf{EQL}(u,\ t)$ and $\mathsf{EQL}(v,\ t)$ are derivable using $\mathcal{D}_{\mathsf{L}}$.

$$\mathsf{Length}(\mathsf{n}) = l_\eta \qquad \mathsf{Length}(0_{l_e}) = l_e$$

$$\mathsf{Length}(u) = \mathsf{Length}(u') \text{ if } u =_R u' \text{ and } \mathsf{Length}(u), \mathsf{Length}(u') \text{ are not } \mathsf{undefined}$$

$$\mathsf{Length}(\langle u\,,\, v\rangle) = \mathsf{Length}(u) + \mathsf{Length}(v) + l_{\langle\,,\rangle} \qquad \forall l_e.\mathsf{Length}(\mathsf{pad}_{l_e}(u)) = l_e$$

$$\forall k.\mathsf{Length}(\{u\}^{\mathsf{n}}_{\mathsf{pk}}) = k.\mathsf{l}_{\{\mathsf{block}\}} + l_{\{\}} \text{ if } \mathsf{Length}(u) = k.\mathsf{l}_{\mathsf{block}}$$

$$\forall k.\mathsf{Length}(\mathsf{dec}(u, \mathsf{sk})) = k.\mathsf{l}_{\mathsf{block}} \text{ if } \mathsf{Length}(u) = k.\mathsf{l}_{\{\mathsf{block}\}} + l_{\{\}}$$

$$\mathsf{Length}(\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v) = \begin{cases} \mathsf{Length}(u) & \text{if } \mathsf{Length}(u) = \mathsf{Length}(v) \\ \mathsf{undefined} & \text{otherwise} \end{cases}$$

Figure 5.5: Definition of the Length partial function.

We add to all CCA_2 instances the side condition $\mathsf{EQL}(m_l, m_r)$ for every encryption oracle call on (m_l, m_r). Then, we know that our CCA_2 instances are valid in any computational model $\mathcal{M}_\mathsf{c}$ where the encryption is interpreted as a IND-CCA_2 encryption scheme, and where the following property holds: for every ground terms u, v, if $\mathsf{EQL}(u, v)$ is derivable using $\mathcal{D}_\mathsf{L}$, then:

$$[\![\mathsf{length}(u)]\!]_{\mathcal{M}_\mathsf{c}} = [\![\mathsf{length}(v)]\!]_{\mathcal{M}_\mathsf{c}}$$

Example: Block Cipher We give here an example of derivation rules $\mathcal{D}_\mathsf{L}$ that axiomatize the fact that the encryption function is built upon a block cipher, taking blocks of length $\mathsf{l}_{\mathsf{block}}$ and returning blocks of length $\mathsf{l}_{\{\mathsf{block}\}}$. The length constant $l_{\{\}}$ is used to represent the constant length used, e.g., for the IV and the HMAC.

We let $\mathcal{L}$ be a set of length constants, and we define a length expression to be an expression of the form $\sum_{l\in L} k_l.l$, where L is a finite subset of $\mathcal{L}$ and $(k_l)_{l\in L}$ are positive integers. We consider length expressions modulo commutativity (i.e. $3.l_1 + 4.l_2 \approx 4.l_2 + 3.l_1$), and we assume that for every length expression l_e, there exists a function symbol $\mathsf{pad}_{l_e} \in \mathcal{F}$. Intuitively pad_{l_e} is function padding messages to length l: if the message is too long it truncates it, and if the message is too short it pads it. Similarly, we assume that for every l_e, we have a function symbol $0_{l_e} \in \mathcal{F}$ or arity zero which, intuitively, returns l_e zeroes. Also, we assume that $\mathcal{L}$ contains the following length constants: $l_{\langle\,,\rangle}, l_{enc}, \mathsf{l}_{\mathsf{block}}, l_\eta$.

We define the Length (partial) function on terms in Figure 5.5. Then, we let $\mathcal{D}_\mathsf{L}$ be the (recursive) set of unitary axioms:

$$\frac{\mathsf{Length}(u) = \mathsf{Length}(v) \neq \mathsf{undefined}}{\mathsf{EQL}(u, v)}$$

Proposition 5.4. *The function Length is well defined, and the set of axioms $\mathcal{D}_\mathsf{L}$ satisfies the soundness and branch invariance properties.*

Proof. To check that Length is well defined, one just need to look at the critical pairs in the definition and check that they are joinable. Soundness is easy, as $[\![\mathsf{Length}]\!]_{\mathcal{M}_\mathsf{c}}$ is just an under-approximation of $[\![\mathsf{length}]\!]_{\mathcal{M}_\mathsf{c}}$ in every computational model $\mathcal{M}_\mathsf{c}$ where the encryption is interpreted as a block cipher, the padding functions are interpreted as expected etc.

Finally, branch invariance follows directly from the definition of $\mathsf{Length}(\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v)$. ■

Remark 5.5. We can allow the user to add any set of length equations, as long as the branch invariance property holds and the Length function is well-defined. E.g. one may wish to add equations like $\mathsf{Length}(A) = \mathsf{Length}(B) = \mathsf{Length}(C) = l_{\mathsf{agent}}$. □

5.5 Main Result and Difficulties

We let Ax be the conjunction of Struct-Ax and CCA_2. We now state our main result.

Theorem (Main Result)**.** *The following problem is decidable:*
Input: *A ground formula $\vec{u} \sim \vec{v}$.*
Question: *Is $\mathsf{Ax} \wedge \vec{u} \not\sim \vec{v}$ unsatisfiable?*

We give here an overview of the problems that have to be overcome in order to obtain the decidability result. Before starting, a few comments. We close all rules under permutations. The Sym rule commutes with all the other rules, and the CCA_2 unitary axioms are closed under Sym. Therefore we can remove Perm and Sym from the set of rules. Observe that $\mathsf{CS}^{\mathsf{no}}_{\mathsf{if}}, \mathsf{FA}_{\setminus \mathbf{0}}, \mathsf{Dup}$ and CCA_2 are all *decreasing rules*, i.e. the premises are smaller than the conclusion. The only non-decreasing rules are R, which may rewrite a term into a larger one, and Restr, which we eliminate later. Therefore, to obtain a complete and terminating strategy for Ax, we need to bound the size of the terms introduced when applying the R rule. The main result of this chapter is a characterization of unnecessary rewritings, which will yield a bound on the size of the premises of a useful R application. We will deduce an upper-bound on the minimal proof of a formula, if it exists.

First, we define a way of describing fragments of our logic:

Definition 5.12. For every formula ϕ, we write $P \vdash \phi$ if P is a proof of ϕ.

Definition 5.13. Let Σ be the set of axiom names, seen as an alphabet. For all $\mathcal{L} \subseteq \Sigma^*$, we let $\mathfrak{F}(\mathcal{L})$ be the fragment of our logic defined by: a formula ϕ is in the fragment iff there exists a proof P such that $P \vdash \phi$ and, for every branch ρ of P, the word w obtained by collecting the axiom names along ρ (starting from the root) is in $\mathcal{L}$.

Example 5.5. The derivation in Example 5.1, page 176 is the fragments:

$$\mathfrak{F}\left(R \cdot \mathsf{CS}^{\mathsf{no}}_{\mathsf{if}} \cdot \mathsf{Refl}\right) \qquad \square$$

Necessary Introductions As we saw in Example 5.1, it might be necessary to use R in the "wrong direction", typically to introduce new conditionals. A priori, this yields an unbounded search space. Therefore our goal is to characterize in which situations we need to use R in the "wrong direction", and with which instances. We identify two necessary reasons for introducing new conditionals.

- First, to match the shape of the term on the other side, like $g()$ in Example 5.1:

$$\frac{\mathsf{if}\ g()\ \mathsf{then}\ \mathsf{n}_0\ \mathsf{else}\ \mathsf{n}_1 \sim \mathsf{if}\ g()\ \mathsf{then}\ \mathsf{n}\ \mathsf{else}\ \mathsf{n}}{\mathsf{if}\ g()\ \mathsf{then}\ \mathsf{n}_0\ \mathsf{else}\ \mathsf{n}_1 \sim \mathsf{n}}\ R$$

  In this case, the introduced conditional is exactly the conditional that appeared on the other side of $\sim$. With more complex examples this may not be the case. Nonetheless, an introduced conditional is always bounded by the conditional it matches.

- Second, we might introduce a guard in order to fit to the definition of safe decryptions in the CCA_2 axioms, as in (5.4) in Example 5.2. Here also, the introduced guard will be of bounded size. Indeed, guards of $\mathsf{dec}(s, \mathsf{sk})$ are of the form $\mathsf{eq}(s, \alpha)$ where α is a subterm of s. Therefore, for a fixed s, there are a bounded number of them, and they are of bounded size.

These two (informally defined) conditions are actually sufficient: any other rewriting is a unnecessary detour. We illustrate this on an example:

Example 5.6 (Cut Elimination)**.** We consider a proof of $s \sim t$ where the $\mathsf{CS}^{\mathsf{no}}_{\mathsf{if}}$ rule is applied on two conditionals that have just been introduced by the R rule:

$$\frac{\dfrac{a, s \sim b, t \qquad a, s \sim b, t}{\mathsf{if}\ a\ \mathsf{then}\ s\ \mathsf{else}\ s \sim \mathsf{if}\ b\ \mathsf{then}\ t\ \mathsf{else}\ t}\ \mathsf{CS}^{\mathsf{no}}_{\mathsf{if}}}{s \sim t}\ R$$

Here, the conditional a and b can be of arbitrary size. Intuitively, this is not a problem since any proof of $a, s \sim b, t$ includes a proof of $s \sim t$. $\square$

The idea is that we can extract a proof of $s \sim t$ from any proof of $a, s \sim b, t$. We prove this by showing that Restr applications can be eliminated.

Lemma 5.1 (Restr Elimination)**.** *If $P \vdash \vec{u} \sim \vec{v}$ with P in the fragment:*

$$\mathfrak{F}\left((\mathit{CS}^{\mathit{no}}_{\mathit{if}} + R + \mathit{FA}_{\setminus \mathbf{0}} + \mathit{Dup} + \mathrm{CCA}_2 + \mathit{Restr})^*\right)$$

then there exists P' such that $P' \vdash \vec{u} \sim \vec{v}$ and P' contains no Restr applications. Moreover:

- *the height of P' is no larger than the height of P.*
- *if P is in a fragment $\mathfrak{F}(\mathcal{L})$ where $\mathcal{L}$ is closed by sub-words then P' is in $\mathfrak{F}(\mathcal{L})$.*

Proof. We do a proof by induction on the height of the derivation P of $\vec{u} \sim \vec{v}$. More precisely, we prove that for any height n and formula $\vec{u} \sim \vec{v}$, for any derivation P of $\vec{u} \sim \vec{v}$ in the fragment:

$$\mathfrak{F}\left((\mathsf{CS}^{\mathsf{no}}_{\mathsf{if}} + R + \mathsf{FA}_{\setminus \mathbf{0}} + \mathsf{Dup} + \mathrm{CCA}_2 + \mathsf{Restr})^*\right)$$

such that P is of height n, there exists a derivation P' with no Restr of $\vec{u} \sim \vec{v}$ of height no larger than n.

Assume that we have a derivation P of $\vec{u} \sim \vec{v}$ where the last rule applied is Restr:

$$\dfrac{\vec{u}, \vec{t} \sim \vec{v}, \vec{s}}{\vec{u} \sim \vec{v}}\ \mathsf{Restr}$$

We discriminate on the second last rule applied:

- If it is a unitary axiom we conclude easily using the fact that unitary axioms are closed under Restr.
- If it is a $\mathsf{FA}_{\setminus \mathbf{0}}$ axiom and $\vec{t}$ is not involved in this function application then P is of the form:

$$\dfrac{\dfrac{\begin{array}{c}\vdots\ (A)\\ \vec{u}, \vec{u}', \vec{t} \sim \vec{v}, \vec{v}', \vec{t}'\end{array}}{f(\vec{u}), \vec{u}', \vec{t} \sim f(\vec{v}), \vec{v}', \vec{t}'}\ \mathsf{FA}_{\setminus \mathbf{0}}}{f(\vec{u}), \vec{u}' \sim f(\vec{v}), \vec{v}'}\ \mathsf{Restr}$$

  To conclude, we apply the induction hypothesis to extract a proof of $\vec{u}, \vec{u}' \sim \vec{v}, \vec{v}'$ in the wanted fragment from (A). We conclude by applying the $\mathsf{FA}_{\setminus \mathbf{0}}$ rule:

$$\dfrac{\begin{array}{c}\vdots\ (A)\\ \vec{u}, \vec{u}', \vec{t} \sim \vec{v}, \vec{v}', \vec{t}'\end{array}}{\vec{u}, \vec{u}' \sim \vec{v}, \vec{v}'}\ \mathsf{Restr} \quad \overset{\text{ind. hyp.}}{\Longrightarrow} \quad \begin{array}{c}\vdots\ (A')\\ \vec{u}, \vec{u}' \sim \vec{v}, \vec{v}'\end{array} \quad \overset{\text{apply } \mathsf{FA}_{\setminus \mathbf{0}}}{\Longrightarrow} \quad \dfrac{\begin{array}{c}\vdots\ (A')\\ \vec{u}, \vec{u}' \sim \vec{v}, \vec{v}'\end{array}}{f(\vec{u}), \vec{u}' \sim f(\vec{v}), \vec{v}'}\ \mathsf{FA}_{\setminus \mathbf{0}}$$

- If it is a $\mathsf{FA}_{\setminus \mathbf{0}}$ axiom and $\vec{t}$ is involved in this function application then P is of the form:

$$\dfrac{\dfrac{\begin{array}{c}\vdots\ (A)\\ \vec{u}, \vec{u}', \vec{u}'' \sim \vec{v}, \vec{v}', \vec{v}''\end{array}}{\vec{u}, \vec{u}', f(\vec{u}'') \sim \vec{v}, \vec{v}', f(\vec{v}'')}\ \mathsf{FA}_{\setminus \mathbf{0}}}{\vec{u} \sim \vec{v}}\ \mathsf{Restr}$$

  By applying the induction hypothesis, we extract a proof of $\vec{u} \sim \vec{v}$ in the wanted fragment directly:

$$\begin{array}{c}\vdots\ (A)\\ \vec{u}, \vec{u}', \vec{u}'' \sim \vec{v}, \vec{v}', \vec{v}''\end{array} \quad \overset{\text{ind. hyp.}}{\Longrightarrow} \quad \begin{array}{c}\vdots\ (A')\\ \vec{u} \sim \vec{v}\end{array}$$

- If it is $\mathsf{CS}^{\mathsf{no}}_{\mathsf{if}}$:

$$\dfrac{\dfrac{\begin{array}{c}\vdots\ (A)\\ \vec{w}_0, \vec{w}_1, b, (u_i)_{i \in I \cup J} \sim \vec{w}'_0, \vec{w}'_1, b', (u'_i)_{i \in I \cup J}\end{array} \qquad \begin{array}{c}\vdots\ (B)\\ \vec{w}_0, \vec{w}_1, b, (v_i)_{i \in I \cup J} \sim \vec{w}'_0, \vec{w}'_1, b', (v'_i)_{i \in I \cup J}\end{array}}{\vec{w}_0, \vec{w}_1, (\mathsf{if}\ b\ \mathsf{then}\ u_i\ \mathsf{else}\ v_i)_{i \in I \cup J} \sim \vec{w}'_0, \vec{w}'_1, (\mathsf{if}\ b'\ \mathsf{then}\ u'_i\ \mathsf{else}\ v'_i)_{i \in I \cup J}}\ \mathsf{CS}^{\mathsf{no}}_{\mathsf{if}}}{\vec{w}_0, (\mathsf{if}\ b\ \mathsf{then}\ u_i\ \mathsf{else}\ v_i)_{i \in I} \sim \vec{w}'_0, (\mathsf{if}\ b'\ \mathsf{then}\ u'_i\ \mathsf{else}\ v'_i)_{i \in I}}\ \mathsf{Restr}$$

We apply the induction hypothesis twice:

$$\dfrac{\begin{array}{c}\vdots\ (A)\\ \vec{w}_0, \vec{w}_1, b, (u_i)_{i\in I\cup J} \sim \vec{w}'_0, \vec{w}'_1, b', (u'_i)_{i\in I\cup J}\end{array}}{\vec{w}_0, b, (u_i)_{i\in I} \sim \vec{w}'_0, b', (u'_i)_{i\in I}}\ \mathsf{Restr} \qquad \overset{\text{ind. hyp.}}{\Longrightarrow} \qquad \begin{array}{c}\vdots\ (A')\\ \vec{w}_0, b, (u_i)_{i\in I} \sim \vec{w}'_0, b', (u'_i)_{i\in I}\end{array}$$

$$\dfrac{\begin{array}{c}\vdots\ (B)\\ \vec{w}_0, \vec{w}_1, b, (v_i)_{i\in I\cup J} \sim \vec{w}'_0, \vec{w}'_1, b', (v'_i)_{i\in I\cup J}\end{array}}{\vec{w}_0, b, (v_i)_{i\in I} \sim \vec{w}'_0, b', (v'_i)_{i\in I}}\ \mathsf{Restr} \qquad \overset{\text{ind. hyp.}}{\Longrightarrow} \qquad \begin{array}{c}\vdots\ (B')\\ \vec{w}_0, b, (v_i)_{i\in I} \sim \vec{w}'_0, b', (v'_i)_{i\in I}\end{array}$$

We obtain the derivation:

$$\dfrac{\begin{array}{c}\vdots\ (A')\\ \vec{w}_0, b, (u_i)_{i\in I} \sim \vec{w}'_0, b', (u'_i)_{i\in I}\end{array} \qquad \begin{array}{c}\vdots\ (B')\\ \vec{w}_0, b, (v_i)_{i\in I} \sim \vec{w}'_0, b', (v'_i)_{i\in I}\end{array}}{\vec{w}_0, (\mathsf{if}\ b\ \mathsf{then}\ u_i\ \mathsf{else}\ v_i)_{i\in I} \sim \vec{w}'_0, (\mathsf{if}\ b'\ \mathsf{then}\ u'_i\ \mathsf{else}\ v'_i)_{i\in I}}\ \mathsf{CS}^{\mathsf{no}}_{\mathsf{if}}$$

- The Dup and R axioms are trivial to handle. ■

Remark 5.6. In the proof, we need the CCA_2 axioms to be closed under Restr. Note that this created some problems, which we dealt with earlier, in Section 5.4.1. □

Using this lemma, we can deal with Example 5.6 by doing a proof cut elimination. More generally, by induction on the proof size, we can guarantee that no such proof cuts appear. This is the strategy we are going to follow: look for proof cuts that introduce unbounded new terms, eliminate them, and show that after sufficiently many cut eliminations all the subterms appearing in the proof are bounded by the (R-normal form of the) conclusion. But a proof may contain more complex behaviors than just the introduction of a conditional followed by a $\mathsf{CS}^{\mathsf{no}}_{\mathsf{if}}$ application. For example the conditional being matched could have been itself introduced earlier to match another conditional, which itself was introduced to match a third conditional etc.

Example 5.7. We illustrate this on an example. When it is more convenient, we write terms containing only $\mathsf{if\_then\_else\_}$ and other subterms (handled as constants) as binary trees; we also index some subterms with a number, which helps keeping track of them across rule applications. Consider the derivation:

$$\dfrac{\begin{array}{c}\vdots\ (A)\\ a_1, b_2, b_3, u_4, w_5, u_6, v_7 \sim d_1, c_2, d_3, s_4, t_5, r_6, p_7\end{array}}{\begin{array}{ccc} a_1 & & d_1 \\ \swarrow\ \searrow & & \swarrow\ \searrow \\ b_2 \quad v_7 & & c_2 \quad p_7 \\ \swarrow\ \searrow & \sim & \swarrow\ \searrow \\ u_4 \quad b_3 & & s_4 \quad d_3 \\ \swarrow\ \searrow & & \swarrow\ \searrow \\ w_5 \quad u_6 & & t_5 \quad r_6 \end{array}}\ \mathsf{FA}^{(3)}_{\setminus\mathbf{0}}$$

$$\dfrac{}{\mathsf{if}\ a\ \mathsf{then}\ u\ \mathsf{else}\ v \sim \mathsf{if}\ c\ \mathsf{then}\ s\ \mathsf{else}\ t}\ R$$

where $p \equiv \mathsf{if}\ c\ \mathsf{then}\ s\ \mathsf{else}\ t$. Here the conditionals b, d and the terms w, r are, a priori, arbitrary. Therefore we would like to bound them or remove them through a cut elimination. The cut elimination technique used in Example 5.6 does not apply here because we cannot extract a proof of $a \sim c$.

But we can extract a proof of $b_2, b_3 \sim c_2, d_3$. Using the axioms soundness, this means that in every appropriate computational model, $[\![b, b]\!] \approx [\![c, d]\!]$. Therefore, no adversary can distinguish between getting twice the same value sampled from $[\![b]\!]$ and getting a pair of values sampled from $[\![c, d]\!]$. In particular, $[\![c]\!]_{\eta,\rho} = [\![d]\!]_{\eta,\rho}$, except for a negligible number of random tapes ρ. □

A First Key Lemma A natural question is to ask whether the semantic equality $[\![c]\!] = [\![d]\!]$ implies a syntactic equality. While this is not the case in general, there are fragments of our logic in which this holds. To define such a fragment, we annotate the rules $\mathsf{FA}_{\setminus\mathbf{0}}$ by the function symbol involved, and we let $\mathsf{FA}_{\mathsf{s}} = \{\mathsf{FA}_f \mid f \in \mathcal{F}_{\setminus \mathsf{if},\mathbf{0}}\}$ be the restriction of $\mathsf{FA}_{\setminus\mathbf{0}}$ to function symbols different from $\mathsf{if\_then\_else\_}$. Formulas that can be proven in the fragment $\mathfrak{F}(\mathsf{FA}_{\mathsf{s}}{}^* \cdot \mathsf{Dup}^* \cdot \mathrm{CCA}_2)$ have a particular shape, which is completely characterized by the rules applied in the derivation:

Proposition 5.5. *For all $b, b' \in \mathcal{T}(\mathcal{F}, \mathcal{N})$, if $b \sim b'$ is in the fragment $\mathfrak{F}(\mathit{FA_s}^* \cdot \mathit{Dup}^* \cdot \text{CCA}_2)$ then $b \equiv C[\vec{w}, (\alpha_i)_i, (\mathit{dec}_j)_j]$, $b' \equiv C[\vec{w}, (\alpha'_i)_i, (\mathit{dec}'_j)_j]$ and the CCA_2 instance applied is (up-to α-renaming):*

$$\vec{w}, (\alpha_i)_i, (\mathit{dec}_j)_j \sim \vec{w}, (\alpha'_i)_i, (\mathit{dec}'_j)_j$$

where $(\alpha_i, \alpha'_i)_i$ are the encryption oracle calls and $(\mathit{dec}_j, \mathit{dec}'_j)_j$ are the decryption oracle calls.

Proof. This is easy immediate by induction on the proof derivation. ■

Using this characterization, we proof the following key lemma:

Lemma 5.2. *For all b, b', b'', if $b, b \sim b', b''$ is in the fragment $\mathfrak{F}(\mathit{FA_s}^* \cdot \mathit{Dup}^* \cdot \text{CCA}_2)$ then $b' \equiv b''$.*

Proof. From Proposition 5.5 we have:

$$b \equiv C^l[\vec{w}^l, (\alpha_i^l)_{i \in I^l}, (\mathsf{dec}_j^l)_{j \in J^l}] \qquad b' \equiv C^l[\vec{w}^l, (\alpha_i'^l)_{i \in I^l}, (\mathsf{dec}_j'^l)_{j \in J^l}]$$
$$b \equiv C^r[\vec{w}^r, (\alpha_i^r)_{i \in I^r}, (\mathsf{dec}_j^r)_{j \in J^r}] \qquad b'' \equiv C^r[\vec{w}^r, (\alpha_i'^r)_{i \in I^r}, (\mathsf{dec}_j'^r)_{j \in J^r}]$$

Assume that $C^l \not\equiv C^r$. Let p be the position of a hole of C^l such that p is a valid position but not a hole position in C^r (if this is not the case, invert b' and b''). Then we have three cases:

- The hole at $b_{|p}$ is mapped to a term $u \in \vec{w}^l$. Then, we can rewrite the proof such that p is a hole position in both terms.
- The hole at $b_{|p}$ is mapped to an encryption oracle call $\{m\}^{\mathsf{n_e}}_{\mathsf{pk(n)}}$ in b and $\{m'\}^{\mathsf{n_e}}_{\mathsf{pk(n)}}$ in b'. Since $\{m\}^{\mathsf{n_e}}_{\mathsf{pk(n)}}$ is an encryption in the CCA_2 application, we know from the freshness side-condition that $\mathsf{n_e}$ does not appear in $\vec{w}^r$. But since $C^r_{|p}$ is not a hole, the proof of $b, b \sim b', b''$ includes the sub-proof:

$$\dfrac{\dfrac{}{\ldots, \mathsf{n_e} \sim \ldots, \mathsf{n'_e}}\ \text{CCA}_2}{\vdots}$$
$$\dfrac{\ldots, m, \mathsf{pk(n)}, \mathsf{n_e} \sim \ldots, m', \mathsf{pk(n)}, \mathsf{n_e}}{\ldots, \{m\}^{\mathsf{n_e}}_{\mathsf{pk(n)}} \sim \ldots, \{m'\}^{\mathsf{n_e}}_{\mathsf{pk(n)}}}\ \mathsf{FA}_{\setminus \mathbf{0}}$$

  Since $\mathsf{n_e}$ is a name in $\mathcal{N}$ and cannot be modified by any rules in $\{R, \mathsf{FA_s}, \mathsf{Dup}\}$. Therefore $\mathsf{n_e} \in \vec{w}^r$. This contradict the freshness side-condition. Absurd.
- If the hole at $b_{|p}$ is mapped to a decryption oracle call $\mathsf{dec}^l_{i_0}$ in b. Since $C^r_{|p}$ is not a hole, and since function applications on $\mathsf{FA_s}$ cannot be applied on the $\mathsf{if\_then\_else\_}$ function symbols we know that there exists m, m' such that $\mathsf{dec}^l_{i_0} \equiv \mathsf{dec}(m, sk(\mathsf{n}))$ and $\mathsf{dec}'^l_{i_0} \equiv \mathsf{dec}(m', sk(\mathsf{n}))$. Moreover, since $\mathsf{dec}^l_{i_0}$ is a decryption in the CCA_2 application, we know from the key-usability side-condition that $\mathsf{sk(n)}$ appears only in decryption position in $\vec{w}^r$. Then the reasoning we have in the previous cases applies here. Indeed, we know that $C^r_{|p}$ is not a hole, hence the proof of $b, b \sim b', b''$ includes one of the following sub-proofs:

$$\dfrac{\dfrac{}{\ldots, \mathsf{sk(n)} \sim \ldots, \mathsf{sk(n)}}\ \text{CCA}_2}{\vdots}$$
$$\dfrac{\ldots, m, \mathsf{sk(n)} \sim \ldots, m', \mathsf{sk(n)}}{\ldots, \mathsf{dec}(m, \mathsf{sk(n)}) \sim \ldots, \mathsf{dec}(m', \mathsf{sk(n)})}\ \mathsf{FA}_{\setminus \mathbf{0}}$$

  or

$$\dfrac{\dfrac{\dfrac{}{\ldots, m, \mathsf{n} \sim \ldots, m', \mathsf{n}}\ \text{CCA}_2}{\vdots}}{\dfrac{\ldots, m, \mathsf{n} \sim \ldots, m', \mathsf{n}}{\ldots, m, \mathsf{sk(n)} \sim \ldots, m', \mathsf{sk(n)}}\ \mathsf{FA_{sk}}}$$
$$\dfrac{\vdots}{\ldots, m, \mathsf{sk(n)} \sim \ldots, m', \mathsf{sk(n)}}$$
$$\dfrac{}{\ldots, \mathsf{dec}(m, \mathsf{sk(n)}) \sim \ldots, \mathsf{dec}(m', \mathsf{sk(n)})}\ \mathsf{FA}_{\setminus \mathbf{0}}$$

  Hence either $\mathsf{n} \in \vec{w}^r$ or $\mathsf{sk(n)} \in \vec{w}^r$. Absurd. ■

Using this lemma, we can deal with Example 5.7 whenever the proof of $a_1, b_2, b_3 \sim d_1, c_2, d_3$ lies in the fragment $\mathfrak{F}(\mathsf{FA_s}^* \cdot \mathsf{Dup}^* \cdot \text{CCA}_2)$. Using a first time the lemma on $b_2, b_3 \sim c_2, d_3$ we obtain $c \equiv d$, and using again the lemma on $a_1, b_2 \sim d_1, c_2$ (since $d \equiv c$) we deduce $a \equiv b$. Hence:

$$a_1, b_2, b_3, u_4, w_5, u_6, v_7 \sim d_1, c_2, d_3, s_4, t_5, r_6, p_7 \ \equiv\ a_1, a_2, a_3, u_4, w_5, u_6, v_7 \sim c_1, c_2, c_3, s_4, t_5, r_6, p_7$$

Therefore, using Lemma 5.1, we can extract a proof:

$$\begin{array}{c}\vdots\ (A')\\ a_1, u_4, v_7 \sim c_1, s_4, p_7\end{array}$$

Where, we recall, $p \equiv \mathsf{if}\ c\ \mathsf{then}\ s\ \mathsf{else}\ t$. Hence we have the cut elimination:

$$\dfrac{\begin{array}{c}\vdots\ (A')\\ a_1, u_4, v_7 \sim c_1, s_4, p_7\end{array}}{\begin{array}{ccc} & & c_1 \\ & & \diagup\ \diagdown \\ a_1 & & s_4 \quad c \\ \diagup\ \diagdown & \sim & \quad\ \diagup\ \diagdown \\ u_4 \quad v_7 & & \quad\ s \quad t\end{array}}\ \mathsf{FA}_{\backslash \mathbf{0}}$$

$$\dfrac{}{\mathsf{if}\ a\ \mathsf{then}\ u\ \mathsf{else}\ v \sim \mathsf{if}\ c\ \mathsf{then}\ s\ \mathsf{else}\ t}\ R$$

Notice that all sub-terms above are bounded, although the conditional c appears twice on the right.

Proof Sketch We sketch the outline of the completeness proof:

- **Commutations:** first we show that we can assume that rules are applied in some given order. We prove this by showing some commutation results and adding new rules.
- **Proof Cut Eliminations:** through proof cut eliminations, we guarantee that every conditional appearing in the proof is α-bounded. Intuitively a conditional is α-bounded if it is a subterm of the conclusion or if it guards a decryption appearing in an α-bounded term.
- **Decision Procedure:** we give a procedure that, given a goal formula $t \sim t'$, computes the set of α-bounded terms for this formula. We show that this procedure computes a finite set, and deduce that the proof search is finite. This yields an effective algorithm to decide our problem.

5.6 Commutations and Cut Eliminations

In this section we show, through rule commutations, that we can restrict ourselves to proofs using rules in some given order. This is done through two rule commutations lemmas, and a proof cut elimination. In the next section, we show how this restricts the shapes of the terms appearing in a proof.

5.6.1 Rule Commutations

Everything in this subsection applies to any set U of unitary axioms closed under Restr. We specialize to CCA_2 later. We start by showing a set of rule commutations of the form $w \Rightarrow w'$, where w and w' are words over the set of rule names. An entry $w \Rightarrow w'$ means that a derivation in w can be rewritten into a derivation in w', with the same conclusion and premises. Here are the basic commutations we use:

Lemma 5.3. *The following rule commutations are correct:*

$\mathsf{Dup} \cdot R$	$\Rightarrow$	$R \cdot \mathsf{Dup}$
$\mathsf{Dup} \cdot \mathsf{FA}_{\backslash \mathbf{0}}$	$\Rightarrow$	$\mathsf{FA}^*_{\backslash \mathbf{0}} \cdot \mathsf{Dup}$
$\mathsf{Dup} \cdot \mathsf{CS}^{\mathsf{no}}_{\mathsf{if}}$	$\Rightarrow$	$\mathsf{CS}^{\mathsf{no}}_{\mathsf{if}} \cdot \mathsf{Dup}$

$\mathsf{FA}_{\backslash \mathbf{0}} \cdot R$	$\Rightarrow$	$R \cdot \mathsf{FA}_{\backslash \mathbf{0}}$
$\mathsf{FA}_{\backslash \mathbf{0}} \cdot \mathsf{CS}^{\mathsf{no}}_{\mathsf{if}}$	$\Rightarrow$	$R \cdot \mathsf{CS}^{\mathsf{no}}_{\mathsf{if}} \cdot \mathsf{FA}_{\backslash \mathbf{0}}$

Proof. The commutations can be found in Figure 5.6. ∎

Using these rules, we obtain a first restriction.

Lemma 5.4. *For any set of unitary axioms U closed under Restr, the ordered strategy:*

$$\mathfrak{F}((\mathsf{CS}^{\mathsf{no}}_{\mathsf{if}} + R)^* \cdot \mathsf{FA}^*_{\backslash \mathbf{0}} \cdot \mathsf{Dup}^* \cdot \mathsf{U})$$

is complete for $\mathfrak{F}((\mathsf{CS}^{\mathsf{no}}_{\mathsf{if}} + \mathsf{FA}_{\backslash \mathbf{0}} + R + \mathsf{Dup} + \mathsf{U})^*)$.

Delay $\mathsf{FA}_{\setminus \mathbf{0}}$

- $\mathsf{FA}_{\setminus \mathbf{0}} \cdot \mathsf{CS}^{\mathsf{no}}_{\mathsf{if}} \Rightarrow R \cdot \mathsf{CS}^{\mathsf{no}}_{\mathsf{if}} \cdot \mathsf{FA}_{\setminus \mathbf{0}}$:

$$\dfrac{\dfrac{\vec{w}_1, \vec{w}_2, b, (u_i)_{i\in I\cup J} \sim \vec{w}'_1, \vec{w}'_2, b', (u'_i)_{i\in I\cup J} \qquad \vec{w}_1, \vec{w}_2, b, (v_i)_{i\in I\cup J} \sim \vec{w}'_1, \vec{w}'_2, b', (v'_i)_{i\in I\cup J}}{\vec{w}_1, \vec{w}_2, (\mathsf{if}\ b\ \mathsf{then}\ u_i\ \mathsf{else}\ v_i)_{i\in I\cup J} \sim \vec{w}'_1, \vec{w}'_2, (\mathsf{if}\ b'\ \mathsf{then}\ u'_i\ \mathsf{else}\ v'_i)_{i\in I\cup J}}\ \mathsf{CS}^{\mathsf{no}}_{\mathsf{if}}}{\begin{array}{c}\vec{w}_1, (\mathsf{if}\ b\ \mathsf{then}\ u_i\ \mathsf{else}\ v_i)_{i\in I}, f(\vec{w}_2, (\mathsf{if}\ b\ \mathsf{then}\ u_i\ \mathsf{else}\ v_i)_{i\in J}) \\ \sim \quad \vec{w}'_1, (\mathsf{if}\ b'\ \mathsf{then}\ u'_i\ \mathsf{else}\ v'_i)_{i\in I}, f(\vec{w}'_2, (\mathsf{if}\ b'\ \mathsf{then}\ u'_i\ \mathsf{else}\ v'_i)_{i\in J})\end{array}\ \mathsf{FA}_{\setminus \mathbf{0}}}$$

  Can be rewritten into:

$$\dfrac{\dfrac{\dfrac{\vec{w}_1, \vec{w}_2, b, (u_i)_{i\in I\cup J} \sim \vec{w}'_1, \vec{w}'_2, b', (u'_i)_{i\in I\cup J}}{\begin{array}{c}\vec{w}_1, b, (u_i)_{i\in I}, f(\vec{w}_2, (u_i)_{i\in J}) \\ \sim \quad \vec{w}'_1, b', (u'_i)_{i\in I}, f(\vec{w}'_2, (u'_i)_{i\in J})\end{array}\ \mathsf{FA}_{\setminus \mathbf{0}} \qquad \dfrac{\vec{w}_1, \vec{w}_2, b, (v_i)_{i\in I\cup J} \sim \vec{w}'_1, \vec{w}'_2, b', (v'_i)_{i\in I\cup J}}{\begin{array}{c}\vec{w}_1, b, (v_i)_{i\in I}, f(\vec{w}_2, (v_i)_{i\in J}) \\ \sim \quad \vec{w}'_1, b', (v'_i)_{i\in I}, f(\vec{w}'_2, (v'_i)_{i\in J})\end{array}\ \mathsf{FA}_{\setminus \mathbf{0}}}\ \mathsf{CS}^{\mathsf{no}}_{\mathsf{if}}}{\begin{array}{c}\vec{w}_1, (\mathsf{if}\ b\ \mathsf{then}\ u_i\ \mathsf{else}\ v_i)_{i\in I}, \mathsf{if}\ b\ \mathsf{then}\ f(\vec{w}_2, (u_i)_{i\in J})\ \mathsf{else}\ f(\vec{w}_2, (v_i)_{i\in J}) \\ \sim \quad \vec{w}'_1, (\mathsf{if}\ b'\ \mathsf{then}\ u'_i\ \mathsf{else}\ v'_i)_{i\in I}, \mathsf{if}\ b'\ \mathsf{then}\ f(\vec{w}'_2, (u'_i)_{i\in J})\ \mathsf{else}\ f(\vec{w}'_2, (v'_i)_{i\in J})\end{array}\ R}{\begin{array}{c}\vec{w}_1, (\mathsf{if}\ b\ \mathsf{then}\ u_i\ \mathsf{else}\ v_i)_{i\in I}, f(\vec{w}_2, (\mathsf{if}\ b\ \mathsf{then}\ u_i\ \mathsf{else}\ v_i)_{i\in J}) \\ \sim \quad \vec{w}'_1, (\mathsf{if}\ b'\ \mathsf{then}\ u'_i\ \mathsf{else}\ v'_i)_{i\in I}, f(\vec{w}'_2, (\mathsf{if}\ b'\ \mathsf{then}\ u'_i\ \mathsf{else}\ v'_i)_{i\in J})\end{array}}$$

- $\mathsf{FA}_{\setminus \mathbf{0}} \cdot R \Rightarrow R \cdot \mathsf{FA}_{\setminus \mathbf{0}}$:

$$\dfrac{\dfrac{\dfrac{\vec{u}_1, \vec{v}_1 \sim \vec{u}'_1, \vec{v}'_1}{\vec{u}, \vec{v} \sim \vec{u}', \vec{v}'}\ R}{\vec{u}, f(\vec{v}) \sim \vec{u}', f(\vec{v}')}\ \mathsf{FA}_{\setminus \mathbf{0}}}{} \quad \Rightarrow \quad \dfrac{\dfrac{\vec{u}_1, \vec{v}_1 \sim \vec{u}'_1, \vec{v}'_1}{\vec{u}_1, f(\vec{v}_1) \sim \vec{u}'_1, f(\vec{v}'_1)}\ \mathsf{FA}_{\setminus \mathbf{0}}}{\vec{u}, f(\vec{v}) \sim \vec{u}', f(\vec{v}')}\ R$$

Delay Dup

- $\mathsf{Dup} \cdot R \Rightarrow R \cdot \mathsf{Dup}$.

  If the R rules involves a term which is not duplicated then this is trivial. Assume the R rewriting involves a duplicated term, and that $t =_R s$ and $t' =_R s'$:

$$\dfrac{\dfrac{\vec{u}, \vec{v}, s \sim \vec{u}', \vec{v}', s'}{\vec{u}, \vec{v}, t \sim \vec{u}', \vec{v}', t'}\ R}{\vec{u}, \vec{v}, t, \vec{v}, t \sim \vec{u}', \vec{v}', t', \vec{v}', t'}\ \mathsf{Dup} \quad \Rightarrow \quad \dfrac{\dfrac{\vec{u}, \vec{v}, s \sim \vec{u}', \vec{v}', s'}{\vec{u}, \vec{v}, s, \vec{v}, s \sim \vec{u}', \vec{v}', s', \vec{v}', s'}\ \mathsf{Dup}}{\vec{u}, \vec{v}, t, \vec{v}, t \sim \vec{u}', \vec{v}', t', \vec{v}', t'}\ R$$

- $\mathsf{Dup} \cdot \mathsf{FA}_{\setminus \mathbf{0}} \Rightarrow \mathsf{FA}^*_{\setminus \mathbf{0}} \cdot \mathsf{Dup}$.

  Similarly if the $\mathsf{FA}_{\setminus \mathbf{0}}$ rules does not involve a duplicated term then this is trivial. Otherwise:

$$\dfrac{\dfrac{\vec{u}, \vec{v}, \vec{w} \sim \vec{u}', \vec{v}', \vec{w}'}{\vec{u}, \vec{v}, f(\vec{w}) \sim \vec{u}', \vec{v}', f(\vec{w}')}\ \mathsf{FA}_{\setminus \mathbf{0}}}{\vec{u}, \vec{v}, f(\vec{w}), \vec{v}, f(\vec{w}) \sim \vec{u}', \vec{v}', f(\vec{w}'), \vec{v}', f(\vec{w}')}\ \mathsf{Dup} \quad \Rightarrow \quad \dfrac{\dfrac{\dfrac{\vec{u} \sim \vec{u}'}{\vec{u}, \vec{v}, \vec{w}, \vec{v}, \vec{w} \sim \vec{u}', \vec{v}', \vec{w}', \vec{v}', \vec{w}'}\ \mathsf{Dup}}{\vec{u}, \vec{v}, f(\vec{w}), \vec{v}, \vec{w} \sim \vec{u}', \vec{v}', f(\vec{w}'), \vec{v}', \vec{w}'}\ \mathsf{FA}_{\setminus \mathbf{0}}}{\vec{u}, \vec{v}, f(\vec{w}), \vec{v}, f(\vec{w}) \sim \vec{u}', \vec{v}', f(\vec{w}'), \vec{v}', f(\vec{w}')}\ \mathsf{FA}_{\setminus \mathbf{0}}$$

- $\mathsf{Dup} \cdot \mathsf{CS}^{\mathsf{no}}_{\mathsf{if}} \Rightarrow \mathsf{CS}^{\mathsf{no}}_{\mathsf{if}} \cdot \mathsf{Dup}$. Commutation of Dup with $\mathsf{CS}^{\mathsf{no}}_{\mathsf{if}}$ is similar.

Figure 5.6: Function Application and Duplicate Rules Commutations

Proof. Using Lemma 5.3, we commute all the Dup to the right, which yields $\mathfrak{F}((\mathsf{CS}^{\mathsf{no}}_{\mathsf{if}}+R+\mathsf{FA}_{\setminus \mathbf{0}})^*\cdot\mathsf{Dup}^*\cdot\mathsf{U})$. Then, we commute all $\mathsf{FA}_{\setminus \mathbf{0}}$ to the right, stopping at the first Dup. ■

Example 5.8. We give an example of such a proof rewriting:

$$\dfrac{\dfrac{\dfrac{x\sim z}{\pi_1(\langle x\,,\,y\rangle)\sim z}\,R}{g(\pi_1(\langle x\,,\,y\rangle))\sim g(z)}\,\mathsf{FA}_{\setminus \mathbf{0}}}{g(\pi_1(\langle x\,,\,y\rangle)),g(\pi_1(\langle x\,,\,y\rangle))\sim g(z),g(z)}\,\mathsf{Dup} \quad\Rightarrow\quad \dfrac{\dfrac{\dfrac{\dfrac{x\sim z}{x,x\sim z,z}\,\mathsf{Dup}}{x,g(x)\sim z,g(z)}\,\mathsf{FA}_{\setminus \mathbf{0}}}{g(x),g(x)\sim g(z),g(z)}\,\mathsf{FA}_{\setminus \mathbf{0}}}{g(\pi_1(\langle x\,,\,y\rangle)),g(\pi_1(\langle x\,,\,y\rangle))\sim g(z),g(z)}\,R$$ □

Splitting the $\mathsf{FA}_{\setminus \mathbf{0}}$ Rule To go further, we split the function application rules $\mathsf{FA}_{\setminus \mathbf{0}}$ as follows: if the deconstructed symbol is $\mathsf{if\_then\_else\_}$ then we denote the function application by $\mathsf{FA}_{\setminus \mathbf{0}}(b,b')$, where b,b' are the involved conditionals; if the deconstructed symbol f is in $\mathcal{F}_{\setminus \mathsf{if},\mathbf{0}}$, then we denote the function application by FA_f. We give below the two new rules:

$$\dfrac{\vec{w},a,u,v\sim\vec{r},b,s,t}{\vec{w},\mathsf{if}\ a\ \mathsf{then}\ u\ \mathsf{else}\ v\sim\vec{r},\mathsf{if}\ b\ \mathsf{then}\ s\ \mathsf{else}\ t}\,\mathsf{FA}(b,b') \qquad \dfrac{\vec{u},\vec{v}\sim\vec{s},\vec{t}}{\vec{u},f(\vec{v})\sim\vec{s},f(\vec{t})}\,\mathsf{FA}_f$$

The set of rule names is now infinite, since there is a rule $\mathsf{FA}_{\setminus \mathbf{0}}(b,b')$ for every pair of ground terms b,b'.

Intuitively, we want to use R at the beginning of the proof only. This is helpful since, as we observed earlier, all the other rules are decreasing (i.e. premises are smaller than the conclusion). The problem is that we cannot fully commute $\mathsf{CS}^{\mathsf{no}}_{\mathsf{if}}$ and R. For example, in:

$$\dfrac{\dfrac{a_1,u_1\sim c_1,s_1}{a,u\sim c,s}\,R \qquad \dfrac{a_2,v_1\sim c_2,t_1}{a,v\sim c,t}\,R}{\mathsf{if}\ a\ \mathsf{then}\ u\ \mathsf{else}\ v\sim\mathsf{if}\ c\ \mathsf{then}\ s\ \mathsf{else}\ t}\,\mathsf{CS}^{\mathsf{no}}_{\mathsf{if}} \tag{5.5}$$

we can commute the rewritings on u,v,s and t, but not on a and c because they appear twice in the premises, and a_1 and a_2 may be different (same for c_1 and c_2).

We solve this by adding new rules to track relations between branches. We first give simplified versions. For every if-free ground conditionals a and c in R-normal form, we introduce the rules:

$$\dfrac{\vec{u},C\big[\boxed{a}\boxed{a}_a\big]\sim\vec{v},C'\big[\boxed{c}\boxed{c}_c\big]}{\vec{u},C[a]\sim\vec{v},C'[c]}\,\mathsf{2Box}^{\mathsf{s}} \qquad \dfrac{a_1,u\sim c_1,s \qquad a_2,v\sim c_2,t}{\mathsf{if}\ \boxed{a_1}\boxed{a_2}_a\ \mathsf{then}\ u\ \mathsf{else}\ v\sim\mathsf{if}\ \boxed{c_1}\boxed{c_2}_c\ \mathsf{then}\ s\ \mathsf{else}\ t}\,\mathsf{CS}^{\mathsf{s}}_{\square}$$

where $\boxed{\ }\boxed{\ }_a$ is a new symbol of sort $\mathsf{bool}^2\to\mathsf{bool}$, and of fixed semantics: it ignores its arguments and has the semantics $[\![a]\!]$. Intuitively, $\boxed{a_1}\boxed{a_2}_a$ stands for the conditional a, and a_1,a_2 are, respectively, the left and right versions of a. Then, using these rules, we can rewrite the derivation in (5.5):

$$\dfrac{\dfrac{\dfrac{a_1,u_1\sim c_1,s_1 \qquad a_2,v_1\sim c_2,t_1}{\mathsf{if}\ \boxed{a_1}\boxed{a_2}_a\ \mathsf{then}\ u_1\ \mathsf{else}\ v_1\sim\mathsf{if}\ \boxed{c_1}\boxed{c_2}_c\ \mathsf{then}\ s_1\ \mathsf{else}\ t_1}\,\mathsf{CS}^{\mathsf{b}}_{\square}}{\mathsf{if}\ \boxed{a}\boxed{a}_a\ \mathsf{then}\ u\ \mathsf{else}\ v\sim\mathsf{if}\ \boxed{c}\boxed{c}_c\ \mathsf{then}\ s\ \mathsf{else}\ t}\,R}{\mathsf{if}\ a\ \mathsf{then}\ u\ \mathsf{else}\ v\sim\mathsf{if}\ c\ \mathsf{then}\ s\ \mathsf{else}\ t}\,\mathsf{2Box}^{\mathsf{s}}$$

The $\mathsf{2Box}^{\mathsf{s}}$ allows to introduce two versions of a and c, which can be independently rewritten. Using this, we can do both rewritings before applying the $\mathsf{CS}^{\mathsf{s}}_{\square}$ rule.

Let us define formally the unrestricted rules. First, we denote $\mathcal{B}$ the set of new function symbols.

Definition 5.14. We let $\mathcal{B}$ be the set of function symbols:

$$\mathcal{F}\cup\{\boxed{\_}\boxed{\_}_b \mid b \text{ if-free ground conditional}\}$$

We need the functions in $\mathcal{B}$ to block the if-homomorphism to ensure that for all $\boxed{a}\boxed{c}_b\in\mathsf{st}(t)$, $[\![a]\!]=[\![c]\!]=[\![b]\!]$. Therefore the set of equalities R_2 is *not* extended to $\mathcal{B}$. For example we have:

$$\boxed{\mathsf{if}\ a\ \mathsf{then}\ c\ \mathsf{else}\ d}\boxed{e}_b \not\to_R^*\mathsf{if}\ a\ \mathsf{then}\ \boxed{c}\boxed{e}_b\ \mathsf{else}\ \boxed{d}\boxed{e}_b$$

The R rule is replaced by $R_\square$ which has an extra side-condition: $R_\square$ can rewrite $\vec{w},u[s]$ into $\vec{w},u[t]$ as long as $\vec{w},u[s]$'s boxed conditionals $\{\boxed{a}\boxed{c}_b\in\mathsf{st}(\vec{w},u[s])\}$ contain t's boxed conditionals $\{\boxed{a}\boxed{c}_b\in\mathsf{st}(t)\}$.

Definition 5.15. We let $R_\square$ be the following axiom schema:

$$\frac{\vec{w}, u[t] \sim \vec{v}}{\vec{w}, u[s] \sim \vec{v}}\ R_\square \qquad \text{when } s =_R t \text{ and } \{\boxed{a}\boxed{c}_b \in \mathsf{st}(t)\} \subseteq \{\boxed{a}\boxed{c}_b \in \mathsf{st}(\vec{w}, u[s])\}$$

The side-condition ensures that no new arbitrary $\boxed{a}\boxed{c}_b$ is introduced. New boxed conditionals are only introduced through the 2Box rule. Similarly, the $\mathsf{FA}_{\setminus \mathbf{0}}$ axiom is *not* extended to $\mathcal{B}$: boxed conditionals can only be open using the $\mathsf{CS}_\square$ rule.

Example 5.9. We give two examples of valid application of the $R_\square$ rules. The first $R_\square$ application is valid because we do not introduce any boxed conditional on the left, and because we remove a boxed conditional on the right. The second $R_\square$ application is valid because the introduced boxed conditional already appears in the conclusion:

$$\frac{\mathsf{if}\ \mathsf{eq}(g(\{0\}^{\mathsf{n}}_{\mathsf{pk}}), \{0\}^{\mathsf{n}}_{\mathsf{pk}})\ \mathsf{then}\ \mathsf{dec}(g(\{0\}^{\mathsf{n}}_{\mathsf{pk}}), \mathsf{sk}) \sim v \quad \mathsf{else}\ \mathsf{dec}(g(\{0\}^{\mathsf{n}}_{\mathsf{pk}}), \mathsf{sk})}{\mathsf{dec}(g(\{0\}^{\mathsf{n}}_{\mathsf{pk}}), \mathsf{sk}) \sim \mathsf{if}\ \boxed{a}\boxed{c}_b\ \mathsf{then}\ v\ \mathsf{else}\ v}\ R_\square \qquad \frac{\mathsf{if}\ \boxed{a}\boxed{c}_b\ \mathsf{then}\ (\mathsf{if}\ \boxed{a}\boxed{c}_b\ \mathsf{then}\ u\ \mathsf{else}\ w) \sim t \quad \mathsf{else}\ v}{\mathsf{if}\ \boxed{a}\boxed{c}_b\ \mathsf{then}\ u\ \mathsf{else}\ v \sim t}\ R_\square$$

□

When boxing a conditional c, we want the term $c_\square$ indexing the box $\boxed{c}\boxed{c}_{c_\square}$ to characterize c's semantics in a proof invariant way. By consequence, we replace all boxes $\boxed{a_1}\boxed{a_2}_a$ in c by a, and we normalize the resulting term. Formally, we introduce the following erasure function which removes boxed conditional:

Definition 5.16. We let 2erase be the function defined on if-free ground terms by:

$$\mathsf{2erase}(t) \equiv \begin{cases} \mathsf{2erase}(b) & \text{if } t \equiv \boxed{b_1}\boxed{b_2}_b \\ \mathsf{n} & \text{if } t \equiv \mathsf{n} \text{ and } \mathsf{n} \in \mathcal{N} \\ f(\mathsf{2erase}(t_1), \ldots, \mathsf{2erase}(t_n)) & \text{if } t \equiv f(t_1, \ldots, t_n) \text{ and } f \neq \mathsf{if\_then\_else\_} \end{cases}$$

Example 5.10. We give a simple example with a term containing only one boxed conditional $\boxed{a}\boxed{c}_b$:

$$\mathsf{2erase}\left(\mathsf{eq}(\mathsf{if}\ \boxed{a}\boxed{c}_b\ \mathsf{then}\ u\ \mathsf{else}\ v, \mathsf{A})\right) \equiv \mathsf{eq}(\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v, \mathsf{A})$$ □

This function is used to define the full (not simplified) versions of 2Box and $\mathsf{CS}_\square$:

Definition 5.17. We let 2Box and $\mathsf{CS}_\square$ be the axioms:

$$\frac{\vec{u}, C\left[\boxed{a}\boxed{a}_{\mathsf{2erase}(a)\downarrow_R}\right] \sim \vec{u}', C'\left[\boxed{a'}\boxed{a'}_{\mathsf{2erase}(a')\downarrow_R}\right]}{\vec{u}, C[a] \sim \vec{u}', C'[a']}\ \mathsf{2Box} \qquad \text{when } a, a' \in \mathcal{T}(\mathcal{F}_{\setminus \mathsf{if}} \cup \mathcal{B}, \mathcal{N})$$

$$\frac{\vec{w}, a_1, (u_i)_i \sim \vec{w}', a_1', (u_i')_i \qquad \vec{w}, a_2, (v_i)_i \sim \vec{w}', a_2', (v_i')_i}{\vec{w}, \left(\mathsf{if}\ \boxed{a_1}\boxed{a_2}_a\ \mathsf{then}\ u_i\ \mathsf{else}\ v_i\right)_i \sim \vec{w}', \left(\mathsf{if}\ \boxed{a_1'}\boxed{a_2'}_{a'}\ \mathsf{then}\ u_i'\ \mathsf{else}\ v_i'\right)_i}\ \mathsf{CS}_\square \qquad \text{when } a, a' \in \mathcal{T}(\mathcal{F}_{\setminus \mathsf{if}}, \mathcal{N})$$

Remark that for the $\mathsf{CS}_\square$ rule to be sound we need $[\![a_1]\!]$, $[\![a_2]\!]$ and $[\![a]\!]$ to be equal, up to a negligible number of samplings (same for a_1', a_2' and a). This is not enforced by the rules, so it has to be an invariant of our strategy.

Definition 5.18. A term t is *well-formed* if and only if for every $\boxed{a}\boxed{c}_b \in \mathsf{st}(t)$, $a =_R c =_R b$. We lift this to formulas as expected.

Proposition 5.6. *The following rules preserve well-formedness:*

$$R_\square, \mathit{2Box}, \mathit{CS}_\square, \mathit{FA}_s, \{\mathit{FA}_{\setminus \mathbf{0}}(b, b')\}, \mathit{Dup}$$

Besides, $R_\square$, $\mathit{CS}_\square$ and 2Box are sound on well-formed formulas.

Proof. The only rule not obviously preserving well-formedness is $R_\square$, but its side-conditions guarantee the well-formedness invariant. The only rule that is not always sound is $\mathsf{CS}_\square$, and it is trivially sound on well-formed formulas. ■

Remark 5.7. We extend cond-st to terms in $\mathcal{T}(\mathcal{B}, \mathcal{N})$ in a non-obvious way, by erasing all boxes. Formally, for all $t \in \mathcal{T}(\mathcal{B}, \mathcal{N})$, we let:

$$\mathsf{cond\text{-}st}(t) = \mathsf{cond\text{-}st}(\mathsf{2erase}(t))$$ □

Ordered Strategy We can now give the new rule commutations.

Lemma 5.5. *The following rule commutations are correct:*

$$\boxed{\begin{array}{lcl} \mathit{FA_s} \cdot \mathit{FA}_{\setminus \mathbf{0}}(b,b') & \Rightarrow & R \cdot \mathit{FA}_{\setminus \mathbf{0}}(b,b') \cdot \mathit{FA_s}^* \cdot \mathit{Dup} \end{array}}$$

$$\boxed{\begin{array}{lcl} \mathit{CS}_\Box \cdot R_\Box & \Rightarrow & R_\Box \cdot \mathit{CS}_\Box \\ \mathit{CS}_\Box \cdot \mathit{2Box} & \Rightarrow & R_\Box \cdot \mathit{2Box} \cdot \mathit{CS}_\Box \end{array}}$$

Proof. The rule commutations can be found in Figure 5.7. ■

This allows to have $R_\Box$ rules only at the beginning of the proof.

Lemma 5.6. *For any set of unitary axioms U closed under Restr, the ordered strategy:*

$$\mathfrak{F}((\mathit{2Box} + R_\Box)^* \cdot \mathit{CS}_\Box^* \cdot \{\mathit{FA}_{\setminus \mathbf{0}}(b,b')\}^* \cdot \mathit{FA_s}^* \cdot \mathit{Dup}^* \cdot \mathsf{U})$$

is complete for $\mathfrak{F}((\mathit{CS}_{\mathit{if}}^{\mathit{no}} + \mathit{FA}_{\setminus \mathbf{0}} + R + \mathit{Dup} + \mathsf{U})^*)$.

Proof. We start from the result of Lemma 5.4, split the $\mathsf{FA}_{\setminus \mathbf{0}}$ rules and commute rules until we get:

$$\mathfrak{F}((\mathsf{CS}_{\mathsf{if}}^{\mathsf{no}} + R)^* \cdot \{\mathsf{FA}_{\setminus \mathbf{0}}(b,b')\}^* \cdot \mathsf{FA_s}^* \cdot \mathsf{Dup}^* \cdot \mathsf{U})$$

We then replace all applications of $\mathsf{CS}_{\mathsf{if}}^{\mathsf{no}}$ by $\mathsf{2Box} \cdot \mathsf{CS}_\Box$. All $\boxed{a}\boxed{a}_a$ introduced are immediately "opened" by a $\mathsf{CS}_\Box$ application, hence we know that the side-conditions of $R_\Box$ hold every time we apply R. Therefore we can replace all applications of R by $R_\Box$, which yields:

$$\mathfrak{F}((\mathsf{CS}_\Box + \mathsf{2Box} + R_\Box)^* \cdot \{\mathsf{FA}_{\setminus \mathbf{0}}(b,b')\}^* \cdot \mathsf{FA_s}^* \cdot \mathsf{Dup}^* \cdot \mathsf{U})$$

Finally we commute the $\mathsf{CS}_\Box$ applications to the right. ■

5.6.2 The Freeze Strategy

We now show that we can restrict the terms on which the rules in $\{\mathsf{FA}_{\setminus \mathbf{0}}(b,b')\}$ can be applied: when we apply a rule in $\{\mathsf{FA}_{\setminus \mathbf{0}}(b,b')\}$, we "freeze" the conditionals b and b' to forbid any further applications of $\{\mathsf{FA}_{\setminus \mathbf{0}}(b,b')\}$ to them.

Example 5.11. Let $a_i \equiv \mathsf{if}\ b_i\ \mathsf{then}\ c_i\ \mathsf{else}\ d_i$ $(i \in \{1,2\})$, we want to forbid the following partial derivation to appear:

$$\dfrac{\dfrac{b_1, c_1, d_1, u_1, v_1 \sim b_2, c_2, d_2, u_2, v_2}{a_1, u_1, v_1 \sim a_2, u_2, v_2}\ \mathsf{FA}_{\setminus \mathbf{0}}(b_1, b_2)}{\mathsf{if}\ a_1\ \mathsf{then}\ u_1\ \mathsf{else}\ v_1 \sim \mathsf{if}\ a_2\ \mathsf{then}\ u_2\ \mathsf{else}\ v_2}\ \mathsf{FA}_{\setminus \mathbf{0}}(a_1, a_2)$$

□

For this, we define a new function symbol $\overline{\ }$ arity one, which allows to freeze a conditional and prevent applications of $\{\mathsf{FA}_{\setminus \mathbf{0}}(b,b')\}$. Basically, when we apply a rule in $\{\mathsf{FA}_{\setminus \mathbf{0}}(b,b')\}$ on the conditionals b_1 and b_2:

$$b_1 \equiv \mathsf{if}\ a_1\ \mathsf{then}\ u_1\ \mathsf{else}\ v_1 \qquad\qquad b_2 \equiv \mathsf{if}\ a_2\ \mathsf{then}\ u_2\ \mathsf{else}\ v_2$$

We replace, in the premise, a_1 by $\overline{a_1}$ in b_1 and a_2 by $\overline{a_2}$ in b_2. Then, we show that we can restrict ourselves to proofs where we never apply $\mathsf{FA}_{\setminus \mathbf{0}}$ on a frozen $\mathsf{if\_then\_else\_}$ conditional.

Definition 5.19. Let $\overline{\ }$ be a new function symbol of arity one. For every ground term s, we let $\widetilde{s}$ be:

$$\widetilde{s} \equiv \begin{cases} \mathsf{if}\ \overline{b}\ \mathsf{then}\ u\ \mathsf{else}\ v & \text{if } s \equiv \mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v \\ s & \text{if } s \in \mathcal{T}(\mathcal{F}_{\setminus \mathsf{if}}, \mathcal{N}) \end{cases}$$

Moreover we replace every $\mathsf{FA}_{\setminus \mathbf{0}}(b_1, b_2)$ by the rule $\mathsf{BFA}(b_1, b_2)$ which freezes conditionals b_1 and b_2:

- $\mathsf{FA_s} \cdot \mathsf{FA}_{\setminus \mathbf{0}}(b, b') \;\Rightarrow\; R \cdot \mathsf{FA}_{\setminus \mathbf{0}}(b, b') \cdot \mathsf{FA_s}^* \cdot \mathsf{Dup}$

$$\dfrac{\dfrac{\vec{u}, \vec{v}, b, s, t \sim \vec{u}', \vec{v}', b', s', t'}{\vec{u}, \vec{v}, \mathsf{if}\ b\ \mathsf{then}\ s\ \mathsf{else}\ t \sim \vec{u}', \vec{v}', \mathsf{if}\ b'\ \mathsf{then}\ s'\ \mathsf{else}\ t'}\ \mathsf{FA}_{\setminus \mathbf{0}}(b, b')}{\vec{u}, f(\vec{v}, \mathsf{if}\ b\ \mathsf{then}\ s\ \mathsf{else}\ t) \sim \vec{u}', f(\vec{v}', \mathsf{if}\ b'\ \mathsf{then}\ s'\ \mathsf{else}\ t')}\ \mathsf{FA}_f$$

Can be rewritten into:

$$\dfrac{\dfrac{\dfrac{\dfrac{\vec{u}, b, s, \vec{v}, t \sim \vec{u}', b', s', \vec{v}', t'}{\vec{u}, b, \vec{v}, s, \vec{v}, t \sim \vec{u}', b', \vec{v}', s', \vec{v}', t'}\ \mathsf{Dup}}{\vec{u}, b, f(\vec{v}, s), f(\vec{v}, t) \sim \vec{u}', b', f(\vec{v}', s'), f(\vec{v}', t')}\ \mathsf{FA}_f^{(2)}}{\vec{u}, \mathsf{if}\ b\ \mathsf{then}\ f(\vec{v}, s)\ \mathsf{else}\ f(\vec{v}, t) \sim \vec{u}', \mathsf{if}\ b'\ \mathsf{then}\ f(\vec{v}', s')\ \mathsf{else}\ f(\vec{v}', t')}\ \mathsf{FA}_{\setminus \mathbf{0}}(b, b')}{\vec{u}, f(\vec{v}, \mathsf{if}\ b\ \mathsf{then}\ s\ \mathsf{else}\ t) \sim \vec{u}', f(\vec{v}', \mathsf{if}\ b'\ \mathsf{then}\ s'\ \mathsf{else}\ t')}\ R$$

- $\mathsf{CS}_\square \cdot R_\square \;\Rightarrow\; R_\square \cdot \mathsf{CS}_\square$:

$$\dfrac{\dfrac{(w_j^1)_j, b_1, (u_i^1)_i \sim (w_j'^1)_j, b_1', (u_i'^1)_i}{(w_j)_j, a_1, (u_i)_i \sim (w_j')_j, a_1', (u_i')_i}\ R_\square \quad \dfrac{(w_j^2)_j, b_2, (v_i^1)_i \sim (w_j'^2)_j, b_2', (v_i'^1)_i}{(w_j)_j, a_2, (v_i)_i \sim (w_j')_j, a_2', (v_i')_i}\ R_\square}{(w_j)_j, (\mathsf{if}\ \boxed{a_1 \mid a_2}_b\ \mathsf{then}\ u_i\ \mathsf{else}\ v_i)_i \sim (w_j')_j, (\mathsf{if}\ \boxed{a_1' \mid a_2'}_{b'}\ \mathsf{then}\ u_i'\ \mathsf{else}\ v_i')_i}\ \mathsf{CS}_\square$$

can be rewritten into:

$$\dfrac{\begin{array}{c}\dfrac{(w_j^1)_j, b_1, (u_i^1)_i \sim (w_j'^1)_j, b_1', (u_i'^1)_i \quad (w_j^2)_j, b_2, (v_i^1)_i \sim (w_j'^1)_j, b_2', (v_i'^1)_i}{(\mathsf{if}\ \boxed{b_1 \mid b_2}_b\ \mathsf{then}\ w_j^1\ \mathsf{else}\ w_j^2)_j, (\mathsf{if}\ \boxed{b_1 \mid b_2}_b\ \mathsf{then}\ u_i^1\ \mathsf{else}\ v_i^1)_i}\ \mathsf{CS}_\square \\ \sim \quad (\mathsf{if}\ \boxed{b_1' \mid b_2'}_{b'}\ \mathsf{then}\ w_j'^1\ \mathsf{else}\ w_j'^2)_j, (\mathsf{if}\ \boxed{b_1' \mid b_2'}_{b'}\ \mathsf{then}\ u_i'^1\ \mathsf{else}\ v_i'^1)_i\end{array}}{(w_j)_j, (\mathsf{if}\ \boxed{a_1 \mid a_2}_b\ \mathsf{then}\ u_i\ \mathsf{else}\ v_i)_i \sim (w_j')_j, (\mathsf{if}\ \boxed{a_1' \mid a_2'}_{b'}\ \mathsf{then}\ u_i'\ \mathsf{else}\ v_i')_i}\ R_\square$$

- $\mathsf{CS}_\square \cdot \mathsf{2Box} \;\Rightarrow\; R_\square \cdot \mathsf{2Box} \cdot \mathsf{CS}_\square$. Let $b, b' \in \mathcal{T}(\mathcal{F}_{\setminus \mathsf{if}} \cup \mathcal{B}, \mathcal{N})$, and let:

$$b_\square \equiv \boxed{b \mid b}_{\mathsf{2erase}(b)\downarrow_R} \qquad \text{and} \qquad b'_\square \equiv \boxed{b' \mid b'}_{\mathsf{2erase}(b')\downarrow_R}$$

Then the following proof:

$$\dfrac{\dfrac{(w_j[b_\square])_j, a_1[b_\square], (u_i[b_\square])_i \sim (w_j'[b_\square'])_j, a_1'[b_\square'], (u_i'[b_\square'])_i}{(w_j[b])_j, a_1[b], (u_i[b])_i \sim (w_j'[b'])_j, a_1'[b'], (u_i'[b'])_i}\ \mathsf{2Box} \qquad \begin{array}{c}(w_j[b])_j, a_2[b], (v_i[b])_i \\ \sim (w_j'[b'])_j, a_2'[b'], (v_i'[b'])_i\end{array}}{(w_j[b])_j, \big(\mathsf{if}\ \boxed{a_1[b] \mid a_2[b]}_a\ \mathsf{then}\ u_i[b]\ \mathsf{else}\ v_i[b]\big)_i \sim (w_j'[b'])_j, \big(\mathsf{if}\ \boxed{a_1'[b'] \mid a_2'[b']}_{a'}\ \mathsf{then}\ u_i'[b']\ \mathsf{else}\ v_i'[b']\big)_i}\ \mathsf{CS}_\square$$

can be rewritten into:

$$\dfrac{\dfrac{\begin{array}{c}(w_j[b_\square])_j, a_1[b_\square], (u_i[b_\square])_i \\ \sim \quad (w_j'[b_\square'])_j, a_1'[b_\square'], (u_i'[b_\square'])_i\end{array} \qquad \begin{array}{c}(w_j[b])_j, a_2[b], (v_i[b])_i \\ \sim \quad (w_j'[b'])_j, a_2'[b'], (v_i'[b'])_i\end{array}}{\begin{array}{c}\Big(\mathsf{if}\ \boxed{a_1[b_\square] \mid a_2[b]}_a\ \mathsf{then}\ w_j[b_\square]\ \mathsf{else}\ w_j[b]\Big)_j, \Big(\mathsf{if}\ \boxed{a_1[b_\square] \mid a_2[b]}_a\ \mathsf{then}\ u_i[b_\square]\ \mathsf{else}\ v_i[b]\Big)_i \\ \sim \quad \Big(\mathsf{if}\ \boxed{a_1'[b_\square'] \mid a_2'[b']}_{a'}\ \mathsf{then}\ w_j'[b_\square']\ \mathsf{else}\ w_j'[b']\Big)_j, \Big(\mathsf{if}\ \boxed{a_1'[b_\square'] \mid a_2'[b']}_{a'}\ \mathsf{then}\ u_i'[b_\square']\ \mathsf{else}\ v_i'[b']\Big)_i\end{array}}\ \mathsf{CS}_\square$$

$$\dfrac{\begin{array}{c}\Big(\mathsf{if}\ \boxed{a_1[b] \mid a_2[b]}_a\ \mathsf{then}\ w_j[b]\ \mathsf{else}\ w_j[b]\Big)_j, \Big(\mathsf{if}\ \boxed{a_1[b] \mid a_2[b]}_a\ \mathsf{then}\ u_i[b]\ \mathsf{else}\ v_i[b]\Big)_i \\ \sim \quad \Big(\mathsf{if}\ \boxed{a_1'[b'] \mid a_2'[b']}_{a'}\ \mathsf{then}\ w_j'[b']\ \mathsf{else}\ w_j'[b']\Big)_j, \Big(\mathsf{if}\ \boxed{a_1'[b'] \mid a_2'[b']}_{a'}\ \mathsf{then}\ u_i'[b']\ \mathsf{else}\ v_i'[b']\Big)_i\end{array}}{\begin{array}{c}(w_j[b])_j, \Big(\mathsf{if}\ \boxed{a_1[b] \mid a_2[b]}_a\ \mathsf{then}\ u_i[b]\ \mathsf{else}\ v_i[b]\Big)_i \\ \sim \quad (w_j'[b'])_j, \Big(\mathsf{if}\ \boxed{a_1'[b'] \mid a_2'[b']}_{a'}\ \mathsf{then}\ u_i'[b']\ \mathsf{else}\ v_i'[b']\Big)_i\end{array}}\ R_\square$$

The commutation with an application of 2Box in the right branch is exactly the same.

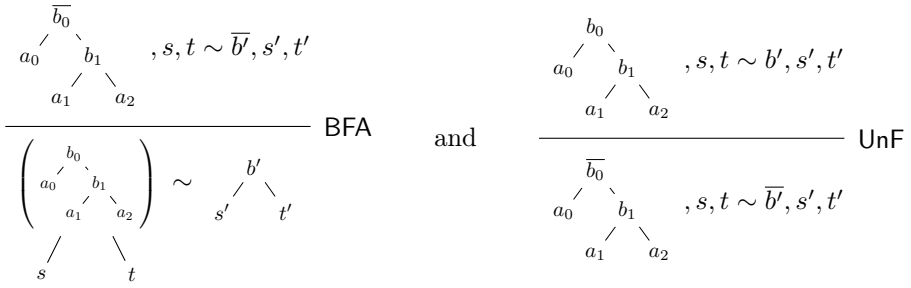
Figure 5.7: Function Application and Boxed Case Study Rules Commutations

Definition 5.20. We let BFA be the rule:

$$\frac{\vec{w}_1, \widetilde{b_1}, u_1, v_1 \sim \vec{w}_2, \widetilde{b_2}, u_2, v_2}{\vec{w}_1, \mathsf{if}\ b_1\ \mathsf{then}\ u_1\ \mathsf{else}\ v_1 \sim \vec{w}_2, \mathsf{if}\ b_2\ \mathsf{then}\ u_2\ \mathsf{else}\ v_2}\ \mathsf{BFA}(b_1, b_2)$$

We let $\{\overline{\mathsf{BFA}}(b_1, b_2)\}$ be the restriction of $\{\mathsf{BFA}(b_1, b_2)\}$ to instances where b_1 and b_2 are not frozen. Finally, we let UnF be the rule which unfreezes all conditionals: every $\overline{b}$ is replaced by b.

Example 5.12. If the conditionals b' is if-free then:



□

We can extend the Restr elimination procedure of Lemma 5.1 to deal with the new rules $\mathsf{CS}_\Box$ and 2Box (but not $R_\Box$):

Lemma 5.7. *If $P \vdash \vec{u} \sim \vec{v}$ with P in the fragment:*

$$\mathfrak{F}\left((\mathit{CS}_\Box + \mathit{2Box} + \mathit{FA}_{\setminus 0} + \mathit{Dup} + \mathrm{CCA}_2 + \mathit{Restr})^*\right)$$

then there exists P' such that $P' \vdash \vec{u} \sim \vec{v}$ and P' contains no Restr applications. Moreover:

- *the height of P' is no larger than the height of P.*
- *if P is in a fragment $\mathfrak{F}(\mathcal{L})$ where $\mathcal{L}$ is closed by sub-words then P' is in $\mathfrak{F}(\mathcal{L})$.*

Proof. This is the same proof than for Lemma 5.1, without the R case and replacing the $\mathsf{CS}^{\mathsf{no}}_{\mathsf{if}}$ axiom by the $\mathsf{CS}_\Box$ axiom. Note that the 2Box rule is trivial to handle. ■

We can state the following ordered strategy lemma:

Lemma 5.8. *For any set of unitary axioms U closed under Restr, the ordered strategy:*

$$\mathfrak{F}((\mathit{2Box} + R_\Box)^* \cdot \mathit{CS}^*_\Box \cdot \{\overline{\mathit{BFA}}(b, b')\}^* \cdot \mathit{UnF} \cdot \mathit{FA_s}^* \cdot \mathit{Dup}^* \cdot U)$$

is complete for $\mathfrak{F}((\mathit{CS}^{\mathit{no}}_{\mathit{if}} + \mathit{FA}_{\setminus 0} + R + \mathit{Dup} + U)^*)$.

Basically, the proof consists in eliminating all proof cuts of the shape given in Example 5.11. The cut elimination is simple, though voluminous. Before starting the proof, we define the induction ordering used in the proof.

Proof ordering Let us consider the following well-founded order on proofs: a proof is interpreted by the multi-set of pair (b, b') appearing as (potentially frozen) labels of BFA applications where we erased the function symbol $\overline{\ }$. We then order these multi-set using the multi-set ordering $\succ_{\mathrm{mult}}$, which is induced by the product ordering $\succ_\times$, which itself is built upon an arbitrary total rewrite ordering on ground terms without boxes $\succ$ (e.g a LPO for some arbitrary precedence over function symbols).

Example 5.13. Assume that $b_1 \equiv \mathsf{if}\ b\ \mathsf{then}\ a\ \mathsf{else}\ c$ and $b_2 \equiv \mathsf{if}\ b'\ \mathsf{then}\ a'\ \mathsf{else}\ c'$. Let P_1 be the derivation:

$$\dfrac{\dfrac{\dfrac{b, a, c, u_1, v_1 \sim b', a', c', u_2, v_2}{\overline{b}, a, c, u_1, v_1 \sim \overline{b'}, a', c', u_2, v_2}\ \mathsf{UnF}}{\widetilde{b_1}, u_1, v_1 \sim \widetilde{b_2}, u_2, v_2}\ \mathsf{BFA}(\overline{b}, \overline{b'})}{\mathsf{if}\ b_1\ \mathsf{then}\ u_1\ \mathsf{else}\ v_1 \sim \mathsf{if}\ b_2\ \mathsf{then}\ u_2\ \mathsf{else}\ v_2}\ \mathsf{BFA}(b_1, b_2)$$

And P_2 be the derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{b, a, c, u_1, v_1 \sim b', a', c', u_2, v_2}{\widetilde{b}, \widetilde{a}, \widetilde{c}, u_1, v_1 \sim \widetilde{b'}, \widetilde{a'}, \widetilde{c'}, u_2, v_2}\,\mathsf{UnF}}{\widetilde{b}, \widetilde{a}, u_1, v_1, \widetilde{c}, u_1, v_1 \sim \widetilde{b'}, \widetilde{a'}, u_2, v_2, \widetilde{c'}, u_2, v_2}\,\mathsf{Dup}}{\widetilde{b}, \widetilde{a}, u_1, v_1, \mathsf{if}\ c\ \mathsf{then}\ u_1\ \mathsf{else}\ v_1 \sim \widetilde{b'}, \widetilde{a'}, u_2, v_2, \mathsf{if}\ c'\ \mathsf{then}\ u_2\ \mathsf{else}\ v_2}\,\mathsf{BFA}(c, c')}{\widetilde{b}, \mathsf{if}\ a\ \mathsf{then}\ u_1\ \mathsf{else}\ v_1, \mathsf{if}\ c\ \mathsf{then}\ u_1\ \mathsf{else}\ v_1 \sim \widetilde{b'}, \mathsf{if}\ a'\ \mathsf{then}\ u_2\ \mathsf{else}\ v_2, \mathsf{if}\ c'\ \mathsf{then}\ u_2\ \mathsf{else}\ v_2}\,\mathsf{BFA}(a, a')}{\mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ u_1\ \mathsf{else}\ v_1)\ \mathsf{else}\ (\mathsf{if}\ c\ \mathsf{then}\ u_1\ \mathsf{else}\ v_1) \sim \mathsf{if}\ b'\ \mathsf{then}\ (\mathsf{if}\ a'\ \mathsf{then}\ u_2\ \mathsf{else}\ v_2)\ \mathsf{else}\ (\mathsf{if}\ c'\ \mathsf{then}\ u_2\ \mathsf{else}\ v_2)}\,\mathsf{BFA}(b, b')$$
$$\dfrac{\sim\ \mathsf{if}\ b'\ \mathsf{then}\ (\mathsf{if}\ a'\ \mathsf{then}\ u_2\ \mathsf{else}\ v_2)\ \mathsf{else}\ (\mathsf{if}\ c'\ \mathsf{then}\ u_2\ \mathsf{else}\ v_2)}{\mathsf{if}\ b_1\ \mathsf{then}\ u_1\ \mathsf{else}\ v_1 \sim \mathsf{if}\ b_2\ \mathsf{then}\ u_2\ \mathsf{else}\ v_2}\,R$$

Observe that P_1 and P_2 are two different derivations of the same formula. P_1 and P_2 are respectively interpreted as the multi-sets:

$$\{(b_1, b_2), (b, b')\} \qquad \text{and} \qquad \{(b, b'), (a, a'), (c, c')\}$$

Remark that when interpreting the derivation as multi-sets, we unfroze the conditionals. The conditionals b, a, c (resp. b', a', c') are strict subterms of b_1 (resp. b_2), therefore we have $(b_1, b_2) \succ_\times (b, b')$, $(b_1, b_2) \succ_\times (a, a')$ and $(b_1, b_2) \succ_\times (c, c')$. Hence:

$$\{(b_1, b_2), (b, b')\} \succ_{\text{mult}} \{(b, b'), (a, a'), (c, c')\}$$

By consequence, P_2 is a smaller proof of $\mathsf{if}\ b_1\ \mathsf{then}\ u_1\ \mathsf{else}\ v_1 \sim \mathsf{if}\ b_2\ \mathsf{then}\ u_2\ \mathsf{else}\ v_2$ than P_1. □

Proof of Lemma 5.8. First we are going to show a cut elimination strategy to get rid of the deconstruction of frozen conditionals introduced by:

$$\dfrac{\vec{w}_1, \widetilde{b_1}, u_1', v_1' \sim \vec{w}_2, \widetilde{b_2}, u_2', v_2'}{\vec{w}_1, \mathsf{if}\ b_1\ \mathsf{then}\ u_1\ \mathsf{else}\ v_1 \sim \vec{w}_2, \mathsf{if}\ b_2\ \mathsf{then}\ u_2\ \mathsf{else}\ v_2}\,\mathsf{BFA}(b_1, b_2)$$

Assume now that $u \sim v$ is not provable without deconstructing frozen conditionals introduced as described above. We consider a proof P_1 of $u \sim v$ that we suppose minimal for $\succ_{\text{mult}}$. We consider the first conditionals (b_1, b_2) (starting from the bottom) which are deconstructed. We let $b_1 \equiv \mathsf{if}\ b\ \mathsf{then}\ a\ \mathsf{else}\ c$ and $b_2 \equiv \mathsf{if}\ b'\ \mathsf{then}\ a'\ \mathsf{else}\ c'$, we know that our proof has the following shape:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots\ (A_3)}{\vec{x}, \overline{b}, a, c, \vec{y} \sim \vec{x}', \overline{b'}, a', c', \vec{y}'}\,\mathsf{BFA}(\overline{b}, \overline{b'})}{\vec{x}, \widetilde{b_1}, \vec{y} \sim \vec{x}', \widetilde{b_2}, \vec{y}'}\ \vdots\ (A_2)}{\vec{w}_1, \widetilde{b_1}, u_1, v_1 \sim \vec{w}_2, \widetilde{b_2}, u_2, v_2}\,\mathsf{BFA}(b_1, b_2)}{\vec{w}_1, \mathsf{if}\ b_1\ \mathsf{then}\ u_1\ \mathsf{else}\ v_1 \sim \vec{w}_2, \mathsf{if}\ b_2\ \mathsf{then}\ u_2\ \mathsf{else}\ v_2}\ \vdots\ (A_1)}{C[\mathsf{if}\ b_1\ \mathsf{then}\ u_1\ \mathsf{else}\ v_1] \sim C[\mathsf{if}\ b_2\ \mathsf{then}\ u_2\ \mathsf{else}\ v_2]}\,R$$

Where C is a one-hole context. Since (b_1, b_2) are the first conditionals deconstructed in this proof we know that C is such that the hole does not appear in a conditional branch. This proof can be rewritten

as the following proof P_2:

$$
\begin{array}{cl}
\vdots\ (A_3) & \\
\vec{x},\widetilde{b},\widetilde{a},\widetilde{c},\vec{y} \sim \vec{x}',\widetilde{b'},\widetilde{a'},\widetilde{c'},\vec{y}' & \\
\vdots\ (A_2) & \\
\vec{w}_1,\widetilde{b},\widetilde{a},\widetilde{c},u_1,v_1 \sim \vec{w}_2,\widetilde{b'},\widetilde{a'},\widetilde{c'},u_2,v_2 & \\
\cline{1-1}
\vec{w}_1,\widetilde{b},\widetilde{a},u_1,v_1,\widetilde{c},u_1,v_1 \sim \vec{w}_2,\widetilde{b'},\widetilde{a'},u_2,v_2,\widetilde{c'},u_2,v_2 & \mathsf{Dup} \\
\cline{1-1}
\vec{w}_1,\widetilde{b},\widetilde{a},u_1,v_1,\mathsf{if}\ c\ \mathsf{then}\ u_1\ \mathsf{else}\ v_1 \sim \vec{w}_2,\widetilde{b'},\widetilde{a'},u_2,v_2,\mathsf{if}\ c'\ \mathsf{then}\ u_2\ \mathsf{else}\ v_2 & \mathsf{BFA}(c,c') \\
\cline{1-1}
\vec{w}_1,\widetilde{b},\mathsf{if}\ a\ \mathsf{then}\ u_1\ \mathsf{else}\ v_1,\mathsf{if}\ c\ \mathsf{then}\ u_1\ \mathsf{else}\ v_1 \sim \vec{w}_2,\widetilde{b'},\mathsf{if}\ a'\ \mathsf{then}\ u_2\ \mathsf{else}\ v_2,\mathsf{if}\ c'\ \mathsf{then}\ u_2\ \mathsf{else}\ v_2 & \mathsf{BFA}(a,a') \\
\cline{1-1}
\vec{w}_1,\mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ u_1\ \mathsf{else}\ v_1)\ \mathsf{else}\ (\mathsf{if}\ c\ \mathsf{then}\ u_1\ \mathsf{else}\ v_1) & \mathsf{BFA}(b,b') \\
\sim\ \vec{w}_2,\mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ a'\ \mathsf{then}\ u_2\ \mathsf{else}\ v_2)\ \mathsf{else}\ (\mathsf{if}\ c'\ \mathsf{then}\ u_2\ \mathsf{else}\ v_2) & \\
\vdots\ (A_1) & \\
C[\mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ u_1\ \mathsf{else}\ v_1)\ \mathsf{else}\ (\mathsf{if}\ c\ \mathsf{then}\ u_1\ \mathsf{else}\ v_1)] & \\
\sim\ C[\mathsf{if}\ b'\ \mathsf{then}\ (\mathsf{if}\ a'\ \mathsf{then}\ u_2\ \mathsf{else}\ v_2)\ \mathsf{else}\ (\mathsf{if}\ c'\ \mathsf{then}\ u_2\ \mathsf{else}\ v_2)] & \\
\cline{1-1}
C[\mathsf{if}\ b_1\ \mathsf{then}\ u_1\ \mathsf{else}\ v_1] \sim C[\mathsf{if}\ b_2\ \mathsf{then}\ u_2\ \mathsf{else}\ v_2] & R \\
\cline{1-1}
u \sim v & R
\end{array}
$$

One can check that A_1 remains the same in the second proof tree since the hole in C is not in a conditional branch. The A_1, A_2, A_3 parts are the same in both proofs, so let M be the interpretation of A_1, A_2, A_3 as a multi-set. Then the interpretation of P_1 and P_2 are, respectively, the multi-sets:

$$M \cup \{(b_1, b_2), (b, b')\} \qquad \text{and} \qquad M \cup \{(b, b'), (a, a'), (c, c')\}$$

Therefore P_2 is a strictly smaller proof of $u \sim v$ than P_1 (this is almost the same multi-sets than in Example 5.13). Absurd. ∎

5.7 Shape of the Terms

Most of the completeness results shown before are for any set of unitary axioms closed under Restr. We now specialize these results to CCA$_2$, to get some further restrictions.

When applying the unitary axioms CCA$_2$, we would like to require that terms are in R-normal form, e.g. to avoid the application of CCA$_2$ to terms with an unbounded component, such as $\pi_1(\langle u\,,\,v\rangle)$. Unfortunately, the side-conditions of CCA$_2$ are not stable by R. E.g., consider the CCA$_2$ instance:

$$\frac{}{\{\mathsf{if}\ \mathsf{eq}(g(\mathsf{n}_u),\mathsf{n}_u)\ \mathsf{then}\ A\ \mathsf{else}\ B\}^{\mathsf{n}_r}_{\mathsf{pk(n)}} \sim \{C\}^{\mathsf{n}_r}_{\mathsf{pk(n)}}}\ \text{CCA}_2$$

The R-normal form of the left term is:

$$\mathsf{if}\ \mathsf{eq}(g(\mathsf{n}_u),\mathsf{n}_u)\ \mathsf{then}\ \{A\}^{\mathsf{n}_r}_{\mathsf{pk(n)}}\ \mathsf{else}\ \{B\}^{\mathsf{n}_r}_{\mathsf{pk(n)}}$$

which cannot be used in a valid CCA$_2$ instance, since the conditional $\mathsf{eq}(g(\mathsf{n}_u),\mathsf{n}_u)$ should be somehow "hidden" by the encryption. To avoid this difficulty, we use a different normal form for terms: we try to be as close as possible to the R-normal form, while keeping conditional branching below their encryption. This normalization strategy preserves the shape of the terms required by the CCA$_2$ axiom, as well as its side-conditions. In other word, if $\vec{u} \sim \vec{v}$ is a valid CCA$_2$ instance then its normalization $\vec{u}_\mathsf{n} \sim \vec{v}_\mathsf{n}$ is also a valid CCA$_2$ instance. We illustrate this on an example. The term:

$$\left\{\mathsf{if}\ (\mathsf{if}\ b\ \mathsf{then}\ a\ \mathsf{else}\ c)\ \mathsf{then}\ \{\mathsf{if}\ d\ \mathsf{then}\ u\ \mathsf{else}\ v\}^{\mathsf{n}_1}_{\mathsf{pk}}\ \mathsf{else}\ w\right\}^{\mathsf{n}_2}_{\mathsf{pk}}$$

is normalized as follows:

$$\left\{\begin{array}{l}\mathsf{if}\ b\ \mathsf{then}\ \mathsf{if}\ a\ \mathsf{then}\ \{\mathsf{if}\ d\ \mathsf{then}\ u\ \mathsf{else}\ v\}^{\mathsf{n}_1}_{\mathsf{pk}}\ \mathsf{else}\ w \\ \qquad \mathsf{else}\ \mathsf{if}\ c\ \mathsf{then}\ \{\mathsf{if}\ d\ \mathsf{then}\ u\ \mathsf{else}\ v\}^{\mathsf{n}_1}_{\mathsf{pk}}\ \mathsf{else}\ w\end{array}\right\}^{\mathsf{n}_2}_{\mathsf{pk}} \tag{5.6}$$

Observe that CCA$_2$ side-conditions are preserved. For example, the condition on occurrences of encryption randomness in (5.6) holds: e.g. the randomness n_1 is only used for the encryption $\{\mathsf{if}\ d\ \mathsf{then}\ u\ \mathsf{else}\ v\}^{\mathsf{n}_1}_{\mathsf{pk}}$.

5.7.1 Definitions

We omit the rewriting strategy for now. Instead, we describe the final shape of the terms, and prove some of their properties terms. We let $\mathcal{A}_\succ$ be the ordered strategy from Lemma 5.8, and we define several restriction of $\mathcal{A}_\succ$:

$$\mathfrak{F}\big((2\mathsf{Box} + R_\square)^* \cdot \mathsf{CS}_\square^* \cdot \{\overline{\mathsf{BFA}}(b,b')\}^* \cdot \mathsf{UnF} \cdot \mathsf{FA_s}^* \cdot \mathsf{Dup}^* \cdot \text{CCA}_2\big) \qquad (\mathcal{A}_\succ)$$

$$\mathfrak{F}\big(\mathsf{CS}_\square^* \cdot \{\overline{\mathsf{BFA}}(b,b')\}^* \cdot \mathsf{UnF} \cdot \mathsf{FA_s}^* \cdot \mathsf{Dup}^* \cdot \text{CCA}_2\big) \qquad (\mathcal{A}_{\mathsf{CS}_\square})$$

$$\mathfrak{F}\big(\{\overline{\mathsf{BFA}}(b,b')\}^* \cdot \mathsf{UnF} \cdot \mathsf{FA_s}^* \cdot \mathsf{Dup}^* \cdot \text{CCA}_2\big) \qquad (\mathcal{A}_{\overline{\mathsf{BFA}}})$$

$$\mathfrak{F}\big(\mathsf{FA_s}^* \cdot \mathsf{Dup}^* \cdot \text{CCA}_2\big) \qquad (\mathcal{A}_{\mathsf{FA_s}})$$

The rule $\mathsf{CS}_\square$ is the only branching rule, therefore, after applying all the $\mathsf{CS}_\square$ rules, we can associate to each branch l of the proof a left CCA$_2$ trace $\mathcal{S}_l = (\mathcal{K}_l, \mathcal{R}_l, \mathcal{E}_l, \mathcal{D}_l)$ of the CCA$_2$ axiom, where $\mathcal{K}_l$, $\mathcal{R}_l$, $\mathcal{E}_l$ and $\mathcal{D}_l$ are the sets of, respectively, secret keys, encryption randomness, encryptions and decryptions on the left side. Similarly we have a right CCA$_2$ trace $\mathcal{S}'_l = (\mathcal{K}'_l, \mathcal{R}'_l, \mathcal{E}'_l, \mathcal{D}'_l)$.

Definition 5.21. A CCA$_2$ trace $\mathcal{S}$ is a tuple $(\mathcal{K}, \mathcal{R}, \mathcal{E}, \mathcal{D})$ where:

- $\mathcal{K} \subseteq \{\mathsf{sk}(\mathsf{n}) \mid \mathsf{n} \in \mathcal{N}\}$ is a set of secret keys.
- $\mathcal{R} \subseteq \mathcal{N}$ is a set of encryption randomness.
- $\mathcal{E} \subseteq \{\{m\}^{\mathsf{n_e}}_{\mathsf{pk(n)}} \mid \mathsf{n_e} \in \mathcal{R} \wedge \mathsf{sk}(\mathsf{n}) \in \mathcal{K}\}$ is a set of encryptions.
- $\mathcal{D} \subseteq \{\mathsf{dec}(m, \mathsf{sk}(\mathsf{n})) \mid \mathsf{sk}(\mathsf{n}) \in \mathcal{K}\}$ is a set of decryptions.

Given a CCA$_2$ instance $\phi \sim \psi$ and its corresponding CCA$_2^a$ application:

$$(\_, \mathcal{X}_{\mathsf{enc}}, \mathcal{X}_{\mathsf{dec}}, \sigma_{\mathsf{rand}}, \theta_{\mathsf{enc}}, \lambda_{\mathsf{dec}}) R^{\mathcal{K}}_{\text{CCA}_2^a} (\_, \mathcal{X}_{\mathsf{enc}}, \mathcal{X}_{\mathsf{dec}}, \sigma'_{\mathsf{rand}}, \theta'_{\mathsf{enc}}, \lambda'_{\mathsf{dec}})$$

we define the left CCA$_2$ trace $\mathcal{S} = \mathsf{l\text{-}trace}(\phi \sim \psi)$ by:

$$\mathcal{S} = (\mathcal{K}, \mathcal{X}_{\mathsf{enc}}\sigma_{\mathsf{rand}}, \mathcal{X}_{\mathsf{enc}}\theta_{\mathsf{enc}}, \mathcal{X}_{\mathsf{dec}}\lambda_{\mathsf{dec}})$$

We define similarly its right CCA$_2$ trace $\mathcal{S}' = \mathsf{r\text{-}trace}(\phi \sim \psi)$.

Let $\phi \sim \psi$ be a CCA$_2$ instance and $\mathcal{S} = \mathsf{l\text{-}trace}(\phi \sim \psi)$ be its left CCA$_2$ trace. We use $\mathcal{S}$ to define the normal form of the terms appearing, on the left, in branch using the CCA$_2$ instance $\phi \sim \psi$. This is done through four mutually inductive definitions:

- *$\mathcal{S}$-encryption oracle calls* are well-formed encryptions.
- *$\mathcal{S}$-decryption oracle calls* are well-formed decryptions.
- *$\mathcal{S}$-normalized basic terms* are terms built using function symbols in $\mathcal{F}_{\setminus \mathsf{if},\mathbf{0}}$ and well-formed encryptions and decryptions.
- *$\mathcal{S}$-normalized simple terms* are combinations of normalized basic terms using $\mathsf{if\_then\_else\_}$.

Later, we prove that all intermediate terms in proofs can be assumed to be in these normal forms. To keep the proof tractable, this will be done in two steps. Therefore we introduce two versions of some forms. E.g., we define *$\mathcal{S}$-simple terms* to be terms having a particular form, and *$\mathcal{S}$-normalized simple terms* to be *$\mathcal{S}$-simple terms* satisfying some further properties.

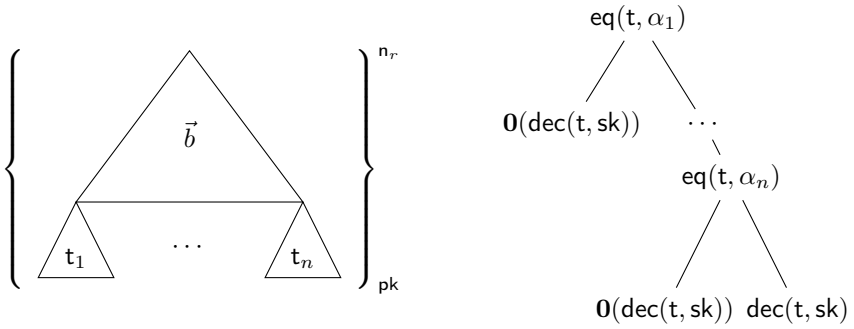
A public/private key pair is valid if the same name has been used to generate the keys.

Definition 5.22. A valid public/private key pair is a pair of terms $(\mathsf{pk}(\mathsf{n}), \mathsf{sk}(\mathsf{n}))$ where n is a name.

An *$\mathcal{S}$-encryption oracle call* is a valid encryption in $\mathcal{E}$ of the form $\{u\}^{\mathsf{n_e}}_{\mathsf{pk}}$, where $\mathsf{n_e}$ is a valid encryption randomness in $\mathcal{R}$, pk is a valid public/private key pair appearing in $\mathcal{K}$ and the encrypted plain-text u is, inductively, a *$\mathcal{S}$-normalized simple term.*

Definition 5.23. A *$\mathcal{S}$-encryption oracle call* is a term of the form $\{u\}^{\mathsf{n_e}}_{\mathsf{pk}}$ where:

- $\{u\}^{\mathsf{n_e}}_{\mathsf{pk}} \in \mathcal{E}$, $\mathsf{n_e} \in \mathcal{R}$, $(\mathsf{pk}, \mathsf{sk})$ is a valid public/private key pair and with $\mathsf{sk} \in \mathcal{K}$.
- u is a *$\mathcal{S}$-normalized simple terms.*



Encryption Oracle Call **Decryption Oracle Call**

Convention: $\alpha_1, \ldots, \alpha_n$ are the encryptions of $\mathcal{E}$ under pk appearing directly in t.

Figure 5.8: Shapes of Encryption and Decryption Oracle Calls

Similarly, a $\mathcal{S}$-*decryption oracle calls* t is valid decryption in $\mathcal{D}$ under secret key $\mathsf{sk} \in \mathcal{K}$ such that all other encryptions and decryptions appearing directly in t, either in guards or in the decrypted term, are themselves $\mathcal{S}$-*encryption oracle calls* and $\mathcal{S}$-*decryption oracle calls*.

Definition 5.24. A $\mathcal{S}$-*decryption oracle call* is a term of the form $C\,[\vec{g} \diamond (s_i)_{i \leq p}]$ in $\mathcal{D}$ where:

- $(\mathsf{pk}, \mathsf{sk})$ is valid public/private key pair and $\mathsf{sk} \in \mathcal{K}$.
- There exists a context u if-free and in R-normal form, and a term t such that:

$$t \equiv u[(\alpha_j)_j, (\mathsf{dec}_k)_k] \qquad \forall i < p,\ s_i \equiv \mathbf{0}(\mathsf{dec}(t, \mathsf{sk})) \qquad s_p \equiv \mathsf{dec}(t, \mathsf{sk}) \qquad \forall g \in \vec{g},\ g \equiv \mathsf{eq}(t, \alpha_j)$$

- For all j, α_j is a $\mathcal{S}$-*encryption oracle call*.
- For all k, dec_k is a $\mathcal{S}$-*decryption oracle call*.

$(\alpha_j)_j$ are called u's encryptions. We often write $(\mathsf{dec}_k)_k$ to denote a vector of decryption oracle calls.

Figure 5.8 gives a visual representation of the shapes of encryption and decryption oracle calls.

A $\mathcal{S}$-*basic term* is a term build using $\mathcal{S}$-*encryption oracle calls*, $\mathcal{S}$-*decryption oracle calls*, function symbols in $\mathcal{F}_{\setminus \mathsf{if}, \mathbf{0}}$ and names in $\mathcal{N}$, with some restrictions. More precisely, we require that:

- We do not use names in $\mathcal{R}$, as this would contradict CCA$_2$ randomness side-conditions.
- We do not decrypt terms using secret keys in $\mathcal{K}$.

Definition 5.25. A $\mathcal{S}$-*basic term* is a term of the form $U[\vec{w}, (\alpha_j)_j, (\mathsf{dec}_k)_k]$ where:

- U and $\vec{w}$ are if-free, U does not contain $\mathbf{0}(\_)$, $\mathsf{fresh}(\mathcal{R}; \vec{w})$ and $\mathsf{nodec}(\mathcal{K}, \vec{w})$.
- $(\alpha_j)_j$ are $\mathcal{S}$-*encryption oracle calls*.
- $(\mathsf{dec}_k)_k$ are $\mathcal{S}$-*decryption oracle calls*.

A $\mathcal{S}$-*basic conditional* is a $\mathcal{S}$-*basic term* of sort bool.

A $\mathcal{S}$-*normalized basic term* is a a $\mathcal{S}$-*basic term* that has been built without introducing any R-redex.

Definition 5.26. A $\mathcal{S}$-*normalized basic term* is a $\mathcal{S}$-*basic term* of the form $U[\vec{w}, (\alpha_j)_j, (\mathsf{dec}_k)_k]$ where:

- $(\alpha_j)_j$ are encryptions under $(\mathsf{pk}_j, \mathsf{sk}_j)_j$, and $(\mathsf{dec}_k)_k$ are decryptions under $(\mathsf{pk}_k, \mathsf{sk}_k)_k$.
- $U[\vec{w}, (\{[]_j\}^0_{\mathsf{pk}_j})_j, (\mathsf{dec}([]_k, \mathsf{sk}_k))_k]$ is in R-normal form.

A $\mathcal{S}$-*normalized basic conditional* is a $\mathcal{S}$-*normalized basic term* of sort bool.

Finally, a $\mathcal{S}$-*simple term* is a term build using only $\mathcal{S}$-*basic term* and the $\mathsf{if\_then\_else\_}$ function symbols. Moreover, if we use only $\mathcal{S}$-*normalized basic term*, then we get an a $\mathcal{S}$-*normalized simple term*.

Definition 5.27. A $\mathcal{S}$-*simple term* (resp. $\mathcal{S}$-*normalized simple term*) is a term of the form $C[\vec{b} \diamond \vec{u}]$ where:

- C is an if-context.
- $\vec{b}$ are $\mathcal{S}$-*basic conditionals* (resp. $\mathcal{S}$-*normalized basic conditionals*).
- $\vec{u}$ are $\mathcal{S}$-*basic terms* (resp. $\mathcal{S}$-*normalized basic terms*).

Remark 5.8. For all term u, the guards of a $\mathcal{S}_l$-decryption oracle calls are $\mathcal{S}_l$-normalized basic terms. But the *leaves* of $\mathcal{S}$-decryption oracle calls *are not* $\mathcal{S}$-normalized basic terms, because they do not satisfy the condition $\mathsf{nodec}(\mathcal{K}, \cdot)$. □

Orderings The inductive definition of $\mathcal{S}$-normalized basic terms naturally gives us a well-founded relation $<^{\mathcal{S}}_{\mathsf{ind}}$ between $\mathcal{S}$-normalized basic terms, $\mathcal{S}$-normalized simple terms, $\mathcal{S}$-decryption oracle calls and $\mathcal{S}$-encryption oracle calls.

Definition 5.28. $<^{\mathcal{S}}_{\mathsf{ind}}$ is the reflexive and transitive closure of the relation $<^{\mathcal{S}}$ defined as:

- For all $\mathcal{S}$-encryption oracle call $t \equiv \{u\}^r_{\mathsf{pk}}$, $u <^{\mathcal{S}} t$.
- For all $\mathcal{S}$-decryption oracle call:

$$t \equiv C\left[\vec{g} \diamond (s_i[(\alpha_j)_j, (\mathsf{dec}_k)_k])_{i \leq p}\right]$$

  for all j, $\alpha_j <^{\mathcal{S}} t$ and for all k, $\mathsf{dec}_k <^{\mathcal{S}} t$.
- For all $\mathcal{S}$-normalized basic term $t \equiv U[\vec{w}, (\alpha_j)_j, (\mathsf{dec}_k)_k]$, for all j, $\alpha_j <^{\mathcal{S}} t$ and for all k, $\mathsf{dec}_k <^{\mathcal{S}} t$.
- For all $\mathcal{S}$-normalized simple term $t \equiv C[\vec{b} \diamond \vec{u}]$, $\forall b \in \vec{b}, b <^{\mathcal{S}} t$ and $\forall u \in \vec{u}, u <^{\mathcal{S}} t$.

We let $\leq^{\mathcal{S}}_{\mathsf{bt}}$ be union of the restriction of $<^{\mathcal{S}}_{\mathsf{ind}}$ to the instances where the left term is a $\mathcal{S}$-normalized basic term, and the set of guards appearing in the right-term. Formally:

Definition 5.29. Let $<'^{\mathcal{S}}_{\mathsf{ind}}$ be the reflexive and transitive closure of the order $<'^{\mathcal{S}}$, which has the same definition than $<^{\mathcal{S}}$, apart for the $\mathcal{S}$-decryption oracle call:

- For all $\mathcal{S}$-decryption oracle call:

$$t \equiv C\left[\vec{g} \diamond (s_i[(\alpha_j)_j, (\mathsf{dec}_k)_k])_{i \leq p}\right]$$

  for all j, $\alpha_j <'^{\mathcal{S}} t$; for all k, $\mathsf{dec}_k <'^{\mathcal{S}} t$; and for all $b \in \vec{g}$, $b <'^{\mathcal{S}} t$.

We finally define $\leq^{\mathcal{S}}_{\mathsf{bt}}$ by requiring that for every terms u, v:

$$u \leq^{\mathcal{S}}_{\mathsf{bt}} v \quad \text{iff} \quad u <'^{\mathcal{S}}_{\mathsf{ind}} v \ \text{ and } \ u \text{ is a } \mathcal{S}\text{-normalized basic term}$$

5.7.2 Eager Reduction for $\mathcal{A}_{\mathsf{FA_s}}$

We state here a key result about the $\mathcal{A}_{\mathsf{FA_s}} = \mathfrak{F}(\mathsf{FA_s}^* \cdot \mathsf{Dup}^* \cdot \text{CCA}_2)$ fragment, which deals with the following proof cut: when trying to prove that $u \sim u'$ holds, one may rewrite u and u' into, respectively, $\pi_1(\langle u\,,\, v\rangle)$ and $\pi_1(\langle u'\,,\, v'\rangle)$, using R. The problem is that v and v' are arbitrary large terms, which makes the proof space unbounded. E.g. this is the case in the following proof:

$$\dfrac{\dfrac{\begin{array}{c}\vdots\ (P)\\ u, v \sim u', v'\end{array}}{\pi_1(\langle u\,,\, v\rangle) \sim \pi_1(\langle u'\,,\, v'\rangle)}\ \mathsf{FA}_{\pi_1} \cdot \mathsf{FA}_{\langle\,,\rangle}}{u \sim u'}\ R$$

Of course there is a shortcut here: P is a proof of $u, v \sim u', v'$, hence by Restr we have a proof of $u \sim u'$. Using the Restr elimination procedure (Lemma 5.1), we obtain a proof P_{cut} of $u \sim u'$ such that P_{cut} is no larger than P. By generalizing this proof cut elimination, we are going to show that if we have a proof $P \vdash_{\mathcal{A}_{\mathsf{FA_s}}} \beta \sim \beta'$ where β and β' are *basic terms*, then we can rewrite β and β' into *normalized basic terms* γ, γ' such that there exists P' no larger than P with $P' \vdash_{\mathcal{A}_{\mathsf{FA_s}}} \gamma \sim \gamma'$.

To prove this, we may have to extract several sub-proofs of P, and then recombine them into a single proof P'. While the rule $\mathsf{FA_s}$ and Dup can be easily re-combined, this is not the case for CCA$_2$. Therefore, given a finite family of CCA$_2$ instances $(\vec{u}_i \sim \vec{v}_i)_{i \in I}$, we give a sufficient condition guaranteeing that they can be recombined into a single proof $(\vec{u}_i)_{i \in I} \sim (\vec{v}_i)_{i \in I}$.

Definition 5.30. For every proof P in $\mathcal{A}_{\mathsf{FA_s}}$, we let $\mathsf{instance}(P)$ be the unique CCA_2 instance used in P.

Example 5.14. If P is the proof:

$$\dfrac{\dfrac{}{\vec{w},(\alpha_i)_{i\in I},(\mathsf{dec}_j)_{j\in J}\sim \vec{w},(\alpha'_i)_{i\in I},(\mathsf{dec}'_j)_{j\in J}}\ \text{CCA}_2}{\vdots}\ \mathsf{FA_s}^*\cdot\mathsf{Dup}^*}{C[\vec{w},(\alpha_i)_{i\in I},(\mathsf{dec}_j)_{j\in J}]\sim C[\vec{w},(\alpha'_i)_{i\in I},(\mathsf{dec}'_j)_{j\in J}]}$$

then $\mathsf{instance}(P)$ is the CCA_2 instance $\vec{w},(\alpha_i)_{i\in I},(\mathsf{dec}_j)_{j\in J}\sim \vec{w},(\alpha'_i)_{i\in I},(\mathsf{dec}'_j)_{j\in J}$. □

We say that a CCA_2 instance ϕ is a sub-instance of another CCA_2 instance ψ if the set of encryptions and decryptions of ϕ are included into, respectively, the set of encryptions and decryptions of ψ. Moreover, we require that the symmetric part of ϕ contains only sub-terms of the symmetric part of ψ.

Definition 5.31. A CCA_2 instance:

$$\vec{w}_0,(\alpha_i)_{i\in I_0},(\mathsf{dec}_j)_{j\in J_0}\sim \vec{w}_0,(\alpha'_i)_{i\in I_0},(\mathsf{dec}'_j)_{j\in J_0}$$

is a sub-instance of a CCA_2 instance:

$$\vec{w},(\alpha_i)_{i\in I},(\mathsf{dec}_j)_{j\in J}\sim \vec{w},(\alpha'_i)_{i\in I},(\mathsf{dec}'_j)_{j\in J}$$

if and only if $\mathsf{st}(\vec{w}_0)\subseteq\mathsf{st}(\vec{w})$, $I_0\subseteq I$ and $J_0\subseteq J$.

The following proposition allows to re-combine several proofs $P_1,\dots,P_n$, as long as there exists a CCA_2 instance $\vec{u}\sim\vec{v}$ such that for every i, $\mathsf{instance}(P_i)$ is a sub-instance of $\vec{u}\sim\vec{v}$.

Proposition 5.7. *Let $(\beta_n)_{n\in N}$ and $(\beta'_n)_{n\in N}$ be such that for every $n\in N$, there exists a proof $P_n\vdash_{\mathcal{A}_{\mathsf{FA_s}}}\beta_n\sim\beta'_n$. If there exists a CCA_2 instance $\vec{u}\sim\vec{v}$ such that for every n, $\mathsf{instance}(P_n)$ is a sub-instance of $\vec{u}\sim\vec{v}$, then there exists P such that:*

- $P\vdash_{\mathcal{A}_{\mathsf{FA_s}}}(\beta_n)_{n\in N}\sim(\beta'_n)_{n\in N}$
- $\mathsf{instance}(P)$ *is a sub-instance of* $\vec{u}\sim\vec{v}$.
- P *contain the same number of* $\mathsf{FA_s}$ *rules than the derivations* $P_1,\dots,P_N$ *altogether.*

Proof. Axioms $\mathsf{FA_s}$ and Dup verify a frame property. More precisely:

$$\text{if}\quad \frac{\vec{u}'\sim\vec{v}'}{\vec{u}\sim\vec{v}}\ Ax \qquad \text{then for every } \vec{w}_\mathsf{l},\vec{w}_\mathsf{r} \text{ of the same length} \qquad \frac{\vec{w}_\mathsf{l},\vec{u}'\sim\vec{w}_\mathsf{r},\vec{v}'}{\vec{w}_\mathsf{l},\vec{u}\sim\vec{w}_\mathsf{r},\vec{v}}\ Ax$$

Therefore we can easily combine all proofs $(P_n)_{n\in N}$. For every $n\in N$, we let $\mathsf{instance}(P_n)\equiv\vec{u}_n\sim\vec{u}'_n$. Moreover, we let $(\vec{v}_n)_{n\in N}\sim(\vec{v}'_n)_{n\in N}$ be the formula obtained from $(\vec{u}_n)_{n\in N}\sim(\vec{u}'_n)_{n\in N}$ by removing all duplicates, and where for every n, $\vec{v}_n\subseteq\vec{u}_n$ and $\vec{v}'_n\subseteq\vec{u}'_n$. Then we have the derivation:

$$\dfrac{\dfrac{(\vec{v}_n)_{n\in N}\sim(\vec{v}'_n)_{n\in N}}{(\vec{u}_n)_{n\in N}\sim(\vec{u}'_n)_{n\in N}}\ \mathsf{Dup}^*}{\vdots}}{(\beta_n)_{n\in N}\sim(\beta'_n)_{n\in N}}$$

Now, we want to conclude by applying the CCA_2 axiom. The problem is that CCA_2 does not verify the frame property. But using the fact that for every n, $\vec{u}_n\sim\vec{u}'_n$ is a sub-instance of $\vec{u}\sim\vec{v}$, and that $(\vec{v}_n)_{n\in N}\sim(\vec{v}'_n)_{n\in N}$ does not contain duplicates, we can check that $(\vec{v}_n)_{n\in N}\sim(\vec{v}'_n)_{n\in N}$ is a sub-instance of $\vec{u}\sim\vec{v}$. Hence we have a valid derivation in $\mathcal{A}_{\mathsf{FA_s}}$. ■

Lemma 5.9. *Let $P\vdash_{\mathcal{A}_{\mathsf{FA_s}}}\beta\sim\beta'$ and $\mathcal{S},\mathcal{S}'$ be the, respectively, left and right CCA_2 trace corresponding to $\mathsf{instance}(P)$. If β and β' are, respectively, $\mathcal{S}$-*basic term *and $\mathcal{S}'$-*basic term *then there exist $\gamma=_R\beta$ and $\gamma'=_R\beta'$ such that:*

- γ *and* γ' *are, respectively,* $\mathcal{S}$-normalized basic term *and* $\mathcal{S}'$-normalized basic term.

- *There exists P' such that $P' \vdash_{\mathcal{A}_{\mathsf{FA_s}}} \gamma \sim \gamma'$, $\mathsf{instance}(P')$ is a sub-instance of $\mathsf{instance}(P)$ and P' contains less $\mathsf{FA_s}$ rules than P.*

Proof. Let $\mathcal{S} = (\mathcal{K}, \mathcal{R}, \mathcal{E}, \mathcal{D})$. We prove the lemma by induction on the number of $\mathsf{FA_s}$ rules in P. If P has no $\mathsf{FA_s}$ application, then we have three cases:

- β and β' are identical, up to α-renaming. In that case, we can check that $\gamma \equiv \beta\downarrow_R$ and $\gamma' \equiv \beta'\downarrow_R$ satisfy the wanted properties.
- β and β' are, resp., a $\mathcal{S}$-encryption oracle call and a $\mathcal{S}'$-encryption oracle call. Since an $\mathcal{S}$-encryption oracle call is also a $\mathcal{S}$-normalized basic term, we conclude by taking $\gamma \equiv \beta$ and $\gamma' \equiv \beta'$.
- β and β' are, resp., a $\mathcal{S}$-decryption oracle call and a $\mathcal{S}'$-decryption oracle call. Similarly, a $\mathcal{S}$-decryption oracle call is also a $\mathcal{S}$-normalized basic term. We conclude by taking $\gamma \equiv \beta$ and $\gamma' \equiv \beta'$.

For the inductive case, β and β' must start with the same function symbol. Hence:

$$\beta \equiv f(\beta_1, \dots, \beta_n) \qquad\qquad \beta' \equiv f(\beta'_1, \dots, \beta'_n)$$

First, we check that $\beta_1, \dots, \beta_n$ are $\mathcal{S}$-basic terms. Indeed, the only way that some β_i could not be a $\mathcal{S}$-basic term was if β was an $\mathcal{S}$-encryption oracle call or a $\mathcal{S}$-decryption oracle call. Then, f must be $\{\_\}_\_$ or $\mathsf{dec}(\_,\_)$:

- in the former case, $\beta \equiv \{\_\}^{\mathsf{n_e}}_\_$ where $\mathsf{n_e} \in \mathcal{R}$ and one of the β_i is equal to $\mathsf{n_e}$. Since β is a $\mathcal{S}$-basic term, we know that $\mathsf{fresh}(\mathcal{R}; \mathsf{n_e})$. Contradiction.
- in the latter case, $\beta \equiv \mathsf{dec}(\_, \mathsf{sk}(\mathsf{n}))$ where $\mathsf{sk}(\mathsf{n}) \in \mathcal{K}$. Since β is a $\mathcal{S}$-basic term, we know that $\mathsf{nodec}(\mathcal{K}, \mathsf{sk}(\mathsf{n}))$. Contradiction.

Hence $\beta_1, \dots, \beta_n$ are $\mathcal{S}$-basic terms. Similarly $\beta'_1, \dots, \beta'_n$ are $\mathcal{S}'$-basic terms.

Using Lemma 5.1, we know that for every i, we can extract from P a proof of $Q_i \vdash_{\mathcal{A}_{\mathsf{FA_s}}} \beta_i \sim \beta'_i$. One can check that the procedure described in Lemma 5.1 is such that P has as many $\mathsf{FA}_{\setminus \mathbf{0}}$ applications than all the $(Q_i)_i$ altogether. By induction hypothesis, let:

$$P_1 \vdash_{\mathcal{A}_{\mathsf{FA_s}}} \gamma_1 \sim \gamma'_1, \quad \dots \quad , P_n \vdash_{\mathcal{A}_{\mathsf{FA_s}}} \gamma_n \sim \gamma'_n$$

be such that for every i, $\gamma_i =_R \beta_i$, $\gamma'_i =_R \beta'$, γ_i is a $\mathcal{S}$-normalized basic term and γ'_i is a $\mathcal{S}'$-normalized basic term, $\mathsf{instance}(P_i)$ is a sub-instance of $\mathsf{instance}(P)$ and P_i has less $\mathsf{FA_s}$ applications than Q_i. By Proposition 5.7, there exists a proof P' of:

$$P' \vdash_{\mathcal{A}_{\mathsf{FA_s}}} (\gamma_n)_{n\in N} \sim (\gamma'_n)_{n\in N}$$

such that $\mathsf{instance}(P')$ is a sub-instance of $\mathsf{instance}(P)$ and P' has as many $\mathsf{FA_s}$ applications than the $(P_i)_i$ altogether. Since P_i has less $\mathsf{FA_s}$ applications than Q_i, and since P has as many $\mathsf{FA_s}$ applications than all the $(Q_i)_i$ altogether, P' has less $\mathsf{FA_s}$ applications than P.

$f(\beta_1, \dots, \beta_n)$ and $f(\beta'_1, \dots, \beta'_n)$ can only have R_1 redexes at the top-level. If they have no R_1 redexes, then $f(\beta_1, \dots, \beta_n)$ and $f(\beta'_1, \dots, \beta'_n)$ are, respectively, $\mathcal{S}$-normalized basic term and $\mathcal{S}'$-normalized basic term. We conclude by applying FA_f:

$$\dfrac{\begin{array}{c}\vdots\ (P')\\ \gamma_1, \dots, \gamma_n \sim \gamma'_1, \dots, \gamma'_n\end{array}}{f(\gamma_1, \dots, \gamma_n) \sim f(\gamma'_1, \dots, \gamma'_n)}\ \mathsf{FA}_f$$

Therefore, assume $f(\beta_1, \dots, \beta_n)$ or $f(\beta'_1, \dots, \beta'_n)$ have a R_1 redex. We have several cases:

- Both left and right sides can be reduced by $\pi_i(\langle x_1\,,\, x_2\rangle) \to x_i$. W.l.o.g. we assume $i = 1$:

$$\dfrac{\langle \gamma_1\,,\, \gamma_2\rangle \sim \langle \gamma'_1\,,\, \gamma'_2\rangle}{\pi_1(\langle \gamma_1\,,\, \gamma_2\rangle) \sim \pi_1(\langle \gamma'_1\,,\, \gamma'_2\rangle)}\ \mathsf{FA}_{\pi_1}$$

  We look at the next rule in P':

  - If it is $\textsc{cca}_2$, then $\langle \gamma_1\,,\, \gamma_2\rangle$ and $\langle \gamma'_1\,,\, \gamma'_2\rangle$ are the same terms, up to α-renaming. We conclude by taking $\gamma \equiv \gamma_1$ and $\gamma' \equiv \gamma'_1$.

– Or it is a function application:

$$\dfrac{\dfrac{\vdots\ (Q)}{\gamma_1,\gamma_2 \sim \gamma_1',\gamma_2'}\ \mathsf{FA}_{\langle\,,\,\rangle}}{\pi_1(\langle\gamma_1\,,\,\gamma_2\rangle) \sim \pi_1(\langle\gamma_1'\,,\,\gamma_2'\rangle)}\ \mathsf{FA}_{\pi_1}$$

Using Lemma 5.1, we extract from Q a proof $Q' \vdash_{\mathcal{A}_{\mathbf{FA_s}}} \gamma_1 \sim \gamma_1'$ no larger than Q. We conclude by taking $\gamma \equiv \gamma_1$ and $\gamma' \equiv \gamma_1'$:

$$\dfrac{\dfrac{\vdots\ (Q')}{\gamma_1 \sim \gamma_1'}}{\pi_1(\langle\gamma_1\,,\,\gamma_2\rangle) \sim \pi_1(\langle\gamma_1'\,,\,\gamma_2'\rangle)}\ R \tag{5.7}$$

- Only one side can be reduced by $\pi_i(\langle x_1\,,\,x_2\rangle) \to x_i$. Therefore the next rule applied in (P') must be CCA_2 (since the head function symbols differ). But in a CCA_2 application, we cannot have $\langle \_\,,\,\_\rangle \sim f'(\_)$ with $f' \neq \langle\,,\,\rangle$. Contradiction.
- Both sides can be reduced by $\mathsf{dec}(\{x\}^r_{\mathsf{pk(n)}}, \mathsf{sk(n)}) \to x$. Hence $n = 2$, $\gamma_1,\gamma_2 \equiv \{u\}^r_{\mathsf{pk(n)}}, \mathsf{sk(n)}$, $\gamma_1',\gamma_2' \equiv \{u'\}^{r'}_{\mathsf{pk(n')}}, \mathsf{sk(n')}$ and P' is of the form:

$$\dfrac{\{u\}^r_{\mathsf{pk(n)}}, \mathsf{sk(n)} \sim \{u'\}^{r'}_{\mathsf{pk(n')}}, \mathsf{sk(n')}}{\mathsf{dec}(\{u\}^r_{\mathsf{pk(n)}}, \mathsf{sk(n)}) \sim \mathsf{dec}(\{u'\}^{r'}_{\mathsf{pk(n')}}, \mathsf{sk(n')})}\ \mathsf{FA}_{\mathsf{dec}}$$

We look at the next rule applied on $\{u\}^r_{\mathsf{pk}(n)}, \_ \sim \{u'\}^{r'}_{\mathsf{pk}(n')}, \_$. If it is a function application then we have a shortcut using Lemma 5.1, as we did for (5.7). If it is CCA_2, we have two cases:

– $\{u\}^r_{\mathsf{pk}(n)}$ and $\{u'\}^{r'}_{\mathsf{pk}(n')}$ are the same terms, up to α-renaming. We conclude by taking $\gamma \equiv u$ and $\gamma' \equiv u'$.

– $\{u\}^r_{\mathsf{pk}(n)}$ and $\{u'\}^{r'}_{\mathsf{pk}(n')}$ are, respectively, a $\mathcal{S}$-encryption oracle call and a $\mathcal{S}'$-encryption oracle call. Then $\mathsf{sk(n)} \in \mathcal{K}$. Since $\gamma_2 \equiv \mathsf{sk(n)}$ and γ_2 is a $\mathcal{S}$-normalized basic term, we know that $\mathsf{nodec}(\mathcal{K}, \mathsf{sk(n)})$. Contradiction.

- Only one side can be reduced by $\mathsf{dec}(\{x\}^r_{\mathsf{pk(n)}}, \mathsf{sk(n)}) \to x$. Then (P') is necessarily of the form:

$$\dfrac{\{t\}^r_{\mathsf{pk(n)}}, \mathsf{sk(n)} \sim \{t'\}^{r'}_{p'}, \mathsf{sk}'(\mathsf{n}')}{\mathsf{dec}(\{t\}^r_{\mathsf{pk(n)}}, \mathsf{sk(n)}) \sim \mathsf{dec}(\{t'\}^{r'}_{p'}, \mathsf{sk}'(\mathsf{n}'))}\ \mathsf{FA}_{\mathsf{dec}}$$

We look at the next rule applied to $\{t\}^r_{\mathsf{pk(n)}}$ and $\{t'\}^{r'}_{p'}$:

– If it is CCA_2, then $p' \equiv \mathsf{pk}(\mathsf{n}')$. Therefore the right side can be reduced by $\mathsf{dec}(\{x\}^r_{\mathsf{pk(n')}}, \mathsf{sk(n')}) \to x$. Contradiction.

– If it is $\mathsf{FA}_{\{\_\}_{\_}}$ then there is a proof of $\_\mathsf{pk(n)}, \mathsf{sk(n)} \sim \_, p', \mathsf{sk(n')}$, which implies that $p' \equiv \mathsf{pk}(\mathsf{n}')$. Therefore the right side can be reduced by $\mathsf{dec}(\{x\}^r_{\mathsf{pk(n')}}, \mathsf{sk(n')}) \to x$. Contradiction.

- Both side can be reduced by $\mathsf{eq}(x,x) \to \mathsf{true}$. In this case the proof cut elimination is trivial.
- Only one side can be reduced by $\mathsf{eq}(x,x) \to \mathsf{true}$. Therefore we have a proof of the form:

$$\dfrac{t, t \sim t', t''}{\mathsf{eq}(t,t) \sim \mathsf{eq}(t',t'')}\ \mathsf{FA}_{\mathsf{eq}(,)}$$

Using Lemma 5.2 we know that $t' \equiv t''$, therefore both side can be reduced by $\mathsf{eq}(x,x) \to \mathsf{true}$. Contradiction. ■

5.8 Proof Form

5.8.1 Early Proof Form

We showed in Lemma 5.8 that:

$$\mathfrak{F}\left((2\mathsf{Box} + R_{\Box})^* \cdot \mathsf{CS}_{\Box}^* \cdot \{\overline{\mathsf{BFA}}(b, b')\}^* \cdot \mathsf{UnF} \cdot \mathsf{FA_s}^* \cdot \mathsf{Dup}^* \cdot \textsc{cca}_2\right) \qquad (\mathcal{A}_{\succ})$$

is complete for $\mathfrak{F}((\mathsf{CS}_{\mathsf{if}}^{\mathsf{no}} + \mathsf{FA}_{\setminus \mathbf{0}} + R + \mathsf{Dup} + \textsc{cca}_2)^*)$. Let us consider a proof P following this ordering. The only branching rule in $\mathcal{A}_{\succ}$ is the $\mathsf{CS}_{\Box}$ rule, which has two premises. Hence after having completed all the $\mathsf{CS}_{\Box}$ applications we know that the proof will be non-branching and in $\mathcal{A}_{\overline{\mathsf{BFA}}}$. We want to name each branch of the proof tree, and its corresponding instance of the $\textsc{cca}_2$ axiom. To do so, we index each branch of the proof tree P by some $l \in L$ where L is a finite set of labels,

Definition 5.32. We let $\vdash^{\mathsf{b}}$ be the proof system $\vdash$ with branch annotations. When $P \vdash^{\mathsf{b}} t \sim t'$, we let $\mathsf{label}(P)$ be the set of labels annotating the branches in P, and for all $l \in \mathsf{label}(P)$, we let $\mathsf{instance}(P, l)$ be the $\textsc{cca}_2$ instance used in branch l.

When applying the $\mathsf{CS}_{\Box}$ rule on two boxed conditionals $\boxed{b_1}\boxed{b_2}_b$ and $\boxed{b'_1}\boxed{b'_2}_{b'}$, we know that the sub-proofs of $b_1 \sim b'_1$ and $b_2 \sim b'_2$ lie in the fragment $\mathcal{A}_{\mathsf{CS}_{\Box}}$. This gives us useful information on the shape of the terms. To use this, we define the $\mathsf{extract_l}$ and $\mathsf{extract_r}$ functions which allow to retrieve the left and right sub-proofs of, respectively, $b_1 \sim b'_1$ and $b_2 \sim b'_2$.

Definition 5.33. Given a proof $P \vdash \vec{u} \sim \vec{v}$ and a position h in the proof P such that:

$$P_{|h} = \dfrac{\vec{w}, b_1, (u_i)_i \sim \vec{w}', b'_1, (u'_i)_i \qquad \vec{w}, b_2, (v_i)_i \sim \vec{w}', b'_2, (v'_i)_i}{\vec{w}, \left(\mathsf{if}\ \boxed{b_1}\boxed{b_2}_b\ \mathsf{then}\ u_i\ \mathsf{else}\ v_i\right)_i \sim \vec{w}', \left(\mathsf{if}\ \boxed{b'_1}\boxed{b'_2}_{b'}\ \mathsf{then}\ u'_i\ \mathsf{else}\ v'_i\right)_i}\ \mathsf{CS}_{\Box}$$

We let $\mathsf{extract_l}(h, P)$ be proof of $b_1 \sim b'_1$ extracted from $P_{|h}$, and $\mathsf{extract_r}(h, P)$ be proof of $b_2 \sim b'_2$ extracted from $P_{|h}$, using the Restr elimination procedure described in the proof of Lemma 5.7.

Using this, we define what are proofs in *early proof form*.

Definition 5.34. For all terms t, t' and proofs P such that $P \vdash^{\mathsf{b}}_{\mathcal{A}_{\mathsf{cs}_{\Box}}} t \sim t'$, we say that P proof in *early proof form* if t and t' are of the following form:

$$t \equiv C\left[\left(\boxed{b^{h_{\mathsf{l}}}}\boxed{b^{h_{\mathsf{r}}}}_{b^h}\right)_{h \in H} \diamond (u_l)_{l \in \mathsf{label}(P)}\right] \qquad \wedge \qquad t' \equiv C\left[\left(\boxed{b'^{h_{\mathsf{l}}}}\boxed{b'^{h_{\mathsf{r}}}}_{b'^h}\right)_{h \in H} \diamond (u'_l)_{l \in \mathsf{label}(P)}\right]$$

where H is a set of positions in P such that:

- for all $h \in H$, the rule applied at position h in P is a $\mathsf{CS}_{\Box}$ rule on the conditionals:

$$\left(\boxed{b^{h_{\mathsf{l}}}}\boxed{b^{h_{\mathsf{r}}}}_{b^h}, \boxed{b'^{h_{\mathsf{l}}}}\boxed{b'^{h_{\mathsf{r}}}}_{b'^h}\right)$$

- Let $P^{h_{\mathsf{l}}} = \mathsf{extract_l}(h, P)$ and $P^{h_{\mathsf{r}}} = \mathsf{extract_r}(h, P)$, then:

$$P^{h_{\mathsf{l}}} \vdash^{\mathsf{b}}_{\mathcal{A}_{\mathsf{cs}_{\Box}}} b^{h_{\mathsf{l}}} \sim b'^{h_{\mathsf{l}}} \qquad \text{and} \qquad P^{h_{\mathsf{r}}} \vdash^{\mathsf{b}}_{\mathcal{A}_{\mathsf{cs}_{\Box}}} b^{h_{\mathsf{r}}} \sim b'^{h_{\mathsf{r}}}$$

  and these two proofs are in *early proof form*.

- $\mathsf{label}(P^{h_{\mathsf{l}}}) \subseteq \mathsf{label}(P)$, and for all $l \in \mathsf{label}(P^{h_{\mathsf{l}}})$, $\mathsf{instance}(P^{h_{\mathsf{l}}}, l)$ is a sub-instance of $\mathsf{instance}(P, l)$ (same for $\mathsf{label}(P^{h_{\mathsf{r}}})$).
- For all $l \in \mathsf{label}(P)$, the proof of $u_l \sim u'_l$ extracted from P is in the fragment $\mathcal{A}_{\overline{\mathsf{BFA}}}$.

Moreover, we let $\mathsf{cs\text{-}pos}(P) \equiv H$.

Proposition 5.8. *For all terms t, t' and proofs P such that $P \vdash_{\mathcal{A}_{\mathsf{cs}_{\Box}}} t \sim t'$, there exists a labelling P' of P such that $P' \vdash^{\mathsf{b}}_{\mathcal{A}_{\mathsf{cs}_{\Box}}} t \sim t'$ and P' is in early proof form.*

Proof. We can check that the proof P has the wanted shape and is properly labelled by induction on the size of the proof, by observing that for all $h \in \mathsf{cs\text{-}pos}(P)$ and $\mathsf{x} \in \{\mathsf{l}, \mathsf{r}\}$, $\mathsf{extract}_{\mathsf{x}}(h, P)$ is of size strictly smaller that P. We only need to do some α-renaming to have the labelling of the sub-proofs coincide.

Finally we can check that the resulting proof Q is such that for all $h \in \mathsf{cs\text{-}pos}(Q), \mathsf{x} \in \{\mathsf{l}, \mathsf{r}\}$, for all $l \in \mathsf{label}(\mathsf{extract}_{\mathsf{x}}(h, P))$, the CCA_2 instance $\mathsf{instance}(\mathsf{extract}_{\mathsf{x}}(h, P), l)$ is a sub-instance of $\mathsf{instance}(P, l)$. This follows from the fact that $\mathsf{extract}_{\mathsf{x}}(h, P)$ is obtained through the Restr elimination procedure from P. ■

We define below the set $\mathsf{index}(P)$ of *all* positions of P where a $\mathsf{CS}_\Box$ rule is applied. This includes the set of positions $\mathsf{cs\text{-}pos}(P)$, as well as the $\mathsf{CS}_\Box$ applications in sub-proofs of conditionals $b \sim b'$. This set is naturally ordered using the prefix ordering on positions.

Definition 5.35. Let $P \vdash^{\mathsf{b}}_{\mathcal{A}_{\mathsf{cs}_\Box}} t \sim t'$ in early proof form.

- We let $\mathsf{index}(P)$ be the set of indices where $\mathsf{CS}_\Box$ rules occur in the proof P:

$$\mathsf{index}(P) = \mathsf{cs\text{-}pos}(P) \cup \bigcup_{h \in \mathsf{cs\text{-}pos}(P)} \mathsf{index}\left(\mathsf{extract}_{\mathsf{l}}(h, P)\right) \cup \mathsf{index}\left(\mathsf{extract}_{\mathsf{r}}(h, P)\right)$$

- For all $h, h' \in \mathsf{index}(t, P)$, we let $<$ be the ancestor relation on positions, defined by $h < h'$ if and only if h is a strict prefix of h'.
- For all $\mathsf{h} = h_{\mathsf{x}}$, where $h \in \mathsf{index}(P)$ and $\mathsf{x} \in \{\mathsf{l}, \mathsf{r}\}$, we let $\mathsf{cs\text{-}pos}_P(\mathsf{h}) = \mathsf{cs\text{-}pos}(\mathsf{extract}_{\mathsf{x}}(h, P))$. When there is no ambiguity on the proof P, we write $\mathsf{cs\text{-}pos}(\mathsf{h})$ instead of $\mathsf{cs\text{-}pos}_P(\mathsf{h})$.

We define the set $\mathsf{h\text{-}branch}(l)$ of positions of P where a $\mathsf{CS}_\Box$ rule is applied on the branch l. Of course, for all $l \in \mathsf{label}(P)$, $\epsilon \in \mathsf{h\text{-}branch}(l)$ since ϵ is the index of the toplevel proof P.

Definition 5.36. Let $P \vdash^{\mathsf{b}}_{\mathcal{A}_{\mathsf{cs}_\Box}} t \sim t'$ in early proof form. For all $l \in \mathsf{label}(P)$, we define:

$$\mathsf{h\text{-}branch}_P(l) = \{h_{\mathsf{x}} \mid h \in \mathsf{index}(P) \wedge \mathsf{x} \in \{\mathsf{l}, \mathsf{r}\} \wedge l \in \mathsf{label}(\mathsf{extract}_{\mathsf{x}}(h, P))\} \cup \{\epsilon\}$$

We abuse the notation and say that $h \in \mathsf{h\text{-}branch}_P(l)$ if there exists $\mathsf{x} \in \{\mathsf{l}, \mathsf{r}\}$ such that $h_{\mathsf{x}} \in \mathsf{h\text{-}branch}_P(l)$. In that case, we say that x is the direction taken at h in l.

We omit the proof P when there is no ambiguity, writing $\mathsf{h\text{-}branch}(l)$ instead of $\mathsf{h\text{-}branch}_P(l)$.

5.8.2 Shape of the Terms

For all proofs in $\mathcal{A}_{\succ}$, all R rewritings are done at the beginning of the proofs in the $(\mathsf{2Box} + R_\Box)^*$ part, and, afterwards, all rules (apart from Dup) only "peel off" terms by removing the top-most function symbol. Therefore the terms just after $(\mathsf{2Box} + R_\Box)^*$ characterize the shape of the subsequent proof. This observation is illustrated in Figure 5.9. Recall that for all $P \vdash^{\mathsf{b}}_{\mathcal{A}_{\mathsf{cs}_\Box}} t \sim t'$ in early proof form, we have:

$$t \equiv C\Big[\Big(\boxed{b^{h_{\mathsf{l}}} \mid b^{h_{\mathsf{r}}}}_{b^h}\Big)_{h \in H} \diamond (u_l)_{l \in \mathsf{label}(P)}\Big] \qquad \text{and} \qquad t' \equiv C\Big[\Big(\boxed{b'^{h_{\mathsf{l}}} \mid b'^{h_{\mathsf{r}}}}_{b'^h}\Big)_{h \in H} \diamond (u'_l)_{l \in \mathsf{label}(P)}\Big]$$

where for all $l \in \mathsf{label}(P)$, the extraction from P of the sub-proof of $u_l \sim u'_l$ is in the fragment $\mathcal{A}_{\overline{\mathsf{BFA}}}$. Therefore, for every l, u_l and u'_l are of the form:

$$u_l \equiv D_l\big[(\beta_{i,l})_{i \in I_l} \diamond (\gamma_{m,l})_{m \in M_l}\big] \qquad\qquad u'_l \equiv D_l\big[(\beta'_{i,l})_{i \in I_l} \diamond (\gamma'_{m,l})_{m \in M_l}\big]$$

where D_l is an if-context and:

- $(\beta_{i,l})_{i \in I_l}$ and $(\beta'_{i,l})_{i \in I_l}$ are conditionals such that the sub-proofs $(\beta_{i,l} \sim \beta'_{i,l})_{i \in I_l}$ extracted from P are in $\mathcal{A}_{\mathsf{FA_s}}$.
- $(\gamma_{j,l})_{j \in M_l}$ and $(\gamma'_{j,l})_{j \in M_l}$ are terms such that the sub-proofs $(\gamma_{j,l} \sim \gamma'_{j,l})_{j \in M_l}$ extracted from P are in $\mathcal{A}_{\mathsf{FA_s}}$.

Using these notation, we give some definitions:

Definition 5.37. Let $P \vdash^{\mathsf{b}}_{\mathcal{A}_{\mathsf{cs}_\Box}} t \sim t'$ in early proof form. For every $l \in \mathsf{label}(P)$, we let:

- $(b, b') \leq^{\epsilon, l}_{\mathsf{cs} \sim \mathsf{cs}} (t \sim t', P)$ if and only if there exists $h_0 \in \mathsf{cs\text{-}pos}(P)$ such that $b \equiv b^{h_0}$ and $b' \equiv b'^{h_0}$.

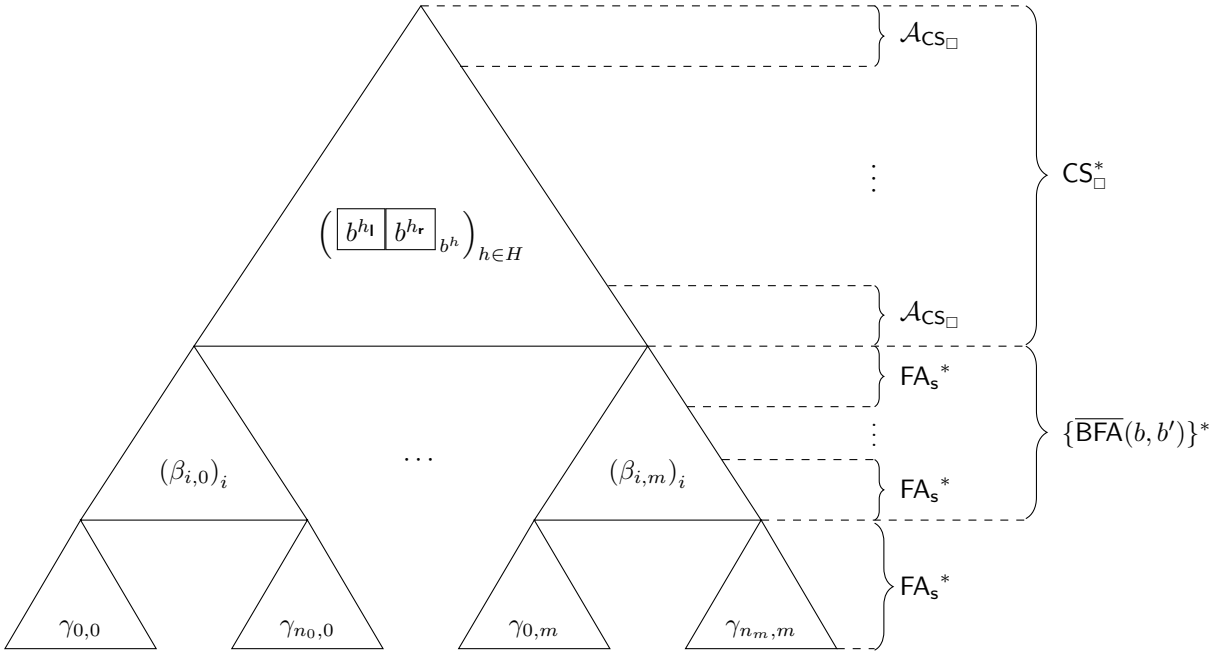


Figure 5.9: The shape of the term is determined by the proof.

- $(\beta,\beta') \leq^{\epsilon,l}_{\mathsf{c}\sim\mathsf{c}} (t\sim t', P)$ if and only if there exists $i\in I_l$ such that $\beta\equiv\beta_{i,l}$ and $\beta'\equiv\beta'_{i,l}$
- $(\gamma,\gamma') \leq^{\epsilon,l}_{\mathsf{l}\sim\mathsf{l}} (t\sim t', P)$ if and only if there exists $m\in M_l$ such that $\gamma\equiv\gamma_{m,l}$ and $\gamma'\equiv\gamma'_{m,l}$.

Remark 5.9. Let $P\vdash^{\mathsf{b}}_{\mathcal{A}_{\mathsf{cs}_\square}} t\sim t'$ in early proof form and $L=\mathsf{label}(P)$. Then:

$$\begin{aligned} t &\equiv C\Big[\_\diamond\big(D_l\big[(\beta)_{\beta\leq^{\epsilon,l}_{\mathsf{c}}(t,P)} \quad \diamond(\gamma)_{\gamma\leq^{\epsilon,l}_{\mathsf{l}}(t,P)} \quad\big]\big)_{l\in L}\Big]\\ \text{and}\quad t' &\equiv C\Big[\_\diamond\big(D_l\big[(\beta')_{\beta'\leq^{\epsilon,l}_{\mathsf{c}}(t',P)}\diamond(\gamma')_{\gamma'\leq^{\epsilon,l}_{\mathsf{l}}(t',P)}\big]\big)_{l\in L}\Big]\end{aligned}$$ □

These relations allow use to obtain all *pairs* of terms appearing at the root level in P. We naturally define the asymmetric relation $\leq_{\mathsf{x}}$ from $\leq_{\mathsf{x}\sim\mathsf{x}}$:

Definition 5.38. Let $P\vdash^{\mathsf{b}}_{\mathcal{A}_{\mathsf{cs}_\square}} t\sim t'$ in early proof form. For every $l\in\mathsf{label}(P)$ and $\mathsf{x}\in\{\mathsf{c},\mathsf{l},\mathsf{cs}\}$, we let:

$$\forall s.\, s\leq^{\epsilon,l}_{\mathsf{x}} (t,P) \qquad \text{if and only if} \qquad (s,\_)\leq^{\epsilon,l}_{\mathsf{x}\sim\mathsf{x}} (t\sim t',P)$$

Let $h\in\mathsf{index}(P)$ and $\mathsf{x}\in\{\mathsf{l},\mathsf{r}\}$. We lift these relations to h_x using the proof $\mathsf{extract}_\mathsf{x}(h,P)$.

Definition 5.39. Let $P\vdash^{\mathsf{b}}_{\mathcal{A}_{\mathsf{cs}_\square}} t\sim t'$ in early proof form. Let $l\in\mathsf{label}(P)$, $h\in\mathsf{index}(P)$, $\mathsf{x}\in\{\mathsf{l},\mathsf{r}\}$ and b,b' be such that $\mathsf{extract}_\mathsf{x}(h,P)$ is a proof of $b\sim b'$. Then:

- For any $\Delta\in\{\mathsf{c}\sim\mathsf{c},\mathsf{l}\sim\mathsf{l},\mathsf{cs}\sim\mathsf{cs}\}$:

$$\forall s,s'.\,(s,s')\leq^{h_\mathsf{x},l}_{\Delta} (t\sim t',P) \qquad \text{if and only if} \qquad (s,s')\leq^{\epsilon,l}_{\Delta} (b\sim b',\mathsf{extract}_\mathsf{x}(h,P))$$

- For any $\Delta\in\{\mathsf{c},\mathsf{l},\mathsf{cs}\}$:

$$\forall s.\, s\leq^{h_\mathsf{x},l}_{\Delta} (t,P) \qquad \text{if and only if} \qquad s\leq^{\epsilon,l}_{\Delta} (b,\mathsf{extract}_\mathsf{x}(h,P))$$

Remark 5.10. We extend these notations to proofs P such that $P\vdash^{\mathsf{b}}_{\mathcal{A}_\succ} t\sim t'$. Let P' be such that:

$$P\equiv \frac{P'}{t\sim t'}\ (2\mathsf{Box}+R_\square)^*$$

and $P'\vdash^{\mathsf{b}}_{\mathcal{A}_{\mathsf{cs}_\square}} t_0\sim t_0'$, then $(s,s')\leq^{\mathsf{h},l}_{\Delta}(t\sim t',P)$ if and only if $(s,s')\leq^{\mathsf{h},l}_{\Delta}(t_0\sim t_0',P')$ for any $\Delta\in\{\mathsf{c}\sim\mathsf{c},\mathsf{l}\sim\mathsf{l},\mathsf{cs}\sim\mathsf{cs}\}$. We have a similar definition for $\Delta\in\{\mathsf{c},\mathsf{l},\mathsf{cs}\}$. □

5.8.3 Proof Form and Normalized Proof Form

Definition 5.40. Let $P \vdash^{\mathsf{b}}_{\mathcal{A}^{\mathsf{cs}}_{\square}} t \sim t'$ in early proof form and $L = \mathsf{label}(P)$. Let $\mathcal{S}_l$ be the left trace of the CCA_2 instance used in branch l, and $\mathcal{S}'_l$ be the right trace of $\mathsf{instance}(P, l)$:

$$\mathcal{S}_l^P = \mathsf{l\text{-}trace}(\mathsf{instance}(P, l)) \qquad \mathcal{S}_l'^P = \mathsf{r\text{-}trace}(\mathsf{instance}(P, l))$$

We say that P is in *proof form* if and only if, for every $l \in L$:

- for every $h \in \mathsf{cs\text{-}pos}(P)$ and $x \in \{\mathsf{l}, \mathsf{r}\}$, the proof $\mathsf{extract}_{\mathsf{x}}(h, P)$ is in *proof forms*.
- $(\beta, \beta') \leq^{\epsilon,l}_{\mathsf{c}\sim\mathsf{c}} (t \sim t', P)$, β is a $\mathcal{S}$-*basic term* and β' is a $\mathcal{S}'$-*basic term*.
- $(\gamma, \gamma') \leq^{\epsilon,l}_{\mathsf{l}\sim\mathsf{l}} (t \sim t', P)$, γ is a $\mathcal{S}$-*basic term* and γ' is a $\mathcal{S}'$-*basic term*.

We obtain the definition of *normalized proof form* by replacing, in the definition above, *basic term* by *normalized basic term*, and *proof form* by *normalized proof form*.

We write $P \vdash^{\mathsf{npf}} t \sim t'$ whenever P is a proof of $t \sim t'$ in normalized proof form.

Let $P \vdash^{\mathsf{npf}} t \sim t'$, we already defined the set of conditionals $\leq^{\mathsf{h},l}_{\mathsf{c}} (t, P)$ used in the $\overline{\mathsf{BFA}}$ rules in the sub-proof P of at index h and branch l. In the case of proof in normalized proof form, these conditionals are normalized basic conditional. Similarly the set of leaf terms $\leq^{\mathsf{h},l}_{\mathsf{l}} (t, P)$ in the sub-proof of P of at index h and branch l is a set of normalized basic terms. Recall that a basic term may contain other basic terms in its subterm. Hence we can define the set of all normalized basic terms appearing in the subterms of $\leq^{\mathsf{h},l}_{\mathsf{c}} (t, P) \cup \leq^{\mathsf{h},l}_{\mathsf{l}} (t, P)$.

Definition 5.41. For every $P \vdash^{\mathsf{npf}} t \sim t'$, for every term s, $s \leq^{\mathsf{h},l}_{\mathsf{bt}} (t, P)$ if and only if there exists $u (\leq^{\mathsf{h},l}_{\mathsf{c}} \cup \leq^{\mathsf{h},l}_{\mathsf{l}})(t, P)$ such that $s \leq^{\mathcal{S}_l}_{\mathsf{bt}} u$.

5.8.4 Restriction to Proofs in Normalized Proof Form

Definition 5.42. We let $\overline{\mathrm{CCA}_2}$ be the restriction of CCA_2 to cases $\vec{w}, (\alpha_i)_i, (\mathsf{dec}_j)_j \sim \vec{w}', (\alpha'_i)_i, (\mathsf{dec}'_j)_j$ where:

- $(\alpha_j)_j, (\alpha'_j)_j$ are encryption oracle calls.
- $(\mathsf{dec}_j)_j, (\mathsf{dec}'_j)_j$ are decryption oracle calls.

Lemma 5.10. *The following strategy is complete for* $\mathfrak{F}((\mathit{CS}^{\mathit{no}}_{\mathit{if}} + \mathit{FA}_{\setminus \mathit{0}} + R + \mathit{Dup} + \mathrm{CCA}_2)^*)$*:*

$$\mathfrak{F}((\mathit{2Box} + R_{\square})^* \cdot \mathit{CS}^*_{\square} \cdot \{\overline{\mathit{BFA}}(b, b')\}^* \cdot \mathit{UnF} \cdot \mathit{FA_s}^* \cdot \mathit{Dup}^* \cdot \overline{\mathrm{CCA}_2})$$

Proof. By Lemma 5.8, the following strategy is complete for $\mathfrak{F}(\mathsf{CS}^{\mathsf{no}}_{\mathsf{if}} + \mathsf{FA}_{\setminus \mathbf{0}} + R + \mathsf{Dup} + \mathrm{CCA}_2)$:

$$\mathfrak{F}((\mathsf{2Box} + R_{\square})^* \cdot \mathsf{CS}^*_{\square} \cdot \{\overline{\mathsf{BFA}}(b, b')\}^* \cdot \mathsf{UnF} \cdot \mathsf{FA_s}^* \cdot \mathsf{Dup}^* \cdot \mathrm{CCA}_2) \qquad (\mathcal{A}_{\succ})$$

For every proof $P \vdash^{\mathsf{b}} t \sim t'$ in this fragment, we let $L^P = \mathsf{label}(P)$ the set of branch indices of P. Moreover, we let $\mathcal{S}^P_l = (\mathcal{K}^P_l, \mathcal{R}^P_l, \mathcal{E}^P_l, \mathcal{D}^P_l)$ be the left trace of the CCA_2 instance of branch l, i.e. $\mathcal{S}^P_l = \mathsf{l\text{-}trace}(\mathsf{instance}(P, l))$. Finally, we define the order $<^l_P$ as follows: for all $u, u' \in \mathcal{E}^P_l \cup \mathcal{D}^P_l$, we let $u <^l_P u'$ hold if u is a strict subterm of u'.

We are going to show that for every proof P of $t \sim t'$ in $\mathcal{A}_{\succ}$, there exists a proof Q of $t \sim t'$ such that for every $l \in \mathsf{label}(Q)$, $\mathcal{E}^Q_l$ and $\mathcal{D}^Q_l$ are sets of, respectively, $\mathcal{S}^Q_l$-encryption oracle calls and $\mathcal{S}^Q_l$-decryption oracle calls, and the right part of Q and P are the same. We prove this by induction on the number of elements of $\bigcup_l \mathcal{E}^P_l \cup \mathcal{D}^P_l$ that are not $\mathcal{S}^P_l$-encryption oracle calls or $\mathcal{S}^P_l$-decryption oracle calls.

Let P be a proof of $t \sim t'$, $l \in L^P$ and let u minimal for $<^l_P$ which is not a $\mathcal{S}^P_l$-encryption oracle call or a $\mathcal{S}^P_l$-decryption oracle call. We have two cases:

- If $u \in \mathcal{E}^P_l$ is an encryption. We know that $u \equiv \{m\}^{\mathsf{n}_r}_{\mathsf{pk}}$ where the corresponding secret key sk is in $\mathcal{K}^P_l$. Let $(\alpha_k)_k$ be $\mathcal{E}^P_l \cap \mathsf{st}(m)$, and $(\mathsf{dec}_n)_n$ be $\mathcal{D}^P_l \cap \mathsf{st}(m)$. Let C be the smallest context such that:

$$m \equiv C[(\alpha_k)_k, (\mathsf{dec}_n)_n]$$

From the definition of CCA_2, we know that $C[]$ does not contain the $\mathbf{0}(\_)$ function symbol. We let A be an if-context and $(B_i[])_i$, $(U_m[])_m$ be if-free contexts in R-normal form such that $C[] =_R A[(B_i[])_i \diamond (U_m[])_m]$. Let m_0 be the term:

$$m_0 \equiv A[(B_i[(\alpha_k)_k, (\mathsf{dec}_n)_n])_i \diamond (U_m[(\alpha_k)_k, (\mathsf{dec}_n)_n])_m]$$

We know that $m_0 =_R m$. We are going to show that m_0 is a $\mathcal{S}_l^P$-simple term. Since $C[]$ does not contain the $\mathbf{0}(\_)$ function symbol, we know that the contexts $(B_i[])_i$ and $(U_m[])_m$ do not contain $\mathbf{0}(\_)$. By minimality of u, we know that the $(\alpha_k)_k$ are $\mathcal{S}_l^P$-encryption oracle calls, and the $(\mathsf{dec}_n)_n$ are $\mathcal{S}_l^P$-decryption oracle calls. For every k, α_k is of the shape $\alpha_k \equiv \{\_\}^{\mathsf{n}_k}_{\mathsf{pk}_k}$. For every n, we let sk_n be the secret key used in dec_n. Assume that there is some i such that:

$$\tilde{m} \equiv B_i[(\{[]_k\}^{\mathsf{n}_k}_{\mathsf{pk}_k})_k, (\mathsf{dec}([]_n, \mathsf{sk}_n))_n]$$

is not in R-normal form. Since $B_i[]$ is in R-normal form, we can only have a redex at one of the encryption. More precisely, there must exist some k such that $\mathsf{dec}(\{[]_k\}^{\mathsf{n}_k}_{\mathsf{pk}_k}, \mathsf{sk}_k)$ is a subterm of $\tilde{m}$. By consequence, sk_k is a subterm of $B_i[]$. But since $\mathsf{sk}_k \in \mathcal{K}_l^P$, we know that $st(B_i)$ does not contain sk_k (sk_k can only appear in $\mathcal{D}_l^P$). Contradiction. Hence $\tilde{m}$ is in R-normal form, which implies that $(B_i[(\alpha_k)_k, (\mathsf{dec}_n)_n])_i$ are $\mathcal{S}_l^P$-normalized basic terms. Similarly we prove that $(U_m[(\alpha_k)_k, (\mathsf{dec}_n)_n])_m$ are $\mathcal{S}_l^P$-normalized basic terms. Hence m_0 is a $\mathcal{S}_l^P$-normalized simple term.

We then rewrite, using R, every occurrence of $\{m\}^{\mathsf{n}_r}_{\mathsf{sk}}$ by $\{m_0\}^{\mathsf{n}_r}_{\mathsf{sk}}$ in branch l of P. We check that this yields a valid proof Q. The only difficulty lies in making sure that the side-conditions of the CCA_2 application for the decryptions still holds. Their is one subtlety here: an encryption $\alpha \equiv \{m_\alpha\}^{\mathsf{n}_\alpha}_{\mathsf{pk}}$ must be guarded in some $\mathsf{dec}(u_0, \mathsf{sk})$ iff it appears directly in u_0. This side-condition is preserved as it is stable by any R rewriting (hence in particular the rewriting of $\{m\}^{\mathsf{n}_r}_{\mathsf{sk}}$ into $\{m_0\}^{\mathsf{n}_r}_{\mathsf{sk}}$).

We can check that the resulting proof Q of $t \sim t'$ has a smaller number of terms in $\mathcal{E}_l^Q \cup \mathcal{D}_l^Q$ which are not $\mathcal{S}_l^Q$-encryption oracle calls or $\mathcal{S}_l^Q$-decryption oracle calls. Since all other branches $l' \in L_P \backslash \{l\}$ are left unchanged, and since the right part of the proof (corresponding to t') is also left unchanged we can conclude using the induction hypothesis.

- One can check that the case where $u \equiv C[(g_e)_e \diamond (s_a)_{a \leq p}] \in \mathcal{D}_l^P$ is a decryption cannot happen. ■

We are now ready to prove that $\vdash^{\mathsf{npf}}$ is complete.

Lemma 5.11. *The restriction of the fragment $\mathcal{A}_\succ$ to formulas provable in $\vdash^{npf}$ is complete for:*

$$\mathfrak{F}((CS_{if}^{no} + FA_{\backslash \mathbf{0}} + R + Dup + \text{CCA}_2)^*)$$

Proof. Using Lemma 5.10, the following strategy is complete for $\mathfrak{F}((\mathsf{CS}_{\mathsf{if}}^{\mathsf{no}} + \mathsf{FA}_{\backslash \mathbf{0}} + R + \mathsf{Dup} + \overline{\text{CCA}_2})^*)$:

$$\mathfrak{F}((2\mathsf{Box} + R_\Box)^* \cdot \mathsf{CS}_\Box^* \cdot \{\overline{\mathsf{BFA}}(b, b')\}^* \cdot \mathsf{UnF} \cdot {\mathsf{FA}_\mathsf{s}}^* \cdot \mathsf{Dup}^* \cdot \overline{\text{CCA}_2})$$

First we show that this strategy remains complete even if with restrict it to proofs such that the terms after $(2\mathsf{Box} + R_\Box)^*$ are in proof form. Let $\vdash_{\mathcal{A}_{\mathbf{cs}_\Box}} t \sim t'$, we want to find $t_0 =_R t, t'_0 =_R t'$ and P' such that $P' \vdash^{\mathsf{npf}} t \sim t'$.

By Proposition 5.8, we know that there exists P such that $P \vdash^{\mathsf{b}}_{\mathcal{A}_{\mathbf{cs}_\Box}} t \sim t'$. Let $h \in \mathsf{index}(P), \mathsf{x} \in \{\mathsf{l}, \mathsf{r}\}$, $\mathsf{h} = h_\mathsf{x}$, and let $b^\mathsf{h}, b'^\mathsf{h}$ be such that $\mathsf{extract}_\mathsf{x}(h, P) \vdash^{\mathsf{b}}_{\mathcal{A}_{\mathbf{cs}_\Box}} b^\mathsf{h} \sim b'^\mathsf{h}$. First, we prove that we can ensure that for every $(\beta, \beta')(\leq^{\mathsf{h},l}_{\mathsf{c}\sim\mathsf{c}} \cup \leq^{\mathsf{h},l}_{\mathsf{l}\sim\mathsf{l}})(t \sim t', P)$, the terms β and β' are, respectively, $\mathcal{S}_l^P$-basic term and $\mathcal{S}_l'^P$-basic terms. We know that:

$$\beta \equiv B[\vec{w}, (\alpha_j)_j, (\mathsf{dec}_k)_k] \qquad \beta' \equiv B[\vec{w}, (\alpha'_j)_j, (\mathsf{dec}'_k)_k]$$

where B and B' are if-free and $\vec{w}, (\alpha_j)_j, (\mathsf{dec}_k)_k \sim \vec{w}, (\alpha'_j)_j, (\mathsf{dec}'_k)_k$ is a sub-instance of $\mathsf{instance}(P, l)$.

Since this is a sub-instance, we know that $\mathsf{fresh}(\mathcal{R}_l^P; \vec{w})$ and $\mathsf{nodec}(\mathcal{K}_l^P, \vec{w})$. Moreover, using the fact that $\mathsf{instance}(P, l)$ is a $\overline{\text{CCA}_2}$ instance, we know that $(\alpha_j)_j$ and $(\mathsf{dec}_k)_k$ are, respectively, $\mathcal{S}_l^P$-encryption oracle calls and $\mathcal{S}_l^P$-decryption oracle calls. Therefore if $\vec{w}$ is if-free then β is a $\mathcal{S}_l^P$-basic term.

Assume that $\vec{w}$ is not if-free. Then there exists contexts B_e, B_c, B_0, B_1 such that:

$$B \equiv B_e[\mathsf{if}\ B_c\ \mathsf{then}\ B_0\ \mathsf{else}\ B_1]] =_R \mathsf{if}\ B_c\ \mathsf{then}\ B_e[B_0]\ \mathsf{else}\ B_e[B_1]$$

Let t_0 be the term obtained from t by replacing this occurrence of β by:

$$\mathsf{if}\ B_c[\vec{w}, (\alpha_j)_j, (\mathsf{dec}_k)_k]\ \mathsf{then}\ (B_e[B_0])[\vec{w}, (\alpha_j)_j, (\mathsf{dec}_k)_k]\ \mathsf{else}\ (B_e[B_1])[\vec{w}, (\alpha_j)_j, (\mathsf{dec}_k)_k]$$

Similarly we define t'_0 by replacing β' by the corresponding term. Then $t_0 =_R t$ and $t'_0 =_R t'$. Moreover it is easy to check that the formula $t_0 \sim t'_0$ is provable in $\vdash^{\mathsf{b}}_{\mathcal{A}_{\mathsf{cs}_\square}}$, as we replaced one $\overline{\mathsf{BFA}}$ application by three $\overline{\mathsf{BFA}}$ applications (without changing the encryptions, decryptions or branches of the proof etc ...).

We replaced B by three terms $B_c, B_e[B_0], B_e[B_1]$ containing strictly less $\mathsf{if\ then\ else}$ applications. Hence, by induction, we ensure that all such contexts B are if-free, by repeating the proof rewriting above. We deduce that there exists a proof Q of $t \sim t'$ where Q is in proof form.

To obtain a *normalized* proof form, we only have to apply the Lemma 5.9 to all branches l, and to commute the new R rewriting to the bottom of the proof. ■

5.9 Properties of Normalized Basic Terms

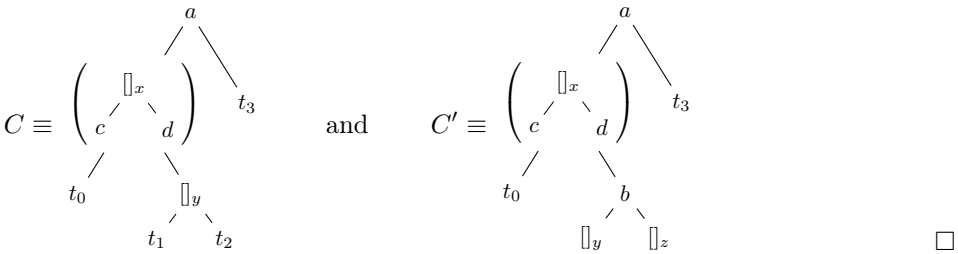
5.9.1 Basic Term Extraction

Definition 5.43. We call a *conditional context* a context $C[]_{\vec{x}}$ such that all holes appear in the conditional part of an $\mathsf{if\_then\_else\_}$. Formally, for every position p, if $C_{|p}$ is a hole $[]_x$ then $p = p'.0$ and there exist u and v such that:

$$C_{|p'} \equiv \mathsf{if}\ []_x\ \mathsf{then}\ u\ \mathsf{else}\ v$$

We say that u is an *almost conditional context* if u a conditional context or a hole.

Example 5.15. We give an example of a conditional context C with two holes on the left, and a context C' which is *not* a conditional context on the right (since it has holes in leaf positions):



□

The main goal of this subsection is to show the following lemma.

Lemma 5.12. *For all $P \vdash^{npf} t \sim t'$, for all h, l and $\beta, \beta' \leq^{h,l}_{bt} (t, P)$, there exists an almost conditional context $\tilde{\beta}'[]$ such that:*

$$\beta' \equiv \tilde{\beta}'\,[\beta] \qquad \textit{and} \qquad \mathsf{leave\text{-}st}(\beta \downarrow_R) \cap \mathsf{cond\text{-}st}\left(\tilde{\beta}'[] \downarrow_R\right) = \emptyset$$

Before delving in the proof, we would like to remark that the above lemma is not entirely satisfactory. Consider the following example:

$$\begin{aligned} \beta_0 \equiv\ & \mathsf{eq}(\{\mathsf{if}\ b\ \mathsf{then}\ s\ \mathsf{else}\ t\}^{\mathsf{n}_r}_{\mathsf{pk(n)}}, 0) \\ =_R\ & \mathsf{if}\ b\ \mathsf{then}\ \underbrace{\mathsf{eq}(\{s\}^{\mathsf{n}_r}_{\mathsf{pk(n)}}, 0)}_{\beta^0_0}\ \mathsf{else}\ \underbrace{\mathsf{eq}(\{t\}^{\mathsf{n}_r}_{\mathsf{pk(n)}}, 0)}_{\beta^1_0} \end{aligned} \qquad \beta_1 \equiv \mathsf{eq}(\{\mathsf{if}\ \beta^0_0\ \mathsf{then}\ u\ \mathsf{else}\ u\}^{\mathsf{n}'_r}_{\mathsf{pk(n)}}, 0)$$

where $\beta^0_0, \beta^1_0 \notin \mathsf{cond\text{-}st}(u \downarrow_R)$ and $s \neq_R t$. Then $\beta^0_0, \beta^1_0 \notin \mathsf{cond\text{-}st}(\beta_1 \downarrow_R)$, because β^0_0 disappear using the rule $\mathsf{if}\ x\ \mathsf{then}\ y\ \mathsf{else}\ y \to y$ in R. Hence, Lemma 5.12 could choose $\tilde{\beta}_1 \equiv \beta_1$. Of course this situation cannot occur, as we cannot have β^0_0 be a subterm of β_1 (this contradicts the freshness side-condition of

encryptions' randomnesses in the CCA_2 axiom). But we cannot rule this situation out simply by applying the lemma, we have to make a more in-depth analysis. We would like to a stronger version of this lemma that somehow directly "includes" the above observation.

To do this we introduce over-approximations $\overline{\mathsf{leave\text{-}st}}(\cdot)$ and $\overline{\mathsf{cond\text{-}st}}(\cdot)$ of, respectively, $\mathsf{leave\text{-}st}(\cdot\downarrow_R)$ and $\mathsf{cond\text{-}st}(\cdot\downarrow_R)$. Then, we show that Lemma 5.12 holds for $\overline{\mathsf{leave\text{-}st}}(\cdot)$ and $\overline{\mathsf{cond\text{-}st}}(\cdot)$.

Definition 5.44. We define the function $\overline{\mathsf{leave\text{-}st}}$ from the set of terms to the set of if-free terms in R-normal form:

$$\overline{\mathsf{leave\text{-}st}}(u_0,\dots,u_n) = \cup_{i\leq n}\overline{\mathsf{leave\text{-}st}}(u_i) \qquad \overline{\mathsf{leave\text{-}st}}(\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v) = \overline{\mathsf{leave\text{-}st}}(u,v)$$

$$\overline{\mathsf{leave\text{-}st}}(f(u_0,\dots,u_n)) = \left\{f(v_0,\dots,v_n)\downarrow_R \middle|\ \forall i\leq n, v_i\in\overline{\mathsf{leave\text{-}st}}(u_i)\right\} \quad (\forall f\in\mathcal{F}_{\setminus\mathsf{if}}\cup\mathcal{N})$$

We define the function $\overline{\mathsf{cond\text{-}st}}$ from the set of terms to the set of if-free conditionals in R-normal form:

$$\overline{\mathsf{cond\text{-}st}}(u_0,\dots,u_n) = \cup_{i\leq n}\overline{\mathsf{cond\text{-}st}}(u_i) \qquad \overline{\mathsf{cond\text{-}st}}(f(\vec{u})) = \overline{\mathsf{cond\text{-}st}}(\vec{u}) \quad (\forall f\in\mathcal{F}_{\setminus\mathsf{if}}\cup\mathcal{N})$$

$$\overline{\mathsf{cond\text{-}st}}(\mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v) = \overline{\mathsf{cond\text{-}st}}(b)\cup\overline{\mathsf{leave\text{-}st}}(b)\cup\overline{\mathsf{cond\text{-}st}}(u,v)$$

Remark 5.11. There are multiples over-approximations. For example, assuming that b,u,v,w,s,t are if-free terms in R-normal forms, there in an over-approximation in the $\mathsf{if\_then\_else\_}$ case:

$$\mathsf{leave\text{-}st}\Big(\big(\ u\ ,\ \mathsf{if}\ b\ \mathsf{then}\ v\ \mathsf{else}\ w\ \big)\downarrow_R\Big) = \{u,w\} \qquad \overline{\mathsf{leave\text{-}st}}\big(\ u\ ,\ \mathsf{if}\ b\ \mathsf{then}\ v\ \mathsf{else}\ w\ \big) = \{u,v,w\}$$

There in another over-approximation in the f case:

$$\mathsf{leave\text{-}st}\Big(f\big(\ u\ ,\ \mathsf{if}\ b\ \mathsf{then}\ v\ \mathsf{else}\ s\ ,\ \mathsf{if}\ b\ \mathsf{then}\ t\ \mathsf{else}\ s\ \big)\downarrow_R\Big) = \{f(u,s),f(v,t)\}$$

$$\overline{\mathsf{leave\text{-}st}}\Big(f\big(\ u\ ,\ \mathsf{if}\ b\ \mathsf{then}\ v\ \mathsf{else}\ u\ ,\ \mathsf{if}\ b\ \mathsf{then}\ s\ \mathsf{else}\ t\ \big)\Big) = \{f(u,s),f(u,t),f(v,s),f(v,t)\}$$

$\overline{\mathsf{cond\text{-}st}}()$ inherits from $\overline{\mathsf{leave\text{-}st}}()$ over-approximations, and also over-approximates in the $\mathsf{if\ \ then\ \ else}$ case. E.g., while $\mathsf{cond\text{-}st}(t\downarrow_R)$ never contains conditionals which are spurious in t, the set $\overline{\mathsf{cond\text{-}st}}(t)$ may:

$$\mathsf{cond\text{-}st}\big(\ u\ ,\ \mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ u\ \downarrow_R\big) = \emptyset \qquad \overline{\mathsf{cond\text{-}st}}\big(\ u\ ,\ \mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ u\ \big) = \{b\}$$

□

$\overline{\mathsf{leave\text{-}st}}(\cdot)$ is a sound over-approximation of $\mathsf{leave\text{-}st}(\cdot\downarrow_R)$. Moreover, $\overline{\mathsf{leave\text{-}st}}(\cdot)$ and $\mathsf{leave\text{-}st}(\cdot\downarrow_R)$ coincides on terms in R-normal form. The same properties hold for $\overline{\mathsf{leave\text{-}st}}(\cdot)$ and $\mathsf{leave\text{-}st}(\cdot\downarrow_R)$.

Proposition 5.9. *$\overline{\mathsf{leave\text{-}st}}$ and $\overline{\mathsf{cond\text{-}st}}$ are sound over-approximations:*

- *For all $u\to_R^* u'$, $\overline{\mathsf{leave\text{-}st}}(u)\supseteq\overline{\mathsf{leave\text{-}st}}(u')$. Moreover $\overline{\mathsf{leave\text{-}st}}(u\downarrow_R)=\mathsf{leave\text{-}st}(u\downarrow_R)$.*
- *For all $u\to_R^* u'$, $\overline{\mathsf{cond\text{-}st}}(u)\supseteq\overline{\mathsf{cond\text{-}st}}(u')$. Moreover $\overline{\mathsf{cond\text{-}st}}(u\downarrow_R)=\mathsf{cond\text{-}st}(u\downarrow_R)$.*

Proof. The facts that $\overline{\mathsf{leave\text{-}st}}(u\downarrow_R)=\mathsf{leave\text{-}st}(u\downarrow_R)$ and $\overline{\mathsf{cond\text{-}st}}(u\downarrow_R)=\mathsf{cond\text{-}st}(u\downarrow_R)$ are straightforward to show. Let us prove by induction on $\to_R^*$ that for all $u\to_R^* u'$, $\overline{\mathsf{leave\text{-}st}}(u)\supseteq\overline{\mathsf{leave\text{-}st}}(u')$. If $u\equiv u'$ this is immediate, assume that $u\to_R v\to_R^* u'$. By induction hypothesis we know that $\overline{\mathsf{leave\text{-}st}}(v)\supseteq\overline{\mathsf{leave\text{-}st}}(u')$. Therefore, we only need to show that $\overline{\mathsf{leave\text{-}st}}(u)\supseteq\overline{\mathsf{leave\text{-}st}}(v)$. We do a case disjunction on the rule applied at $u\to_R v$ (we omit the redundant or obvious cases):

- $u\equiv\mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ s\ \mathsf{else}\ t)\ \mathsf{else}\ w$ and $v\equiv\mathsf{if}\ b\ \mathsf{then}\ s\ \mathsf{else}\ w$ then:

$$\begin{aligned}\overline{\mathsf{leave\text{-}st}}(u) &= \overline{\mathsf{leave\text{-}st}}(s)\cup\overline{\mathsf{leave\text{-}st}}(t)\cup\overline{\mathsf{leave\text{-}st}}(w)\\ &\supseteq \overline{\mathsf{leave\text{-}st}}(s)\cup\overline{\mathsf{leave\text{-}st}}(w)\\ &= \overline{\mathsf{leave\text{-}st}}(v)\end{aligned}$$

- $u \equiv \mathsf{if}\ b\ \mathsf{then}\ s\ \mathsf{else}\ s$ and $v \equiv s$ then:

$$\overline{\mathsf{leave\text{-}st}}(u) = \overline{\mathsf{leave\text{-}st}}(s) = \overline{\mathsf{leave\text{-}st}}(v)$$

- $u \equiv \mathsf{if}\ (\mathsf{if}\ b\ \mathsf{then}\ a\ \mathsf{else}\ c)\ \mathsf{then}\ s\ \mathsf{else}\ t$ and $v \equiv \mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ s\ \mathsf{else}\ t)\ \mathsf{else}\ (\mathsf{if}\ c\ \mathsf{then}\ s\ \mathsf{else}\ t)$:

$$\overline{\mathsf{leave\text{-}st}}(u) = \overline{\mathsf{leave\text{-}st}}(s) \cup \overline{\mathsf{leave\text{-}st}}(t) = \overline{\mathsf{leave\text{-}st}}(v)$$

- $u \equiv \mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ s\ \mathsf{else}\ t)\ \mathsf{else}\ w$ and $v \equiv \mathsf{if}\ a\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ s\ \mathsf{else}\ w)\ \mathsf{else}\ (\mathsf{if}\ b\ \mathsf{then}\ t\ \mathsf{else}\ w)$:

$$\overline{\mathsf{leave\text{-}st}}(u) = \overline{\mathsf{leave\text{-}st}}(s) \cup \overline{\mathsf{leave\text{-}st}}(t) \cup \overline{\mathsf{leave\text{-}st}}(w) = \overline{\mathsf{leave\text{-}st}}(v)$$

- $u \equiv f(\vec{w}, \mathsf{if}\ b\ \mathsf{then}\ \vec{s}\ \mathsf{else}\ \vec{t})$ and $v \equiv \mathsf{if}\ b\ \mathsf{then}\ f(\vec{w},\vec{s})\ \mathsf{else}\ f(\vec{w},\vec{t})$ then:

$$\begin{aligned}
\overline{\mathsf{leave\text{-}st}}(u) \;=\;& \{f(\vec{w}',\vec{w}'')\downarrow_R |\ \forall i, w_i' \in \overline{\mathsf{leave\text{-}st}}(w_i) \wedge \forall j, w_j'' \in \overline{\mathsf{leave\text{-}st}}(s_j) \cup \overline{\mathsf{leave\text{-}st}}(t_j)\} \\
\supseteq\;& \{f(\vec{w}',\vec{w}'')\downarrow_R |\ \forall i, w_i' \in \overline{\mathsf{leave\text{-}st}}(w_i) \wedge \forall j, w_j'' \in \overline{\mathsf{leave\text{-}st}}(s_j)\} \\
& \cup \{f(\vec{w}',\vec{w}'')\downarrow_R |\ \forall i, w_i' \in \overline{\mathsf{leave\text{-}st}}(w_i) \wedge \forall j, w_j'' \in \overline{\mathsf{leave\text{-}st}}(t_j)\} \\
\supseteq\;& \overline{\mathsf{leave\text{-}st}}(f(\vec{w},\vec{s})) \cup \overline{\mathsf{leave\text{-}st}}(f(\vec{w},\vec{t})) \\
\supseteq\;& \overline{\mathsf{leave\text{-}st}}(v)
\end{aligned}$$

- $(u \equiv \pi_i(\langle s_1\ ,\ s_2\rangle),\ v \equiv s_i)$, $(u \equiv \mathsf{dec}(\{m\}^{\mathsf{n}_r}_{\mathsf{pk(n)}}, \mathsf{sk(n)}),\ v \equiv m)$ and $(u \equiv \mathsf{eq}(x,x),\ v \equiv x)$ are trivial.

Similarly, we show by induction on $\to_R^*$ that for all $u \to_R^* u'$, $\overline{\mathsf{cond\text{-}st}}(u) \supseteq \overline{\mathsf{cond\text{-}st}}(u')$. If $u \equiv u'$ this is immediate, assume that $u \to_R v \to_R^* u'$. By induction hypothesis we know that $\overline{\mathsf{leave\text{-}st}}(v) \supseteq \overline{\mathsf{leave\text{-}st}}(u')$. Therefore, we only need to show that $\overline{\mathsf{leave\text{-}st}}(u) \supseteq \overline{\mathsf{leave\text{-}st}}(v)$. We do a case disjunction on the rule applied at $u \to_R v$ (we omit the redundant or obvious cases):

- $u \equiv \mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ s\ \mathsf{else}\ t)\ \mathsf{else}\ w$ and $v \equiv \mathsf{if}\ b\ \mathsf{then}\ s\ \mathsf{else}\ w$ then:

$$\begin{aligned}
\overline{\mathsf{cond\text{-}st}}(u) \;&= \overline{\mathsf{cond\text{-}st}}(s,t,w) \cup \overline{\mathsf{cond\text{-}st}}(b) \cup \overline{\mathsf{leave\text{-}st}}(b) \\
&\supseteq \overline{\mathsf{cond\text{-}st}}(s,w) \cup \overline{\mathsf{cond\text{-}st}}(b) \cup \overline{\mathsf{leave\text{-}st}}(b) \\
&\supseteq \overline{\mathsf{cond\text{-}st}}(v)
\end{aligned}$$

- $(u \equiv \mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ s\ \mathsf{else}\ t)\ \mathsf{else}\ w,\ v \equiv \mathsf{if}\ a\ \mathsf{then}\ (\mathsf{if}\ b\ \mathsf{then}\ s\ \mathsf{else}\ w)\ \mathsf{else}\ (\mathsf{if}\ b\ \mathsf{then}\ t\ \mathsf{else}\ w))$ and $(u \equiv \mathsf{if}\ b\ \mathsf{then}\ s\ \mathsf{else}\ s,\ v \equiv s)$ are simple.
- $u \equiv \mathsf{if}\ (\mathsf{if}\ b\ \mathsf{then}\ a\ \mathsf{else}\ c)\ \mathsf{then}\ s\ \mathsf{else}\ t$ and $v \equiv \mathsf{if}\ b\ \mathsf{then}\ (\mathsf{if}\ a\ \mathsf{then}\ s\ \mathsf{else}\ t)\ \mathsf{else}\ (\mathsf{if}\ c\ \mathsf{then}\ s\ \mathsf{else}\ t)$ then:

$$\overline{\mathsf{cond\text{-}st}}(u) = \overline{\mathsf{cond\text{-}st}}(b,a,c,s,t) \cup \overline{\mathsf{leave\text{-}st}}(b,a,c) = \overline{\mathsf{cond\text{-}st}}(v)$$

- $u \equiv f(\vec{w}, \mathsf{if}\ b\ \mathsf{then}\ \vec{s}\ \mathsf{else}\ \vec{t})$ and $v \equiv \mathsf{if}\ b\ \mathsf{then}\ f(\vec{w},\vec{s})\ \mathsf{else}\ f(\vec{w},\vec{t})$ then:

$$\overline{\mathsf{cond\text{-}st}}(u) = \overline{\mathsf{cond\text{-}st}}(b,\vec{w},\vec{s},\vec{t}) \cup \overline{\mathsf{leave\text{-}st}}(b) = \overline{\mathsf{cond\text{-}st}}(v)$$

- $(u \equiv \pi_i(\langle s_1\ ,\ s_2\rangle),\ v \equiv s_i)$, $(u \equiv \mathsf{dec}(\{m\}^{\mathsf{n}_r}_{\mathsf{pk(n)}}, \mathsf{sk(n)}),\ v \equiv m)$ and $(u \equiv \mathsf{eq}(x,x),\ v \equiv x)$ are trivial. ■

Corollary 5.1. *For every term u, $\overline{\mathsf{leave\text{-}st}}(u) \supseteq \mathsf{leave\text{-}st}(u\downarrow_R)$ and $\overline{\mathsf{cond\text{-}st}}(u) \supseteq \mathsf{cond\text{-}st}(u\downarrow_R)$.*

Let us show the following helpful propositions:

Proposition 5.10. *For all $\mathcal{S}_l$-normalized basic terms β, β' if:*

$$\overline{\mathsf{leave\text{-}st}}(\beta) \cap \overline{\mathsf{leave\text{-}st}}(\beta') \neq \emptyset$$

then we have $\mathcal{S}_l$-normalized basic terms $B[\vec{w}, (\alpha^j)_j, (\delta^k)_k]$ and $B[\vec{w}, (\alpha'^j)_j, (\delta'^k)_k]$ such that:

$$\beta \equiv B[\vec{w}, (\alpha^j)_j, (\delta^k)_k] \qquad \beta' \equiv B[\vec{w}, (\alpha'^j)_j, (\delta'^k)_k]$$

$$\forall j,\ \overline{\mathsf{leave\text{-}st}}(\alpha^j) \cap \overline{\mathsf{leave\text{-}st}}(\alpha'^j) \neq \emptyset \qquad \forall k,\ \overline{\mathsf{leave\text{-}st}}(\delta^k) \cap \overline{\mathsf{leave\text{-}st}}(\delta'^k) \neq \emptyset$$

Proof. We have $\mathcal{S}_l$-normalized basic terms $B[\vec{w}, (\alpha^j)_j, (\delta^k)_k]$ and $D[\vec{w}', (\alpha'^j)_j, (\delta'^k)_k]$ such that:

$$\beta \equiv B[\vec{w}, (\alpha^j)_j, (\delta^k)_k] \qquad \beta' \equiv D[\vec{w}', (\alpha'^j)_j, (\delta'^k)_k]$$

Since β, β' are $\mathcal{S}_l$-normalized basic terms, we know that:

$$B[\vec{w}, (\{[]_j\}_{\_})_j, (\mathsf{dec}([]_k, \_))_k] \qquad D[\vec{w}', (\{[]_j\}_{\_})_j, (\mathsf{dec}([]_k, \_))_k]$$

are in R-normal form, that $B[], D[], \vec{w}, \vec{w}'$ are if-free and that $B[], D[]$ do not contain $\mathbf{0}(\_)$. Hence:

$$\overline{\mathsf{leave\text{-}st}}(\beta) = \left\{B[\vec{w}, (a^j)_j, (d^k)_k] \mid \forall j, a^j \in \overline{\mathsf{leave\text{-}st}}(\alpha^j) \ \wedge\ \forall k, d^k \in \overline{\mathsf{leave\text{-}st}}(\delta^k)\right\}$$
$$\overline{\mathsf{leave\text{-}st}}(\beta') = \left\{D[\vec{w}', (a'^j)_j, (d'^k)_k] \mid \forall j, a'^j \in \overline{\mathsf{leave\text{-}st}}(\alpha'^j) \ \wedge\ \forall k, d'^k \in \overline{\mathsf{leave\text{-}st}}(\delta'^k)\right\}$$

Similarly to what we did in the proof of Lemma 5.2, we prove that we can assume that $B[] \equiv D[]$ by induction on the number of hole positions in $B[]$ or $D[]$ such that $(B[])_{|p}$ differs from $(D[])_{|p}$ (modulo hole renaming). Knowing that $B[] \equiv D[]$, it is then straightforward to show that:

$$\vec{w} \equiv \vec{w}' \qquad \forall j,\ \overline{\mathsf{leave\text{-}st}}(\alpha^j) \cap \overline{\mathsf{leave\text{-}st}}(\alpha'^j) \neq \emptyset \qquad \forall k,\ \overline{\mathsf{leave\text{-}st}}(\delta^k) \cap \overline{\mathsf{leave\text{-}st}}(\delta'^k) \neq \emptyset$$

The base case is trivial, let us prove the inductive case. Let $B[\vec{w}, (a^j)_j, (d^k)_k]$ and $D[\vec{w}', (a'^j)_j, (d'^k)_k]$ be such that:

$$\forall j, k.\, a^j \in \overline{\mathsf{leave\text{-}st}}(\alpha^j) \wedge d^k \in \overline{\mathsf{leave\text{-}st}}(\delta^k) \qquad \forall j, k.\, a'^j \in \overline{\mathsf{leave\text{-}st}}(\alpha'^j) \wedge d'^k \in \overline{\mathsf{leave\text{-}st}}(\delta'^k)$$

and:

$$B[\vec{w}, (a^j)_j, (d^k)_k] \equiv D[\vec{w}', (a'^j)_j, (d'^k)_k] \in \overline{\mathsf{leave\text{-}st}}(\beta) \cap \overline{\mathsf{leave\text{-}st}}(\beta')$$

First, observe that if a position p is valid in both $B[]$ and $D[]$, and is not a hole in both contexts, then $B[]$ and $D[]$ coincide on p.

Let p be the position of a hole in $B[]$ such that p is a valid position in $D[]$, but not a hole (if p is not valid in $D[]$, invert $B[]$ and $D[]$). We then have three cases depending on $(B[])_{|p}$:

- B contains a hole $[]_x$ at position p such that $\beta_{|p} \in \vec{w}$. Then let $\tilde{D}$ be the context D in which we replaced the term at position p by $[]_y$ (where y is a fresh hole variable) and let $\tilde{\vec{w}}'$ be the terms $\vec{w}'$ extended by $\beta_{|p}$ (binded to $[]_y$). Then B differs $\tilde{D}$ on a smaller number of hole position, therefore we can conclude by induction hypothesis.
- B contains a hole $[]_x$ at position p such that $\beta_{|p}$ is an encryption oracle call $\{m\}^{\mathsf{n}_r}_{\mathsf{pk}(\mathsf{n}_p)}$. Since $\{m\}^{\mathsf{n}_r}_{\mathsf{pk}(\mathsf{n}_p)} \in \mathcal{E}_l$ is an encryption in an instance of a CCA$_2$ application, we know from the freshness side-condition that n_r does not appear in $\vec{w}$ and that $\mathsf{n}_r \in \mathcal{R}_l$.
  Moreover since β' is a $\mathcal{S}_l$-normalized basic term, we know that $\mathsf{fresh}(\mathcal{R}_l; \vec{w}')$. But since p is a valid non-hole position in D, we have $\mathsf{n}_r \in \vec{w}'$. Absurd.
- Similarly if B contains a hole $[]_x$ at position p such that $\beta_{|p}$ is a decryption oracle call $\mathsf{dec}(m, \mathsf{sk}(\mathsf{n}))$. Since $\mathsf{dec}(m, \mathsf{sk}(\mathsf{n}))$ is a decryption oracle call we know that $\mathsf{sk}(\mathsf{n}) \in \mathcal{K}_l$. Moreover since β' is a $\mathcal{S}_l$-normalized basic term, we know that $\mathsf{nodec}(\mathcal{K}_l, \vec{w}')$. But since p is a valid non-hole position in D, we know that either $\mathsf{sk}(\mathsf{n}) \in \vec{w}'$ or $\mathsf{n} \in \vec{w}'$. Absurd. ■

We can now state the following proposition.

Proposition 5.11. *For all $\mathcal{S}_l$-normalized basic terms β, β', we have $\beta \equiv \beta'$ whenever:*

$$\overline{\mathsf{leave\text{-}st}}(\beta) \cap \overline{\mathsf{leave\text{-}st}}(\beta') \neq \emptyset$$

Proof. We show this by induction on $|\beta|+|\beta'|$. Using Proposition 5.10 we know that we have $\mathcal{S}_l$-normalized basic terms $B[\vec{w}, (\alpha^j)_j, (\delta^k)_k], B[\vec{w}, (\alpha'^j)_j, (\delta'^k)_k]$ such that:

$$\beta \equiv B[\vec{w}, (\alpha^j)_j, (\delta^k)_k] \qquad \beta' \equiv B[\vec{w}, (\alpha'^j)_j, (\delta'^k)_k]$$

$$\forall j,\ \overline{\mathsf{leave\text{-}st}}(\alpha^j) \cap \overline{\mathsf{leave\text{-}st}}(\alpha'^j) \neq \emptyset \qquad \forall k,\ \overline{\mathsf{leave\text{-}st}}(\delta^k) \cap \overline{\mathsf{leave\text{-}st}}(\delta'^k) \neq \emptyset$$

To conclude we only need to show that for all j, $\overline{\mathsf{leave\text{-}st}}(\alpha^j) \cap \overline{\mathsf{leave\text{-}st}}(\alpha'^j) \neq \emptyset$ implies that $\alpha^j \equiv \alpha'^j$ and that $\overline{\mathsf{leave\text{-}st}}(\delta^k) \cap \overline{\mathsf{leave\text{-}st}}(\delta'^k) \neq \emptyset$ implies that $\delta^k \equiv \delta'^k$. The former is immediate, as $\overline{\mathsf{leave\text{-}st}}(\alpha^j) \cap \overline{\mathsf{leave\text{-}st}}(\alpha'^j) \neq \emptyset$ implies that $\alpha^j \equiv \{m\}^{\mathsf{n}_r}_{\mathsf{pk(n)}}$ and $\alpha'^j \equiv \{m'\}^{\mathsf{n}_r}_{\mathsf{pk(n)}}$. Since $\alpha^j, \alpha'^j \in \mathcal{E}_l$ and since there is *as most one* $\mathcal{S}_l$-encryption oracle call with the same randomness, we have $m \equiv m'$. It only remains to show that for all k, $\delta^k \equiv \delta'^k$. Since δ^k, δ'^k are $\mathcal{S}_l$-decryption oracle calls we know that

$$\delta^k \equiv C\,[\vec{g} \diamond (s_i)_{i \leq p}] \qquad\qquad \delta'^k \equiv C'\,[\vec{g}' \diamond (s'_i)_{i \leq p'}]$$

where:

- There exists contexts u, u', if-free and in R-normal form, such that:

$$\forall i < p,\ s_i \equiv \mathbf{0}(\mathsf{dec}(u[(\alpha_j)_j, (\mathsf{dec}_k)_k], \mathsf{sk})) \qquad s_p \equiv \mathsf{dec}(u[(\alpha_j)_j, (\mathsf{dec}_k)_k], \mathsf{sk})$$

$$\forall g \in \vec{g},\ g \equiv \mathsf{eq}(u[(\alpha_j)_j, (\mathsf{dec}_k)_k], \alpha_g) \text{ where } \alpha_g \in (\alpha_j)_j$$

$$\forall i < p',\ s'_i \equiv \mathbf{0}(\mathsf{dec}(u'[(\alpha'_j)_j, (\mathsf{dec}'_k)_k], \mathsf{sk}')) \qquad s'_p \equiv \mathsf{dec}(u'[(\alpha'_j)_j, (\mathsf{dec}'_k)_k], \mathsf{sk}')$$

$$\forall g \in \vec{g}',\ g \equiv \mathsf{eq}(u'[(\alpha'_j)_j, (\mathsf{dec}'_k)_k], \alpha'_g) \text{ where } \alpha'_g \in (\alpha'_j)_j$$

- $(\alpha_j)_j, (\alpha'_j)_j$ are *$\mathcal{S}_l$-encryption oracle calls.*
- $(\mathsf{dec}_k)_k, (\mathsf{dec}'_k)_k$ are *$\mathcal{S}_l$-decryption oracle call.*

Since $\overline{\mathsf{leave\text{-}st}}(\delta^k) \cap \overline{\mathsf{leave\text{-}st}}(\delta'^k) \neq \emptyset$, and since u, u' are if-free and in R-normal form we know that $u \equiv u'$, for all j, $\overline{\mathsf{leave\text{-}st}}(\alpha_j) \cap \overline{\mathsf{leave\text{-}st}}(\alpha'_j)$ and for all k, $\overline{\mathsf{leave\text{-}st}}(\mathsf{dec}_k) \cap \overline{\mathsf{leave\text{-}st}}(\mathsf{dec}'_k)$. It follows, by induction hypothesis, that for all j, $\alpha_j \equiv \alpha'_j$ and for all k, $\mathsf{dec}_k \equiv \mathsf{dec}'_k$. We only have to check that the guards are the same. Since $\delta^k, \delta'^k \in \mathcal{D}_l$, we know from the definition of the CCA_2 axioms that δ^k (resp. δ'^k) has one guard for every encryption $\alpha \in \mathcal{E}_l$ such that $\alpha \equiv \{\_\}^{\mathsf{n}}_{\mathsf{pk}}$ and n appear directly in s_p (resp. s'_p). Since we showed that $s_p \equiv s'_p$, we deduce that δ^k, δ'^k have the same guards. Since guards are sorted according to an arbitrary but fixed order (the sort function in the definition of $R^{\mathcal{K}}_{\text{CCA}^a_2}$), we know that $\delta^k \equiv \delta'^k$. ■

Corollary 5.2. *For all $P \vdash^{\mathsf{npf}} t \sim t'$, for all h, l:*

(i) *for all $\beta, \beta' \leq^{h,l}_{\mathsf{c}} (t, P)$ if $\mathsf{leave\text{-}st}(\beta \downarrow_R) \cap \mathsf{leave\text{-}st}(\beta' \downarrow_R) \neq \emptyset$ then $\beta \equiv \beta'$.*

(ii) *for all $\gamma, \gamma' \leq^{h,l}_{l} (t, P)$ if $\mathsf{leave\text{-}st}(\gamma \downarrow_R) \cap \mathsf{leave\text{-}st}(\gamma' \downarrow_R) \neq \emptyset$ then $\gamma \equiv \gamma'$.*

(iii) *for all $\beta \leq^{h,l}_{\mathsf{c}} (t, P)$, $\gamma \leq^{h,l}_{l} (t, P)$ if $\mathsf{leave\text{-}st}(\beta \downarrow_R) \cap \mathsf{leave\text{-}st}(\gamma \downarrow_R) \neq \emptyset$ then $\beta \equiv \gamma$.*

We can now show the following lemma, which subsumes Lemma 5.12:

Lemma 5.13. *For all $P \vdash^{\mathsf{npf}} t \sim t'$, for all h, l and $\beta, \beta' \leq^{h,l}_{bt} (t, P)$, there exists an almost conditional context $\tilde{\beta}'[]$ such that:*

$$\beta' \equiv \tilde{\beta}'\,[\beta] \quad \text{ and } \quad \mathsf{leave\text{-}st}(\beta \downarrow_R) \cap \overline{\mathsf{cond\text{-}st}}\big(\tilde{\beta}'[]\big) = \emptyset$$

Proof. Let $l \in \mathsf{label}(P)$. We prove by mutual induction on the definition of $\mathcal{S}_l$-normalized simple terms, $\mathcal{S}_l$-normalized basic terms, $\mathcal{S}_l$-encryption oracle calls and $\mathcal{S}_l$-decryption oracle calls that for every $u \in \mathsf{st}(\beta')$ such that u is in one of the four above cases, there exists a conditional context $u_c[]$ such that:

$$u \equiv u_c\,[\beta] \qquad \mathsf{leave\text{-}st}(\beta \downarrow_R) \cap \overline{\mathsf{cond\text{-}st}}\,(u_c[]) = \emptyset \qquad \overline{\mathsf{leave\text{-}st}}(\vec{u}_c) = \overline{\mathsf{leave\text{-}st}}(\vec{u})$$

Moreover if u is a $\mathcal{S}_l$-normalized basic term then there exists an *almost* conditional context $u_d[]$ such that:

$$u \equiv u_d\,[\beta] \qquad \mathsf{leave\text{-}st}(\beta \downarrow_R) \cap \big(\overline{\mathsf{cond\text{-}st}}\,(u_d[]) \cup \overline{\mathsf{leave\text{-}st}}\,(u_d[])\big) = \emptyset$$

- **Normalized Simple Term:** Let $u \equiv C[\vec{b} \diamond \vec{s}]$, where $\vec{b}$ are $\mathcal{S}_l$-normalized basic conditionals and $\vec{s}$ are $\mathcal{S}_l$-normalized basic terms. Let $\vec{b}_d[]$ and $\vec{s}_c[]$ be contexts obtained from $\vec{b}, \vec{s}$ by induction hypothesis such that $\vec{b}, \vec{s} \equiv \vec{b}_d[\beta], \vec{s}_c[\beta]$ and:

$$\overline{\mathsf{leave\text{-}st}}(\vec{s}_c[]) = \overline{\mathsf{leave\text{-}st}}(\vec{s}) \qquad \mathsf{leave\text{-}st}(\beta \downarrow_R) \cap \Big(\overline{\mathsf{cond\text{-}st}}\big(\vec{b}_d[], \vec{s}_c[]\big) \cup \overline{\mathsf{leave\text{-}st}}\big(\vec{b}_d[]\big)\Big) = \emptyset$$

Moreover:

$$\overline{\mathsf{cond\text{-}st}}(C[\vec{b}_d[] \diamond \vec{s}_c[]]) = \overline{\mathsf{cond\text{-}st}}(\vec{b}_d[], \vec{s}_c[]) \cup \overline{\mathsf{leave\text{-}st}}(\vec{b}_d[])$$
$$\overline{\mathsf{leave\text{-}st}}(C[\vec{b}_d[] \diamond \vec{s}_c[]]) = \overline{\mathsf{leave\text{-}st}}(\vec{s}_c[]) = \overline{\mathsf{leave\text{-}st}}(\vec{s}) = \overline{\mathsf{leave\text{-}st}}(C[\vec{b} \diamond \vec{s}])$$

Hence we can take $\vec{u}_c \equiv C[\vec{b}_d[] \diamond \vec{s}_c[]]$.

- **Normalized Basic Term:** Let $u \equiv B[\vec{w}, (\alpha^i)_i, (\mathsf{dec}^j)_j]$ be a $\mathcal{S}_l$-normalized basic term. Let $(\alpha^i_c)_i, (\alpha^i_d)_i$ and $(\mathsf{dec}^j_c)_j, (\mathsf{dec}^j_d)_j$ be the contexts obtained by applying the induction hypothesis to $(\alpha^i)_i$ and $(\mathsf{dec}^j)_j$. Using the fact that:

$$\overline{\mathsf{leave\text{-}st}}\big((\alpha^i_c)_i, (\mathsf{dec}^j_c)_i\big) = \overline{\mathsf{leave\text{-}st}}\big((\alpha^i)_i, (\mathsf{dec}^j)_i\big)$$

and since B and $\vec{w}$ are if-free, one can check that:

$$\overline{\mathsf{leave\text{-}st}}\big(B[\vec{w}, (\alpha^i_c)_i, (\mathsf{dec}^j_c)_j]\big) = \overline{\mathsf{leave\text{-}st}}\big(B[\vec{w}, (\alpha^i)_i, (\mathsf{dec}^j)_j]\big)$$

It is then immediate to check that $u_c \equiv B[\vec{w}, (\alpha^i_c)_i, (\mathsf{dec}^j_c)_j]$ satisfies the wanted properties.
It remains to construct the context $u_d[]$. If $\mathsf{leave\text{-}st}(\beta \downarrow_R) \cap \overline{\mathsf{leave\text{-}st}}(u) = \emptyset$ then $u_d \equiv u_c$ satisfies the wanted properties. Otherwise using Proposition 5.11 we know that $\beta \equiv u$, hence we can take $u_d \equiv []$.
- **Encryption Oracle Call:** The proof done for the normalized basic term case applies here.
- **Decryption Oracle Call:** The proof done for the normalized simple term case applies here. ■

5.9.2 Well-Nested Sets

Definition 5.45. A simple term $C[\vec{a} \diamond \vec{b}]$ is said to be *flat* if $\vec{a}, \vec{b}$ are if-free terms in R-normal forms.

Definition 5.46. We let *well-nested* be the smallest relation between sets $(\mathcal{C}, \mathcal{D})$ of flat terms such that:

(a) $(\mathcal{C}, \mathcal{D})$ is well-nested if for every $C_0[\vec{a}_0 \diamond \vec{b}_0] \in \mathcal{C}$:

$$\forall C[\vec{a} \diamond \vec{b}] \in \mathcal{C} \cup \mathcal{D}, \quad \vec{b}_0 \cap \vec{a} = \emptyset$$

(b) $(\mathcal{C}, \mathcal{D})$ is well-nested if for every $\beta_0 \equiv C_0[\vec{a}_0 \diamond \vec{b}_0] \in \mathcal{C}$:

(i) For all $\beta \equiv C[\vec{a} \diamond \vec{b}] \in \mathcal{C} \cup \mathcal{D}$, there exist two if-contexts C'_β, C''_β such that:

$$\beta =_R \mathsf{if}\ \beta_0\ \mathsf{then}\ C'_\beta[\vec{a}'_\beta \diamond \vec{b}'_\beta]\ \mathsf{else}\ C''_\beta[\vec{a}''_\beta \diamond \vec{b}''_\beta]$$

where $\vec{a}'_\beta, \vec{a}''_\beta \subseteq \vec{a} \backslash \vec{b}_0$ and $\vec{b}'_\beta, \vec{b}''_\beta \subseteq \vec{b}$.

(ii) The following couples of sets are well-nested:

$$\Big(\Big\{C'_\beta[\vec{a}'_\beta \diamond \vec{b}'_\beta] \mid C[\vec{a} \diamond \vec{b}] \in \mathcal{C}\Big\}, \Big\{C'_\beta[\vec{a}'_\beta \diamond \vec{b}'_\beta] \mid C[\vec{a} \diamond \vec{b}] \in \mathcal{D}\Big\}\Big)$$
$$\Big(\Big\{C''_\beta[\vec{a}''_\beta \diamond \vec{b}''_\beta] \mid C[\vec{a} \diamond \vec{b}] \in \mathcal{C}\Big\}, \Big\{C''_\beta[\vec{a}''_\beta \diamond \vec{b}''_\beta] \mid C[\vec{a} \diamond \vec{b}] \in \mathcal{D}\Big\}\Big)$$

Proposition 5.12. *If $(\mathcal{C}, \mathcal{D})$ verifies the property (a) above, then it satisfies properties (i) and (ii).*

Proof. Trivial by taking $C'_\beta \equiv C''_\beta \equiv C$. ■

Definition 5.47. We let head be the partial function defined on terms such that for all $f \in \mathcal{F}$, for all terms $\vec{t}$, $\mathsf{head}(f(\vec{t})) \equiv f$.

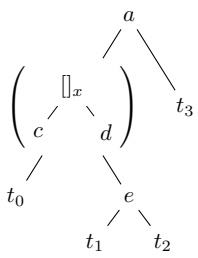
Definition 5.48. For all conditional contexts C_0, C_1, we let $C_0 \sqcup_{\mathsf{c}} C_1$ be the conditional context, if it exists, defined as follows: $\mathsf{pos}(C_1 \sqcup_{\mathsf{c}} C_2) = \mathsf{pos}(C_0) \cap \mathsf{pos}(C_1)$ and for all position p in $\mathsf{pos}(C_0 \sqcup_{\mathsf{c}} C_1)$:

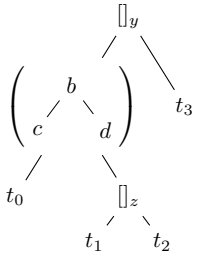
$$(C_0 \sqcup_{\mathsf{c}} C_1)_{|p} \equiv \begin{cases} a & \text{if } \mathsf{head}((C_0)_{|p}) \equiv \mathsf{head}((C_1)_{|p}) \equiv a \qquad (a \in \mathcal{F} \cup \mathcal{N}) \\ []_x & \text{if } (C_0)_{|p} \equiv []_x \wedge \big(\mathsf{head}((C_1)_{|p}) \equiv []_x \vee \mathsf{head}((C_1)_{|p}) \equiv a\big) \qquad (a \in \mathcal{F} \cup \mathcal{N}) \\ []_x & \text{if } (C_1)_{|p} \equiv []_x \wedge \big(\mathsf{head}((C_0)_{|p}) \equiv []_x \vee \mathsf{head}((C_0)_{|p}) \equiv a\big) \qquad (a \in \mathcal{F} \cup \mathcal{N}) \end{cases}$$

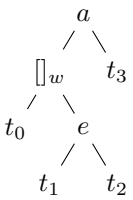
If such a conditional context does not exist, then $C_0 \sqcup_{\mathsf{c}} C_1$ is the special element $\mathsf{undefined}$. We also let:

$$\mathsf{undefined} \sqcup_{\mathsf{c}} C_0 \equiv C_0 \sqcup_{\mathsf{c}} \mathsf{undefined} \equiv \mathsf{undefined}$$

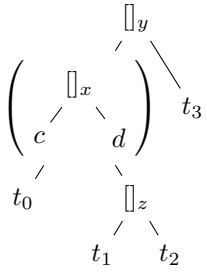
Example 5.16. For all conditionals a, b, c, d, e, f and terms $t_0, \ldots, t_3$, if we let:

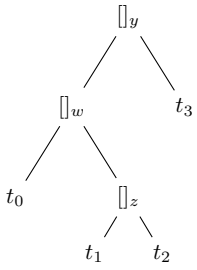
$C_0 \equiv$ 

$C_1 \equiv$ 

$C_2 \equiv$ 

Then we have:

$C_0 \sqcup_{\mathsf{c}} C_1 \equiv$ 

$C_1 \sqcup_{\mathsf{c}} C_2 \equiv$ 

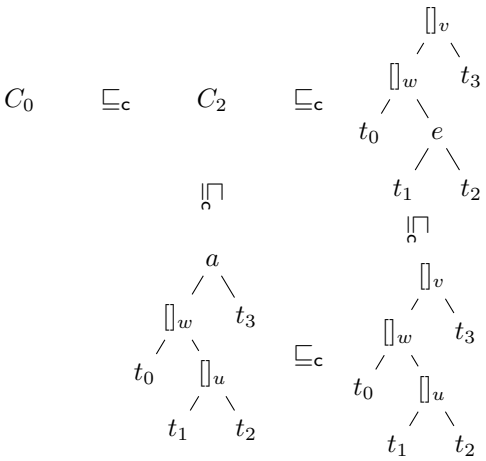
$C_0 \sqcup_{\mathsf{c}} C_2 \equiv C_2$ □

Definition 5.49. We let $\sqsubseteq_{\mathsf{c}}$ be the relation on conditional contexts defined as follows: for all conditional contexts C_0, C_1, we let $C_0 \sqsubseteq_{\mathsf{c}} C_1$ hold if $\mathsf{pos}(C_1) \subseteq \mathsf{pos}(C_0)$ and for all position p in $\mathsf{pos}(C_1)$:

$$\text{if } \mathsf{head}((C_1)_{|p}) \equiv \begin{cases} a & \text{then } \mathsf{head}((C_0)_{|p}) \equiv a \qquad \text{where } (a \in \mathcal{F} \cup \mathcal{N}) \\ []_x & \text{then } \mathsf{head}((C_0)_{|p}) \in \mathcal{F} \cup \mathcal{N} \cup \{[]_x\} \end{cases}$$

Moreover we let $C_0 \sqsubseteq_{\mathsf{c}} \mathsf{undefined}$ for all conditional context C_0 (and $\mathsf{undefined} \sqsubseteq_{\mathsf{c}} \mathsf{undefined}$).

Example 5.17. Using the conditional contexts defined in Example 5.16, we have, for example, the following relations:



□

Proposition 5.13. *Let $\mathcal{S}_{cc}$ be the set of conditional contexts extended with* undefined*. Then $(\mathcal{S}_{cc}, \sqcup_{\mathsf{c}}, \sqsubseteq_{\mathsf{c}})$ is a semi-lattice. That is, we have the following properties:*

(i) $\sqcup_{\mathsf{c}}$ is associative, commutative, idempotent.

(ii) $\sqsubseteq_{\mathsf{c}}$ is an pre-order (i.e. reflexive and transitive).

(iii) For all $C_0, C_1 \in \mathcal{S}_{cc}$, we have $C_0 \sqsubseteq_{\mathsf{c}} (C_0 \sqcup_{\mathsf{c}} C_1)$ and $C_1 \sqsubseteq_{\mathsf{c}} (C_0 \sqcup_{\mathsf{c}} C_1)$. Moreover $(C_0 \sqcup_{\mathsf{c}} C_1)$ is the least upper-bound of C_0 and C_1.

Proof. These properties are straightforward to show, we are only going to give the proof that $(C_0 \sqcup_{\mathsf{c}} C_1)$ is the least upper-bound of C_0 and C_1. Assume that there is C such that:

$$C_0 \sqsubseteq_{\mathsf{c}} C \sqsubseteq_{\mathsf{c}} C_0 \sqcup_{\mathsf{c}} C_1 \qquad\qquad C_1 \sqsubseteq_{\mathsf{c}} C \sqsubseteq_{\mathsf{c}} C_0 \sqcup_{\mathsf{c}} C_1$$

If $C_0 \sqcup_{\mathsf{c}} C_1 \equiv \mathsf{undefined}$ then one can check that $C \equiv \mathsf{undefined}$. Otherwise we know that $\mathsf{pos}(C_0 \sqcup_{\mathsf{c}} C_1) = \mathsf{pos}(C_0) \cap \mathsf{pos}(C_1)$, and that:

$$\mathsf{pos}(C_0) \supseteq \mathsf{pos}(C) \supseteq \mathsf{pos}(C_0 \sqcup_{\mathsf{c}} C_1) \qquad\qquad \mathsf{pos}(C_1) \supseteq \mathsf{pos}(C) \supseteq \mathsf{pos}(C_0 \sqcup_{\mathsf{c}} C_1)$$

Hence $\mathsf{pos}(C) = \mathsf{pos}(C_0 \sqcup_{\mathsf{c}} C_1)$. Using the fact that $C \sqsubseteq_{\mathsf{c}} C_0 \sqcup_{\mathsf{c}} C_1$ we know that for all position $p \in \mathsf{pos}(C)$, if $\mathsf{head}((C_0 \sqcup_{\mathsf{c}} C_1)_{|p}) = a$ (with $a \in \mathcal{F} \cup \mathcal{N}$) then $\mathsf{head}(C_{|p}) = a$. If $\mathsf{head}((C_0 \sqcup_{\mathsf{c}} C_1)_{|p}) = []_x$ then either $\mathsf{head}(C_{|p}) = []_x$ or $\mathsf{head}(C_{|p}) = a$ (with $a \in \mathcal{F} \cup \mathcal{N}$). In the former case there is nothing to show, in the the latter case observe that $\mathsf{head}((C_0 \sqcup_{\mathsf{c}} C_1)_{|p}) = []_x$ implies that either $\mathsf{head}((C_0)_{|p}) = []_x$ or $\mathsf{head}((C_1)_{|p}) = []_x$. W.l.o.g. assume $\mathsf{head}((C_0)_{|p}) = []_x$. Then using the fact that $C_0 \sqsubseteq_{\mathsf{c}} C$, we know that $\mathsf{head}((C_0)_{|p}) = []_x$ implies that $\mathsf{head}((C_0)_{|p}) = []_x$. Absurd.

Therefore $\forall p \in \mathsf{pos}(C)$, $\mathsf{head}(C_{|p}) = \mathsf{head}((C_0 \sqcup_{\mathsf{c}} C_1)_{|p})$. Moreover $\mathsf{pos}(C) = \mathsf{pos}(C_0 \sqcup_{\mathsf{c}} C_1)$. Hence $C \equiv C_0 \sqcup_{\mathsf{c}} C_1$. ■

Let $C_0, C_1 \in \mathcal{S}_{cc}$ such that $C_0 \sqsubseteq_{\mathsf{c}} C_1$. Moreover, assume that:

$$\forall p, p' \in \mathsf{pos}(C_1), (C_1)_{|p} \equiv (C_1)_{|p'} \equiv []_x \;\Rightarrow\; (C_0)_{|p} \equiv (C_0)_{|p'}$$

Then, we know that C_0 and C_1 coincides on $\mathsf{pos}(C_1)$. Therefore, any $\to_R$ reduction done on C_1 can be mimicked on C_0. We simultaneously reduce C_1 and C_0, which yields the terms C_1' and C_0', where C_1' is in R-normal form. Then the conditionals of C_1' which do not have hole variables (i.e. $\mathsf{cond\text{-}st}(C_1' \downarrow_R) \cap \mathcal{T}(\mathcal{F}_{\setminus \mathsf{if}}, \mathcal{N})$) all appear directly as subterm of C_0', hence are in $\overline{\mathsf{cond\text{-}st}}(C_0')$.

Proposition 5.14. *For every $C_0, C_1 \in \mathcal{S}_{cc}$, if $C_0 \sqsubseteq_{\mathsf{c}} C_1$ and if:*

$$\forall p, p' \in \mathit{pos}(C_1), (C_1)_{|p} \equiv (C_1)_{|p'} \equiv []_x \;\Rightarrow\; (C_0)_{|p} \equiv (C_0)_{|p'}$$

then $\mathsf{cond\text{-}st}(C_1 \downarrow_R) \cap \mathcal{T}(\mathcal{F}_{\setminus \mathit{if}}, \mathcal{N}) \subseteq \overline{\mathsf{cond\text{-}st}}(C_0)$.

Proof. Assume that $C_0 \sqsubseteq_{\mathsf{c}} C_1$, with $C_0, C_1 \neq \mathsf{undefined}$ (the cases $C_0 = \mathsf{undefined}$ and $C_1 = \mathsf{undefined}$ are easy to handle, with the convention that $\mathsf{cond\text{-}st}(\mathsf{undefined}) = \emptyset$), and that:

$$\forall p, p' \in \mathsf{pos}(C_1), (C_1)_{|p} \equiv (C_1)_{|p'} \equiv []_x \;\Rightarrow\; (C_0)_{|p} \equiv (C_0)_{|p'} \tag{5.8}$$

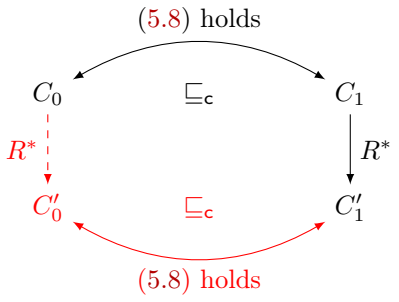
First we show that we can extend this property as follows:

$$\forall p, p' \in \mathsf{pos}(C_1), (C_1)_{|p} \equiv (C_1)_{|p'} \;\Rightarrow\; (C_0)_{|p} \equiv (C_0)_{|p'} \tag{5.9}$$

Let $q = p \cdot q_0$ and $q = p' \cdot q_0$ be positions in $\mathsf{pos}(C_1)$. Since $(C_1)_{|p} \equiv (C_1)_{|p'}$, we know that $\mathsf{head}((C_1)_{|q}) \equiv \mathsf{head}((C_1)_{|q'})$.

- If $\mathsf{head}((C_1)_{|q}) \equiv a$ (with $a \in \mathcal{F} \cup \mathcal{N}$) then, from the fact that $C_0 \sqsubseteq_{\mathsf{c}} C_1$ we get that $\mathsf{head}((C_0)_{|q}) \equiv a$, and that $\mathsf{head}((C_0)_{|q'}) \equiv a$.
- If $\mathsf{head}((C_1)_{|q}) \equiv []_x$ then using (5.8) we get that $(C_0)_{|p} \equiv (C_0)_{|p'}$.

Then, we show by induction on the length of the reduction sequence that for all C_1' such that $C_1 \to_R^* C_1'$, there exists C_0' such that $C_0' \sqsubseteq_{\mathsf{c}} C_1'$, (5.8) holds for C_0', C_1' and $C_0 \to_R^* C_0'$. Graphically (hypothesis are in black, goals are in red):



Let $\to_{R'}$ be $\to_R$ without the non left-linear rules, which are:

$$\mathsf{if}\ x\ \mathsf{then}\ y\ \mathsf{else}\ y \to y \qquad \mathsf{dec}(\{x\}^r_{\mathsf{pk}(y)}, \mathsf{sk}(y)) \to x$$

$$\mathsf{if}\ w\ \mathsf{then}\ (\mathsf{if}\ w\ \mathsf{then}\ x\ \mathsf{else}\ y)\ \mathsf{else}\ z \to \mathsf{if}\ w\ \mathsf{then}\ x\ \mathsf{else}\ z$$

$$\mathsf{if}\ w\ \mathsf{then}\ x\ \mathsf{else}\ (\mathsf{if}\ w\ \mathsf{then}\ y\ \mathsf{else}\ z) \to \mathsf{if}\ w\ \mathsf{then}\ x\ \mathsf{else}\ z$$

We mimic all reduction $\to_R$ on C_1 by a reduction on C_0, while maintaining $\sqsubseteq_{\mathsf{c}}$ and the invariant of (5.8). Mimicking rules in $\to_R$ is easy as they are left-linear. To mimic rules in $(\to_R \setminus \to_{R'})$, we use (5.9).

Therefore let C_1' be in R-normal form such that $C_1 \to_R^* C_1'$. Let C_0' be such that $C_0' \sqsubseteq_{\mathsf{c}} C_1'$,(5.8) holds for C_0', C_1' and $C_0 \to_R^* C_0'$. C_1' is of the form $D[\vec{b}, \vec{b}_{[]} \diamond \vec{u}]$ where $\vec{b}, \vec{u}$ are if-free and in R-normal form, $\vec{b}$ does not contain any hole variable and $\vec{b}_{[]}$ is a vector of hole variables. Therefore:

$$\mathsf{cond\text{-}st}(C_1 \downarrow_R) \cap \mathcal{T}(\mathcal{F}_{\setminus \mathsf{if}}, \mathcal{N}) = \mathsf{cond\text{-}st}(C_1') \cap \mathcal{T}(\mathcal{F}_{\setminus \mathsf{if}}, \mathcal{N}) = \vec{b}$$

Finally, we observe that $\vec{b} \subseteq \overline{\mathsf{cond\text{-}st}}(C_0')$, and that $\overline{\mathsf{cond\text{-}st}}(C_0') \subseteq \overline{\mathsf{cond\text{-}st}}(C_0)$ by Proposition 5.9. ■

Lemma 5.14. *For all $P \vdash^{npf} t \sim t'$, for all h, l, the following couple of sets is well-nested:*

$$\Big(\{\beta \downarrow_R |\ \beta \leq_{\mathsf{c}}^{\mathsf{h},l} (t, P)\}, \{\gamma \downarrow_R |\ \gamma \leq_{\mathsf{l}}^{\mathsf{h},l} (t, P)\}\Big)$$

Proof. We do this proof in the case $\mathsf{h} = \epsilon$. The other cases are identical.

Part 1 We consider an arbitrary ordering $(\beta_i)_{1 \leq i \leq i_{\max}}$ of:

$$\{\beta \mid \beta \leq_{\mathsf{c}}^{\mathsf{h},l} (t, P)\}$$

Using Lemma 5.13, we know that all $i \neq i_0$, there exists a conditional context $\tilde{\beta}_i$ such that:

$$\beta_i \equiv \tilde{\beta}_i\,[\beta_{i_0}] \quad \text{and} \quad \mathsf{leave\text{-}st}(\beta_{i_0} \downarrow_R) \cap \overline{\mathsf{cond\text{-}st}}(\tilde{\beta}_i) = \emptyset$$

From now on we use $\beta_i^{(i_0)}$ to denote this conditional context, and $[]_{i_0}$ the hole variable used in the conditional contexts $\{\beta_i^{(i_0)} \mid 1 \leq i \leq i_{\max}\}$. We extend this notation to $i_0 = 0$ by letting $\beta_i^{(0)} \equiv \beta_i$.

Let $1 \leq i \leq i_{\max}$, and let $l_0, \ldots, l_n$ be a sequence of distinct indices in $\{0, \ldots, i_{\max}\}$ such that $l_0 = 0$. Using Proposition 5.13.((iii)) we know that for every $0 \leq j_0 \leq n$, if $i \neq l_{j_0}$ then:

$$\beta_i^{(l_{j_0})} \sqsubseteq_{\mathsf{c}} \sqcup_{\mathsf{c} j \leq n} \beta_i^{(l_j)}$$

Using Proposition 5.14, we deduce that:

$$\overline{\mathsf{cond\text{-}st}}(\beta_i^{(l_{j_0})}) \supseteq \mathsf{cond\text{-}st}\big(\sqcup_{\mathsf{c} j \leq n} \beta_i^{(l_j)} \downarrow_R\big) \cap \mathcal{T}(\mathcal{F}_{\setminus \mathsf{if}}, \mathcal{N})$$

Which implies that:

$$\mathsf{leave\text{-}st}(\beta_{l_{j_0}} \downarrow_R) \cap \mathsf{cond\text{-}st}\big(\sqcup_{\mathsf{c} j \leq n} \beta_i^{(l_j)} \downarrow_R\big) = \emptyset \tag{5.10}$$

Moreover, if $n = n_0 + 1$ and $i \neq l_{n+1}$, we can check that:

$$\begin{aligned} \sqcup_{\mathsf{c} j \leq n_0} \beta_i^{(l_j)} \ &\equiv\ \big(\sqcup_{\mathsf{c} j \leq n_0+1} \beta_i^{(l_j)}\big)\{\sqcup_{\mathsf{c} j \leq n_0} \beta_{l_{n_0+1}}^{(l_j)} / []_{l_{n_0+1}}\} \\ &=_R \mathsf{if}\ \big(\sqcup_{\mathsf{c} j \leq n_0} \beta_{l_{n_0+1}}^{(l_j)}\big)\ \mathsf{then}\ \big(\sqcup_{\mathsf{c} j \leq n_0+1} \beta_i^{(l_j)}\big)\{\mathsf{true}/[]_{l_{n_0+1}}\} \\ &\qquad\qquad \mathsf{else}\ \big(\sqcup_{\mathsf{c} j \leq n_0+1} \beta_i^{(l_j)}\big)\{\mathsf{false}/[]_{l_{n_0+1}}\} \end{aligned} \tag{5.11}$$

Part 2 Similarly, let $(\gamma_m)_{1 \leq m \leq m_{\max}}$ be an arbitrary ordering of:

$$\{\gamma \mid \gamma \leq_{\mathsf{l}}^{\mathsf{h},l} (t, P)\}$$

Let $1 \leq i_0 \leq i_{\max}$. For every m, we have $\gamma_m^{(i_0)}$ such that:

$$\gamma_m \equiv \gamma_m^{(i_0)}\,[\beta_{i_0}] \quad \text{and} \quad \mathsf{leave\text{-}st}(\beta_{i_0} \downarrow_R) \cap \overline{\mathsf{cond\text{-}st}}(\gamma_m^{(i_0)}) = \emptyset$$

Moreover, we let $\gamma_m^{(0)} \equiv \gamma_m$. Let $1 \leq m \leq m_{\max}$, and let $l_0, \ldots, l_n$ be a sequence of distinct indices in $\{0, \ldots, i_{\max}\}$ such that $l_0 = 0$. We have equations corresponding to (5.10) and (5.11), with $\sqcup_{\mathsf{c} j \leq n} \gamma_m^{(l_j)}$ instead of $\sqcup_{\mathsf{c} j \leq n} \beta_i^{(l_j)}$.

Part 3 Consider the following family of couples of sets:

$$\Big\{\Big(\big(\sqcup_{\mathsf{c}j\leq n}\,\beta_i^{(l_j)}\{e_j/[]_{l_j}\mid j\leq n\}\downarrow_R\big)_{1\leq i\leq i_{\max}},\big(\sqcup_{\mathsf{c}j\leq n}\,\gamma_m^{(l_j)}\{e_j/[]_{l_j}\mid j\leq n\}\downarrow_R\big)_{1\leq m\leq m_{\max}}\Big)\mid \\ l_0,\ldots,l_n \text{ distinct indices in } \{0,\ldots,i_{\max}\} \text{ s.t. } l_0=0 \text{ and } (e_j)_{1\leq j\leq n}\in\{\mathsf{true},\mathsf{false}\}^n\Big\} \quad (5.12)$$

We show by decreasing induction on n, starting from $n=i_{\max}+1$ down to $n=0$, that all the elements above are well-nested.

Let $l_0,\ldots,l_n$ be distinct indices in $\{0,\ldots,i_{\max}\}$ such that $l_0=0$, and let $(e_j)_{1\leq j\leq n}\in\{\mathsf{true},\mathsf{false}\}^n$.

Base case If $n=n_{\max}+1$ then from (5.10) we get that for every $l\neq i$ in $\{1,\ldots,i_{\max}\}$:

$$\mathsf{leave\text{-}st}(\beta_l\downarrow_R)\cap\mathsf{cond\text{-}st}\big(\big(\sqcup_{\mathsf{c}j\leq n}\,\beta_i^{(j)}\big)\{e_j/[]_j\mid j\leq n\}\downarrow_R\big)=\emptyset$$

Moreover:

$$\mathsf{leave\text{-}st}\big(\big(\sqcup_{\mathsf{c}j\leq n}\,\beta_l^{(j)}\big)\{e_j/[]_j\mid j\leq n\}\downarrow_R\big)\subseteq\mathsf{leave\text{-}st}(\beta_l\downarrow_R)$$

Hence:

$$\mathsf{leave\text{-}st}\big(\big(\sqcup_{\mathsf{c}j\leq n}\,\beta_l^{(j)}\big)\{e_j/[]_j\mid j\leq n\}\downarrow_R\big)\cap\mathsf{cond\text{-}st}\big(\big(\sqcup_{\mathsf{c}j\leq n}\,\beta_i^{(j)}\big)\{e_j/[]_j\mid j\leq n\}\downarrow_R\big)=\emptyset \quad (5.13)$$

Similarly, for every $1\leq m\leq m_{\max}$:

$$\mathsf{leave\text{-}st}\big(\big(\sqcup_{\mathsf{c}j\leq n}\,\beta_l^{(j)}\big)\{e_j/[]_j\mid j\leq n\}\downarrow_R\big)\cap\mathsf{cond\text{-}st}\big(\big(\sqcup_{\mathsf{c}j\leq n}\,\gamma_m^{(j)}\big)\{e_j/[]_j\mid j\leq n\}\downarrow_R\big)=\emptyset$$

By consequence, the following set is well-nested:

$$\big(\big(\sqcup_{\mathsf{c}j\leq n}\,\beta_i^{(j)}\{e_j/[]_j\}\downarrow_R\big)_{1\leq i\leq i_{\max}},\big(\sqcup_{\mathsf{c}j\leq n}\,\gamma_m^{(j)}\{e_j/[]_j\}\downarrow_R\big)_{1\leq m\leq m_{\max}}\big)$$

Inductive Case If $n\leq n_{\max}\neq 0$, then from (5.11) we get that for every $l\neq i$ in $\{1,\ldots,i_{\max}\}$:

$$\big(\sqcup_{\mathsf{c}j\leq n}\,\beta_i^{(l_j)}\big)\{e_{l_j}/[]_{l_j}\mid j\leq n\}\;=_R \\ \mathsf{if}\ \big(\big(\sqcup_{\mathsf{c}j\leq n}\,\beta_{l_{n+1}}^{(l_j)}\big)\{e_{l_j}/[]_{l_j}\mid j\leq n\}\big)\ \mathsf{then}\ \big(\sqcup_{\mathsf{c}j\leq n+1}\,\beta_i^{(l_j)}\big)\{e_{l_j}/[]_{l_j}\mid j\leq n\}\{\mathsf{true}/[]_{l_{n+1}}\} \\ \mathsf{else}\ \big(\sqcup_{\mathsf{c}j\leq n+1}\,\beta_i^{(l_j)}\big)\{e_{l_j}/[]_{l_j}\mid j\leq n\}\{\mathsf{false}/[]_{l_{n+1}}\}$$

As we did for (5.13), we can show that for every $i\neq l_{n+1}$:

$$\mathsf{leave\text{-}st}(\sqcup_{\mathsf{c}j\leq n}\beta_{l_{n+1}}^{(l_j)}\downarrow_R)\cap\mathsf{cond\text{-}st}\big(\big(\sqcup_{\mathsf{c}j\leq n+1}\,\beta_i^{(l_j)}\big)\{e_{l_j}/[]_{l_j}\mid j\leq n+1\}\downarrow_R\big)=\emptyset$$

Where $e_{l_{n+1}}$ is either true or false. Similarly, for every m:

$$\mathsf{leave\text{-}st}(\sqcup_{\mathsf{c}j\leq n}\beta_{l_{n+1}}^{(l_j)}\downarrow_R)\cap\mathsf{cond\text{-}st}\big(\big(\sqcup_{\mathsf{c}j\leq n+1}\,\gamma_m^{(l_j)}\big)\{e_{l_j}/[]_{l_j}\mid j\leq n+1\}\downarrow_R\big)=\emptyset$$

Moreover, by induction hypothesis, we know that:

$$\big(\big(\sqcup_{\mathsf{c}j\leq n+1}\,\beta_i^{(l_j)}\big)\{e_{l_j}/[]_{l_j}\mid j\leq n+1\}\downarrow_R\big)_i,\big(\big(\sqcup_{\mathsf{c}j\leq n+1}\,\gamma_i^{(l_j)}\big)\{e_{l_j}/[]_{l_j}\mid j\leq n+1\}\downarrow_R\big)_i\big)$$

is well-nested for $e_{l_{n+1}}\equiv\mathsf{true}$ and for $e_{l_{n+1}}\equiv\mathsf{false}$. We deduce that the following set is well nested:

$$\big(\big(\sqcup_{\mathsf{c}j\leq n}\,\beta_i^{(l_j)}\big)\{e_{l_j}/[]_{l_j}\mid j\leq n\}\downarrow_R\big)_i,\big(\big(\sqcup_{\mathsf{c}j\leq n}\,\gamma_i^{(l_j)}\big)\{e_{l_j}/[]_{l_j}\mid j\leq n\}\downarrow_R\big)_i\big)$$

Conclusion Recall that $\beta_i^{(l_0)}\equiv\beta_i^{(0)}\equiv\beta_i$. Hence:

$$\Big(\big\{\beta\downarrow_R\mid\beta\leq_{\mathsf{c}}^{\epsilon,l}(t,P)\big\},\big\{\gamma\downarrow_R\mid\gamma\leq_{\mathsf{l}}^{\epsilon,l}(t,P)\big\}\Big)$$

is the couple of sets:

$$\big(\big(\sqcup_{\mathsf{c}j\leq 0}\,\beta_i^{(l_j)}\big)\downarrow_R\big)_{1\leq i\leq i_{\max}},\big(\big(\sqcup_{\mathsf{c}j\leq 0}\,\gamma_m^{(l_j)}\big)\big\}\downarrow_R\big)_{1\leq m\leq m_{\max}}\big)$$

which is the family of well-nested sets in (5.12), and is therefore well-nested. ■

5.10 Spurious Conditionals and Persistent Leaves

An if-free conditionals b is *spurious* in a term t if, when we normalize t, the conditional b disappears. For example, the conditional b is spurious in $\mathsf{if}\ b\ \mathsf{then}\ 0\ \mathsf{else}\ 0$.

Definition 5.50. An if-free conditional b is said to be *spurious* in a term t if $b \downarrow_R \notin \mathsf{cond\text{-}st}(t \downarrow_R)$.

An if-free term u is *persistent* in a term t if, when we normalize t, the term u *does not* disappear. For example, n_0 is persistent in $\mathsf{if}\ b\ \mathsf{then}\ \mathsf{n}_0\ \mathsf{else\ if}\ b\ \mathsf{then}\ \mathsf{n}_1\ \mathsf{else}\ \mathsf{n}_2$, but n_2 is not.

Definition 5.51. An if-free terms u is said to be *persistent* in a term t if $u \downarrow_R \in \mathsf{cond\text{-}st}(t \downarrow_R)$.

The notion of *spurious set* is related to the notion of *spurious conditional*. A set of position S in a term is a spurious set if we can safely replace in t the terms at positions S by true.

Definition 5.52. A set of positions S is spurious in a term t if it is non-empty and $t[\mathsf{true}/x \mid x \in S] =_R t[\mathsf{false}/x \mid x \in S] =_R t$. A spurious set is *minimal* (resp. *maximal*) if it has not strict spurious subset (resp. overset), and a spurious set is *rooted* if there exists $p \in S$ such that $\forall p' \in S, p \leq p'$ (i.e. is a common ancestor of all positions in S).

Example 5.18. Let $a \equiv \mathsf{eq}(A, 0)$ and $b \equiv \mathsf{eq}(B, 0)$ be two conditionals. Consider the following term t:

$$\begin{array}{l} \mathsf{if}\ b\ \mathsf{then\ if}\ a\ \mathsf{then\ if}\ b\ \mathsf{then}\ T\ \mathsf{else}\ U \\ \qquad\qquad\qquad\qquad \mathsf{else}\ V \\ \qquad\qquad\quad \mathsf{else\ if}\ a\ \mathsf{then}\ T \\ \qquad\qquad\qquad\qquad \mathsf{else\ if}\ a\ \mathsf{then}\ V\ \mathsf{else}\ V \end{array}$$

Then the conditional b is spurious in t, since b is not a subterm of $t \downarrow_R \equiv \mathsf{if}\ a\ \mathsf{then}\ T\ \mathsf{else}\ V$. Moreover the conditional a is a subterm of $t \downarrow_R$, hence is not spurious. Nonetheless, the set of position $S = \{220\}$ is spurious. Indeed we have:

$$\begin{array}{l} \mathsf{if}\ b\ \mathsf{then\ if}\ a\ \mathsf{then\ if}\ b\ \mathsf{then}\ T\ \mathsf{else}\ U \\ \qquad\qquad\qquad\qquad \mathsf{else}\ V \\ \qquad\qquad\quad \mathsf{else\ if}\ a\ \mathsf{then}\ T \\ \qquad\qquad\qquad\qquad \mathsf{else\ if}\ \boxed{a}_{220}\ \mathsf{then}\ V\ \mathsf{else}\ V \end{array} \quad =_R \quad \begin{array}{l} \mathsf{if}\ b\ \mathsf{then\ if}\ a\ \mathsf{then\ if}\ b\ \mathsf{then}\ T\ \mathsf{else}\ U \\ \qquad\qquad\qquad\qquad \mathsf{else}\ V \\ \qquad\qquad\quad \mathsf{else\ if}\ a\ \mathsf{then}\ T \\ \qquad\qquad\qquad\qquad \mathsf{else\ if}\ \boxed{\mathsf{true}}_{220}\ \mathsf{then}\ V\ \mathsf{else}\ V \end{array}$$

$$=_R \quad \begin{array}{l} \mathsf{if}\ b\ \mathsf{then\ if}\ a\ \mathsf{then\ if}\ b\ \mathsf{then}\ T\ \mathsf{else}\ U \\ \qquad\qquad\qquad\qquad \mathsf{else}\ V \\ \qquad\qquad\quad \mathsf{else\ if}\ a\ \mathsf{then}\ T \\ \qquad\qquad\qquad\qquad \mathsf{else\ if}\ \boxed{\mathsf{false}}_{220}\ \mathsf{then}\ V\ \mathsf{else}\ V \end{array} \qquad \square$$

First Objective Let t be a term, and a be a spurious conditional in t such that a is a sub-term of t. If this happens in a proof $P \vdash^{\mathsf{npf}} t \sim t'$, we would like to find a proof-cut elimination getting rid of a. A way of building such a cut elimination is to find a set of positions S which is spurious in both t and t', and such that for every $p \in S$ and $t_{|p} \equiv a$. Then, under some conditions on S, we may be able to obtain a proof $P' \vdash^{\mathsf{npf}} t\{\mathsf{true}/S\} \sim t'\{\mathsf{true}/S\}$. If we can repeat this proof cut sufficiently many times, we may eventually remove all occurrences of a in t.

Our first goal is the following: given a term t and a spurious conditional a in t, and given a set of positions S such that for every $p \in S$ and $t_{|p} \equiv a$, we give sufficient conditions under which S is a spurious set in t. This is done in Section 5.10.1.

Second Objective Consider a proof $P \vdash^{\mathsf{npf}} t \sim t'$, where t is of the form:

$$t \equiv C[(\beta_i)_{i\in I} \diamond (\gamma_j)_{j\in J}]$$

where $(\beta_i)_{i\in I}$ and $(\gamma_j)_{j\in J}$ are $\mathcal{S}$-normalized basic terms and $\mathcal{S}$ is the left CCA_2 trace of P. Remember that we showed in Corollary 5.2.(ii) that for every $j, j' \in J$:

$$\mathsf{leave\text{-}st}(\gamma_j \downarrow_R) \cap \mathsf{leave\text{-}st}(\gamma_{j'} \downarrow_R) \neq \emptyset \quad \text{implies that} \quad \gamma_j \equiv \gamma_{j'}$$

This followed from the fact that given a leaf $u \in \mathsf{leave\text{-}st}(\gamma_j \downarrow_R)$, there exists a unique way of completing u into a $\mathcal{S}$-normalized basic term. Moreover, we will see later that $|\gamma_j|$ is bounded by $|u|$. Assume that we can show that, for every j, $\mathsf{leave\text{-}st}(\gamma_j \downarrow_R)$ contains a persistent term in t, i.e. that $\mathsf{leave\text{-}st}(\gamma_j \downarrow_R) \cap \mathsf{leave\text{-}st}(t \downarrow_R)$ is non-empty. Since $\mathsf{leave\text{-}st}(t \downarrow_R)$ is bounded by the size of the normal form of t, we just bounded the size of the set $\{\gamma_j \mid j \in J\}$.

Therefore, a way of bounding the size of the $\mathcal{S}$-normalized basic terms $(\gamma_j)_{j\in J}$ is to show that they all have a persistent leaf. In other word, we want to prove that we can assume, w.l.o.g., that for every j:

$$\mathsf{leave\text{-}st}(\gamma_j \downarrow_R) \cap \mathsf{leave\text{-}st}(t \downarrow_R) \neq \emptyset$$

This is a key lemma to show decidability. In Section 5.10.2, we give sufficient conditions for this to hold.

5.10.1 Spurious Conditionals to Spurious Sets

We give sufficient conditions under which a set of positions S is spurious in a term t.

Lemma 5.15. *Let $a, \vec{a}, \vec{b}, \vec{c}$ be if-free conditionals in R-normal form. Let s be the context:*

$$\tau[] \equiv B\left[\vec{c} \diamond \left(\vec{w}, \mathit{if}\ C[\vec{b} \diamond \vec{a}, []]\ \mathit{then}\ u\ \mathit{else}\ v\right)\right]$$

Let t be the term $\tau[a]$, and assume that a is spurious in t, and that:

- $a \notin \vec{a} \cup \vec{b} \cup \{\mathit{true}, \mathit{false}\} \cup \mathsf{cond\text{-}st}(u \downarrow_R) \cup \mathsf{cond\text{-}st}(v \downarrow_R)$.
- *$a \notin \rho$ where ρ is the set of conditionals appearing on the path from the root to $\mathit{if}\ C[\vec{b} \diamond \vec{a}, a]\ \mathit{then}\ u\ \mathit{else}\ v$.*

Then $t \equiv \tau[a] =_R \tau[\mathit{true}]$.

Proof. We start by observing that:

$$\begin{aligned}\mathsf{if}\ C[\vec{b} \diamond \vec{a}, a]\ \mathsf{then}\ u\ \mathsf{else}\ v\ =_R\ &\mathsf{if}\ a\ \mathsf{then}\ \mathsf{if}\ C[\vec{b} \diamond \vec{a}, \mathsf{true}]\ \mathsf{then}\ u\ \mathsf{else}\ v\\ &\mathsf{else}\ \mathsf{if}\ C[\vec{b} \diamond \vec{a}, \mathsf{false}]\ \mathsf{then}\ u\ \mathsf{else}\ v\end{aligned}$$

Let $C_u[\vec{b}_u \diamond \vec{t}_u]$ and $C_v[\vec{b}_v \diamond \vec{t}_v]$ be the R-normal forms of u and v. Let C_l, C_r be such that :

$$\begin{aligned}\mathsf{if}\ C[\vec{b} \diamond \vec{a}, \mathsf{true}]\ \mathsf{then}\ u\ \mathsf{else}\ v =_R C_l[\vec{b}_u, \vec{b}_v, \vec{b}, \vec{a} \diamond \vec{t}_u, \vec{t}_v]\\ \mathsf{if}\ C[\vec{b} \diamond \vec{a}, \mathsf{false}]\ \mathsf{then}\ u\ \mathsf{else}\ v =_R C_r[\vec{b}_u, \vec{b}_v, \vec{b}, \vec{a} \diamond \vec{t}_u, \vec{t}_v]\end{aligned}$$

Since $a \notin \mathsf{cond\text{-}st}(u \downarrow_R), \mathsf{cond\text{-}st}(v \downarrow_R)$ we know that $a \notin \vec{b}_u, \vec{b}_v$. Moreover since $a \notin \vec{a} \cup \vec{b}$ we know that $a \notin \vec{b}_u, \vec{b}_v, \vec{b}, \vec{a}$. Therefore:

$$a \notin \mathsf{cond\text{-}st}(C_l[\vec{b}_u, \vec{b}_v, \vec{b}, \vec{a} \diamond \vec{t}_u, \vec{t}_v]) \qquad a \notin \mathsf{cond\text{-}st}(C_r[\vec{b}_u, \vec{b}_v, \vec{b}, \vec{a} \diamond \vec{t}_u, \vec{t}_v])$$

We get rid in C_l and C_r of all the conditionals appearing in ρ. We let $\vec{a}^{\mathsf{l}}$ and $\vec{a}^{\mathsf{r}}$ be such that:

$$\vec{a}^{\mathsf{l}} \subseteq \vec{b}_u, \vec{b}_v, \vec{b}, \vec{a} \backslash \rho \qquad \vec{a}^{\mathsf{r}} \subseteq \vec{b}_u, \vec{b}_v, \vec{b}, \vec{a} \backslash \rho$$

and C'_l, C'_r such that:

$$B\left[\vec{c} \diamond \left(\vec{w}, C_l[\vec{b}_u, \vec{b}_v, \vec{b}, \vec{a} \diamond \vec{t}_u, \vec{t}_v]\right)\right] =_R B\left[\vec{c} \diamond \left(\vec{w}, C'_l[\vec{a}^{\mathsf{l}} \diamond \vec{t}_u, \vec{t}_v]\right)\right] \tag{5.14}$$

$$B\left[\vec{c} \diamond \left(\vec{w}, C_r[\vec{b}_u, \vec{b}_v, \vec{b}, \vec{a} \diamond \vec{t}_u, \vec{t}_v]\right)\right] =_R B\left[\vec{c} \diamond \left(\vec{w}, C'_r[\vec{a}^{\mathsf{r}} \diamond \vec{t}_u, \vec{t}_v]\right)\right] \tag{5.15}$$

We know that $a \notin \vec{a}^{\mathsf{l}}$ and $a \notin \vec{a}^{\mathsf{r}}$.

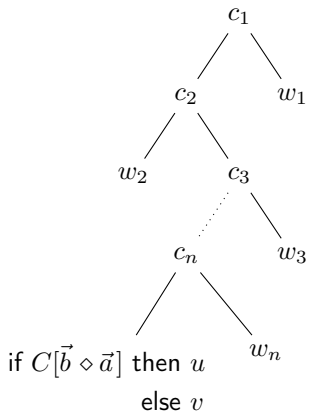
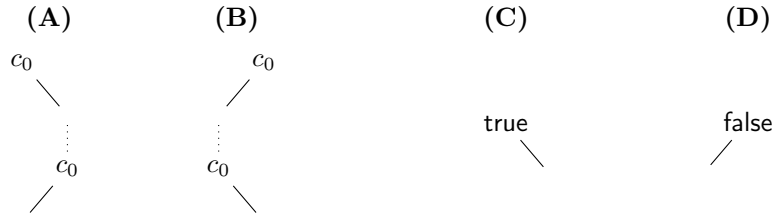


Figure 5.10: Example of if-context B'

Case 1 If there exists $c_0 \in \vec{c}$ such that the path ρ from the root of t to if $C[\vec{b} \diamond \vec{a}]$ then u else v contains one of the following shapes, where solid edges represent one element of the path ρ, and dotted edges represent a summary of a part of the path ρ.



In these four cases, the result is easy to show, since we can do any rewriting we want. For example, in case **(A)**, we use the fact that:

$$\text{if } x \text{ then } y \text{ else } (\text{if } x \text{ then } \mathbf{v} \text{ else } z) \to_R^* \text{if } x \text{ then } y \text{ else } (\text{if } x \text{ then } \mathbf{v}' \text{ else } z) \qquad (\text{ for all term } \mathbf{v}')$$

to rewrite (if $C[\vec{b} \diamond \vec{a}, a]$ then u else v) into the term (if a then if $C[\vec{b} \diamond \vec{a}, \mathsf{true}]$ then u else v else).

Case 2 Let $s[]$ be such that $t \equiv s[\text{if } C[\vec{b} \diamond \vec{a}] \text{ then } u \text{ else } v]$. If none of the shapes of **Case 1** occurs, then we know that there exists B' and $\vec{\omega}$ such that $s[] =_R B' \left[\vec{c} \diamond (\vec{\omega}, [])\right]$ and the path ρ' from the root of s to [] is a subset of ρ and does not contain duplicates, true and false. The existence of such a B' is proved by induction on the number of duplicate conditionals, true and false occurring on ρ': indeed since the shape (A) and (B) (resp. (C) and (D)) are forbidden in ρ, we know that if we have a duplicate (resp. true or false) then we can always rewrite B such that the hole containing s does not disappear.

Let $\rho' = c_1, \ldots, c_n$, we take B' minimal, i.e. only a branch $c_1, \ldots, c_n$. We give an example of such an if-context in Figure 5.10.

Wet let $\vec{\omega} = w_1, \ldots, w_n$, and we have:

$$s =_R B' \left[c_1, \ldots, c_n \diamond w_1, \ldots, w_n, []\right]$$

We let $\succ_u$ be a total ordering on if-free conditional in R-normal form such that the $n+1$ maximum elements are $c_1 \succ_u \cdots \succ c_n \succ_u a$. For every y, we let $W_i[\vec{d}_i \diamond \vec{e}_i]$ be the $R_{\succ_u}$-normal form of w_i. Then:

$$s =_R B' \Big[c_1, \ldots, c_n \diamond \big(W_i[\vec{d}_i \diamond \vec{e}_i]\big)_{i \leq n}, []\Big]$$

We get rid of any occurrence of $c_1, \ldots, c_n$ in $(\vec{d}_i)_i$. For every i, we let $W_i'[\vec{d}_i' \diamond \vec{e}_i']$ be terms in R-normal form such that $\vec{d}_i' \cap \{c_j \mid j \leq i\} = \emptyset$ and:

$$s =_R B' \Big[c_1, \ldots, c_n \diamond \big(W_i'[\vec{d}_i' \diamond \vec{e}_i']\big)_{i \leq n}, []\Big]$$

Using (5.14) and (5.15) we get:

$$t =_R B'\left[c_1,\dots,c_n \diamond \left(W'_i[\vec{d}'_i \diamond \vec{e}'_i]\right)_{i\leq n}, \begin{array}{r}\mathsf{if}\ a\ \mathsf{then}\ C'_l[\vec{a}^{\mathsf{l}} \diamond \vec{t}_u, \vec{t}_v] \\ \mathsf{else}\ C'_r[\vec{a}^{\mathsf{r}} \diamond \vec{t}_u, \vec{t}_v]\end{array}\right]$$

It is then quite easy to show by induction on the length of the reduction sequence that there exists a sequence $1 \leq i_1 < \cdots < i_k \leq n$ and an if-context B'' such that:

$$\begin{aligned} t\downarrow_{R_{\succ_u}} &\equiv \left(B'\left[c_1,\dots,c_n \diamond \left(W'_i[\vec{d}'_i \diamond \vec{e}'_i]\right)_{i\leq n}, \begin{array}{r}\mathsf{if}\ a\ \mathsf{then}\ C'_l[\vec{a}^{\mathsf{l}} \diamond \vec{t}_u, \vec{t}_v] \\ \mathsf{else}\ C'_r[\vec{a}^{\mathsf{r}} \diamond \vec{t}_u, \vec{t}_v]\end{array}\right]\right)\downarrow_{R_{\succ_u}} \\ &\equiv B''\left[c_{i_1},\dots,c_{i_k} \diamond \left(W'_{i_j}[\vec{d}'_{i_j} \diamond \vec{e}'_{i_j}]\right)_{j\leq k}, \begin{pmatrix}\mathsf{if}\ a\ \mathsf{then}\ C'_l[\vec{a}^{\mathsf{l}} \diamond \vec{t}_u, \vec{t}_v] \\ \mathsf{else}\ C'_r[\vec{a}^{\mathsf{r}} \diamond \vec{t}_u, \vec{t}_v]\end{pmatrix}\downarrow_{R_{\succ_u}}\right] \end{aligned}$$

We deduce from this that a is spurious in:

$$\mathsf{if}\ a\ \mathsf{then}\ C'_l[\vec{a}^{\mathsf{l}} \diamond \vec{t}_u, \vec{t}_v]\ \mathsf{else}\ C'_r[\vec{a}^{\mathsf{r}} \diamond \vec{t}_u, \vec{t}_v]$$

Since a will stay the top-most conditional in the R-normal form of this term (because of the order $\succ_u$ we chose), and since $a \neq \mathsf{true}$ $a \neq \mathsf{false}$ and $a \notin \vec{a}^{\mathsf{l}}, \vec{a}^{\mathsf{r}}$, there is only one rule that can be applied: $\mathsf{if}\ a\ \mathsf{then}\ x\ \mathsf{else}\ x \to x$. Consequently:

$$C'_l[\vec{a}^{\mathsf{l}} \diamond \vec{t}_u, \vec{t}_v] =_R C'_r[\vec{a}^{\mathsf{r}} \diamond \vec{t}_u, \vec{t}_v]$$

Hence:

$$\begin{aligned} t &=_R B'\left[c_1,\dots,c_n \diamond \left(W'_i[\vec{d}'_i \diamond \vec{e}'_i]\right)_{i\leq n}, C'_l[\vec{a}^{\mathsf{l}} \diamond \vec{t}_u, \vec{t}_v]\right] \\ &=_R s\left[C'_l[\vec{a}^{\mathsf{l}} \diamond \vec{t}_u, \vec{t}_v]\right] \\ &=_R B\left[\vec{c} \diamond \left(\vec{w}, C'_l[\vec{a}^{\mathsf{l}} \diamond \vec{t}_u, \vec{t}_v]\right)\right] \end{aligned}$$

Hence using (5.14) we get:

$$t =_R B\left[\vec{c} \diamond \left(\vec{w}, C_l[\vec{b}_u, \vec{b}_v, \vec{b}, \vec{a} \diamond \vec{t}_u, \vec{t}_v]\right)\right] =_R B\left[\vec{c} \diamond \left(\vec{w}, \mathsf{if}\ C[\vec{b} \diamond \vec{a}, \mathsf{true}]\ \mathsf{then}\ u\ \mathsf{else}\ v\right)\right]$$ ■

5.10.2 Persistent Terms

Let a be a conditional and $s[]$ be a context. The following proposition give sufficient conditions under which the persistent terms of $s[a]$ are exactly the persistent terms of $s[\mathsf{true}]$ and $s[\mathsf{false}]$.

Proposition 5.15. *Let $a, (\vec{a}_i, \vec{b}_i)_i, (\vec{c}_j, \vec{t}_j)_j$ be if-free terms in R-normal form such that for every i, $a \notin \vec{a}_i \cup \vec{b}_i \cup \vec{c}_j$, and let $s[]$ be a context such that:*

$$s[] \equiv B\left[\left(C_i[\vec{a}_i, [] \diamond \vec{b}_i, []]\right)_i \diamond \left(D_j[\vec{c}_j, [] \diamond \vec{t}_j]\right)_j\right]$$

Then $\mathsf{leave\text{-}st}(s[a]\downarrow_R) = \mathsf{leave\text{-}st}(s[\mathit{true}]\downarrow_R) \cup \mathsf{leave\text{-}st}(s[\mathit{false}]\downarrow_R)$.

Proof. We know that $s[a] =_R \mathsf{if}\ a\ \mathsf{then}\ s[\mathsf{true}]\ \mathsf{else}\ s[\mathsf{false}]$. Let $\succ_u$ be a total order on if-free conditionals in R-normal form such that a is minimal. It is straightforward to check that:

$$\begin{aligned} s[a]\downarrow_{R^{\succ_u}} &\equiv (\mathsf{if}\ a\ \mathsf{then}\ s[\mathsf{true}]\ \mathsf{else}\ s[\mathsf{false}])\downarrow_{R^{\succ_u}} \\ &\equiv \begin{cases} (s[\mathsf{true}])\downarrow_{R^{\succ_u}} & \text{if } s[\mathsf{true}] =_{R^{\succ_u}} s[\mathsf{false}] \\ \mathsf{if}\ a\ \mathsf{then}\ (s[\mathsf{true}])\downarrow_{R^{\succ_u}}\ \mathsf{else}\ (s[\mathsf{false}])\downarrow_{R^{\succ_u}} & \text{otherwise} \end{cases} \end{aligned}$$

Therefore:

$$\mathsf{leave\text{-}st}(s[a]\downarrow_{R^{\succ_u}}) = \mathsf{leave\text{-}st}(s[\mathsf{true}]\downarrow_{R^{\succ_u}}) \cup \mathsf{leave\text{-}st}(s[\mathsf{false}]\downarrow_{R^{\succ_u}})$$

The wanted result follows from Proposition 5.1. ■

We show the following technical proposition, that we use later in this section. Given a conditional a and two terms t_l and t_r, we give sufficient conditions under which a persistent term in t_l or t_r is a persistent term in $\mathsf{if}\ a\ \mathsf{then}\ t_l\ \mathsf{else}\ t_r$.

Proposition 5.16 (Persistent Term Lifting)**.** *Consider the terms:*

$$C[\vec{a} \diamond \vec{b}] \qquad t_l \equiv B^l\left[\left(C_i^l[\vec{a}_i^l \diamond \vec{b}_i^l]\right)_i \diamond \left(D_j^l[\vec{c}_j^l \diamond \vec{t}_j^l]\right)_j\right] \qquad t_r \equiv B^r\left[\left(C_i^r[\vec{a}_i^r \diamond \vec{b}_i^r]\right)_i \diamond \left(D_j^r[\vec{c}_j^r \diamond \vec{t}_j^r]\right)_j\right]$$

where:

- *For every $x \in \{l, r\}$, i and j, the terms $\vec{a}_i^x, \vec{b}_i^x, \vec{c}_j^x, \vec{t}_j^x$ are if-free and in R-normal form.*
- *$\vec{a}, \vec{b}$ are if-free, in R-normal form and $(\vec{a} \cup \vec{b}) \cap \{\mathsf{true}, \mathsf{false}\} = \emptyset$.*
- *$\vec{b} \cap (\bigcup_{x\in\{l,r\},i} \vec{a}_i^x, \vec{b}_i^x) = \emptyset$ and $\vec{b} \cap (\bigcup_{x\in\{l,r\},j} \vec{c}_i^x) = \emptyset$.*
- *$\vec{a} \cap \vec{b} = \emptyset$.*

Then:

$$\mathsf{leave\text{-}st}(t_l \downarrow_R) \cup \mathsf{leave\text{-}st}(t_r \downarrow_R) \subseteq \mathsf{leave\text{-}st}\left((\mathit{if}\ C[\vec{a} \diamond \vec{b}]\ \mathit{then}\ t_l\ \mathit{else}\ t_r) \downarrow_R\right)$$

Proof. We prove this by induction on $|\vec{a}|$.

Base Case If $|\vec{a}| = 0$ then $C[\vec{a} \diamond \vec{b}] \equiv b$, where b is if-free. Let $\succ_u$ be any total order on if-free conditionals in R-normal form such that b is minimal. We then let $D_l[\vec{a}_l \diamond \vec{t}_l]$ and $D_r[\vec{a}_r \diamond \vec{t}_r]$ be the $R_{\succ_u}$-normal form of t_l and t_r. By Proposition 5.1, we know that:

$$\mathsf{leave\text{-}st}(t_l \downarrow_R) = \mathsf{leave\text{-}st}(t_l \downarrow_{R_{\succ_u}}) = \mathsf{leave\text{-}st}\left(\left(D_l[\vec{a}_l \diamond \vec{t}_l]\right) \downarrow_{R_{\succ_u}}\right) \tag{5.16}$$

Using the fact that $(\vec{a}_i^l, \vec{b}_i^l)_i$ and $\vec{c}_j^l, \vec{t}_j^l)_j$ are if-free and in R-normal form, it is simple to show by induction on the length of the reduction that $\vec{a}_l \subseteq (\vec{a}_i^l, \vec{b}_i^l)_i, (\vec{c}_j^l)_j$. Since $b \notin (\bigcup_{x\in\{l,r\},i} \vec{a}_i^x, \vec{b}_i^x)$ and $b \notin (\bigcup_{x\in\{l,r\},j}, \vec{c}_j^x)$, this shows that $b \notin \vec{a}_l$. Similarly $\vec{a}_r \subseteq (\vec{a}_i^r, \vec{b}_i^r)_i, (\vec{c}_j^r)_j$ and $b \notin \vec{a}_r$.

$$(\mathsf{if}\ b\ \mathsf{then}\ t_l\ \mathsf{else}\ t_r) \downarrow_{R^{\succ_u}} \equiv \left(\mathsf{if}\ b\ \mathsf{then}\ D_l[\vec{a}_l \diamond \vec{t}_l]\ \mathsf{else}\ D_r[\vec{a}_r \diamond \vec{t}_r]\right) \downarrow_{R^{\succ_u}}$$

Since b is and if-free conditional in R-normal form minimal for $\succ_u$, since $D_l[\vec{a}_l \diamond \vec{t}_l]$ and $D_r[\vec{a}_r \diamond \vec{t}_r]$ are in $R_{\succ_u}$-normal form, since $b \notin \{\mathsf{true}, \mathsf{false}\}$ and since $b \notin \vec{a}_l \cup \vec{a}_r$, there is only one rule that may be applicable: $\mathsf{if}\ b\ \mathsf{then}\ x\ \mathsf{else}\ x \to x$. Therefore:

$$(\mathsf{if}\ b\ \mathsf{then}\ t_l\ \mathsf{else}\ t_r) \downarrow_{R^{\succ_u}} \equiv \begin{cases} t_l \downarrow_{R^{\succ_u}} & \text{if } t_l =_{R^{\succ_u}} t_r \\ \mathsf{if}\ b\ \mathsf{then}\ t_l \downarrow_{R^{\succ_u}}\ \mathsf{else}\ t_r \downarrow_{R^{\succ_u}} & \text{otherwise} \end{cases}$$

Which shows the wanted result.

Inductive Case Assume that the result holds for m, and consider $\vec{a}, a$ of length $m+1$. First:

$$\begin{aligned}\mathsf{if}\ C[\vec{a}, a \diamond \vec{b}]\ \mathsf{then}\ t_l\ \mathsf{else}\ t_r =_R \mathsf{if}\ a\ \mathsf{then}\ \mathsf{if}\ C[\vec{a}, \mathsf{true} \diamond \vec{b}]\ \mathsf{then}\ t_l\ \mathsf{else}\ t_r \\ \mathsf{else}\ \mathsf{if}\ C[\vec{a}, \mathsf{false} \diamond \vec{b}]\ \mathsf{then}\ t_l\ \mathsf{else}\ t_r\end{aligned}$$

Let $s_l[]$ be a context such that $s_l[a] \equiv t_l$ and $a \notin \mathsf{cond\text{-}st}(s_l[] \downarrow_R)$. Similarly, let $s_r[]$ be such that $s_r[a] \equiv t_r$ and $a \notin \mathsf{cond\text{-}st}(s_r[] \downarrow_R)$. We are going to rewrite the then branch to replace any occurrence of a by true. Similarly, we rewrite the else branch to replace any occurrence of a by false.

Moreover, we get rid of true and false in $C[\vec{a}, \mathsf{true} \diamond \vec{b}]$ and $C[\vec{a}, \mathsf{false} \diamond \vec{b}]$. Let $C'[\vec{a}' \diamond \vec{b}']$ and $C''[\vec{a}'' \diamond \vec{b}'']$ be such that:

$$C[\vec{a} \diamond \vec{b}] =_R \mathsf{if}\ a\ \mathsf{then}\ C'[\vec{a}' \diamond \vec{b}']\ \mathsf{else}\ C''[\vec{a}'' \diamond \vec{b}'']$$

with $\vec{a}' \cup \vec{a}'' \subseteq \vec{a} \backslash \{a\}$ and $\vec{b}' \cup \vec{b}'' \subseteq \vec{b} \backslash \{a\}$. Then:

$$\mathsf{if}\ C[\vec{a}, \mathsf{true} \diamond \vec{b}]\ \mathsf{then}\ t_l\ \mathsf{else}\ t_r =_R \boxed{\mathsf{if}\ C'[\vec{a}' \diamond \vec{b}']\ \mathsf{then}\ s_l[\mathsf{true}]\ \mathsf{else}\ s_r[\mathsf{true}]}$$

$$\mathsf{if}\ C[\vec{a}, \mathsf{false} \diamond \vec{b}]\ \mathsf{then}\ t_l\ \mathsf{else}\ t_r =_R \boxed{\mathsf{if}\ C''[\vec{a}'' \diamond \vec{b}'']\ \mathsf{then}\ s_l[\mathsf{false}]\ \mathsf{else}\ s_r[\mathsf{false}]}$$

We start by checking that the induction hypothesis on the red framed term. The first condition is trivial, we check the other:

- Since $\vec{a}' \subseteq \vec{a}$, $\vec{b}' \subseteq \vec{b}$ and $(\vec{a} \cup \vec{b}) \cap \{\mathsf{true}, \mathsf{false}\} = \emptyset$, we know that $(\vec{a}' \cup \vec{b}') \cap \{\mathsf{true}, \mathsf{false}\} = \emptyset$.
- The term $s_l[a]$ is obtained from t_l by replacing every occurrence of a by true. Hence, since $\mathsf{true} \notin \vec{b}$, $\vec{b}' \subseteq \vec{b}$ and:

  $$\vec{b} \cap (\bigcup_{x \in \{l,r\}, i} \vec{a}_i^x, \vec{b}_i^x) = \emptyset \qquad \vec{b} \cap (\bigcup_{x \in \{l,r\}, j} \vec{c}_i^x) = \emptyset$$

  We know that the third condition holds.
- Since $\vec{a}' \subseteq \vec{a}$, $\vec{b}' \subseteq \vec{b}$ and $\vec{a} \cap \vec{b} = \emptyset$, we know that $\vec{a}' \cap \vec{b}' = \emptyset$.

By applying the induction hypothesis, we deduce that:

$$\mathsf{leave\text{-}st}(s_l[\mathsf{true}] \downarrow_R) \cup \mathsf{leave\text{-}st}(s_r[\mathsf{true}] \downarrow_R) \subseteq \boxed{\mathsf{leave\text{-}st}(\mathsf{if}\ C'[\vec{a}' \diamond \vec{b}']\ \mathsf{then}\ s_l[\mathsf{true}]\ \mathsf{else}\ s_r[\mathsf{true}] \downarrow_R)}$$

Similarly, by applying the induction hypothesis on the rewriting of the blue framed term, we get:

$$\mathsf{leave\text{-}st}(s_l[\mathsf{false}] \downarrow_R) \cup \mathsf{leave\text{-}st}(s_r[\mathsf{false}] \downarrow_R) \subseteq \boxed{\mathsf{leave\text{-}st}(\mathsf{if}\ C''[\vec{a}'' \diamond \vec{b}'']\ \mathsf{then}\ s_l[\mathsf{false}]\ \mathsf{else}\ s_r[\mathsf{false}] \downarrow_R)}$$

Finally, we apply again the induction hypothesis (with $m = 0$) to the term u below:

$$u \equiv \mathsf{if}\ a\ \mathsf{then}\ \boxed{\mathsf{leave\text{-}st}(\mathsf{if}\ C'[\vec{a}' \diamond \vec{b}']\ \mathsf{then}\ s_l[\mathsf{true}]\ \mathsf{else}\ s_r[\mathsf{true}] \downarrow_R)}$$
$$\mathsf{else}\ \boxed{\mathsf{leave\text{-}st}(\mathsf{if}\ C''[\vec{a}'' \diamond \vec{b}'']\ \mathsf{then}\ s_l[\mathsf{false}]\ \mathsf{else}\ s_r[\mathsf{false}] \downarrow_R)}$$

We get that:

$$\mathsf{leave\text{-}st}(u \downarrow_R) \supseteq \mathsf{leave\text{-}st}\left(\boxed{\mathsf{leave\text{-}st}(\mathsf{if}\ C'[\vec{a}' \diamond \vec{b}']\ \mathsf{then}\ s_l[\mathsf{true}]\ \mathsf{else}\ s_r[\mathsf{true}] \downarrow_R)} \downarrow_R\right)$$
$$\cup\ \mathsf{leave\text{-}st}\left(\boxed{\mathsf{leave\text{-}st}(\mathsf{if}\ C''[\vec{a}'' \diamond \vec{b}'']\ \mathsf{then}\ s_l[\mathsf{false}]\ \mathsf{else}\ s_r[\mathsf{false}] \downarrow_R)} \downarrow_R\right)$$

By applying Proposition 5.15 twice, we know that:

$$\mathsf{leave\text{-}st}(t_l \downarrow_R) \cup \mathsf{leave\text{-}st}(t_r \downarrow_R) =$$
$$\mathsf{leave\text{-}st}(s_l[\mathsf{true}] \downarrow_R) \cup \mathsf{leave\text{-}st}(s_r[\mathsf{true}] \downarrow_R) \cup \mathsf{leave\text{-}st}(s_l[\mathsf{false}] \downarrow_R) \cup \mathsf{leave\text{-}st}(s_r[\mathsf{false}] \downarrow_R)$$

Hence we deduce that:

$$\mathsf{leave\text{-}st}(t_l \downarrow_R) \cup \mathsf{leave\text{-}st}(t_r \downarrow_R) \subseteq \mathsf{leave\text{-}st}(u \downarrow_R) = \mathsf{leave\text{-}st}(\mathsf{if}\ C[\vec{a}, a \diamond \vec{b}]\ \mathsf{then}\ t_l\ \mathsf{else}\ t_r \downarrow_R) \qquad \blacksquare$$

We are now ready to prove the main lemma of this section, which, under some conditions, shows that all leaf term γ of a term t has a persistent leaf.

Lemma 5.16. *Let s be a term of the form:*

$$s \equiv A\left[\vec{d} \diamond \left(B_l\left[\left(\beta_{i,l}\right)_i \diamond \left(\gamma_{j,l}\right)_j\right]\right)_l\right]$$

such that:

(i) *$\vec{d}$ are if-free and in R-normal form, and for every i, l, $\mathsf{cond\text{-}st}(\beta_{i,l} \downarrow_R) \cap \mathsf{leave\text{-}st}(\beta_{j,l} \downarrow_R) = \emptyset$.*

(ii) $\left(\vec{d} \cup \bigcup_{i,l} \mathsf{leave\text{-}st}(\beta_{i,l} \downarrow_R)\right) \cap \{\mathit{true}, \mathit{false}\} = \emptyset$.

(iii) *For every positions* $p < p'$ *in* $A[\_ \diamond (B_l)_l]$ *such that* $s_{|p} \equiv \zeta$ *and* $s_{|p'} \equiv \zeta'$, *we have:*

$$\mathsf{leave\text{-}st}(\zeta \downarrow_R) \cap \mathsf{leave\text{-}st}(\zeta' \downarrow_R) = \emptyset$$

(iv) *For every* l, i, j, $\mathsf{leave\text{-}st}(\beta_{i,l} \downarrow_R) \cap \mathsf{leave\text{-}st}(\beta_{j,l} \downarrow_R) \neq \emptyset$ *implies that* $\beta_{i,l} \equiv \beta_{j,l}$.

(v) *For every* l, *the following couple of sets is well-nested:*

$$\left(\{\beta_{i,l} \downarrow_R |\ i\}, \{\gamma_{j,l} \downarrow_R |\ j\}\right)$$

then for every l, j, $\gamma_{j,l}$ *contains a persistent term in* s, *i.e.* $\mathsf{leave\text{-}st}(\gamma_{j,l} \downarrow_R) \cap \mathsf{leave\text{-}st}(s \downarrow_R) \neq \emptyset$.

Proof. We start by showing that the property holds when $\vec{d} = \emptyset$ and $A \equiv []$. We deal with the general case afterward.

Part 1 For all i, j, we let $C_i[]$, $D_j[]$ be if-contexts and $\vec{a}_i$, $\vec{b}_i, \vec{c}_j$, $\vec{t}_j$ be if-free terms in R-normal form such that:

$$\vec{a}_i \equiv \mathsf{cond\text{-}st}(\beta_i \downarrow_R) \qquad \vec{b}_i \equiv \mathsf{leave\text{-}st}(\beta_i \downarrow_R) \qquad \vec{c}_i \equiv \mathsf{cond\text{-}st}(\gamma_j \downarrow_R) \qquad \vec{t}_i \equiv \mathsf{leave\text{-}st}(\gamma_j \downarrow_R)$$

$$\beta_i \downarrow_R \equiv\ C_i[\vec{a}_i \diamond \vec{b}_i] \qquad\qquad \gamma_j \downarrow_R \equiv\ D_j[\vec{c}_j \diamond \vec{t}_j]$$

We know that:

$$s =_R B\left[\left(C_i[\vec{a}_i \diamond \vec{b}_i]\right)_i \diamond \left(D_j[\vec{c}_j \diamond \vec{t}_j]\right)_j\right]$$

satisfying conditions (i) to (v). We prove the proposition by structural induction on $B[]$.

Part 1: Base Case The base case is simple. It suffices to notice that since $\vec{c}, \vec{t}$ are if-free and in R-normal form:

$$\mathsf{leave\text{-}st}(s \downarrow_R) = \mathsf{leave\text{-}st}(D[\vec{c} \diamond \vec{t}] \downarrow_R) \subseteq \vec{t}$$

Part 1: Inductive Case Consider:

$$s \equiv \mathsf{if}\ C_0[\vec{a}_0 \diamond \vec{b}_0]\ \mathsf{then}\ B^l\left[\left(C_i[\vec{a}_i \diamond \vec{b}_i]\right)_{i \in I^l} \diamond \left(D_j[\vec{c}_j \diamond \vec{t}_j]\right)_{j \in J^l}\right]$$
$$\mathsf{else}\ B^r\left[\left(C_i[\vec{a}_i \diamond \vec{b}_i]\right)_{i \in I^r} \diamond \left(D_j[\vec{c}_j \diamond \vec{t}_j]\right)_{j \in J^r}\right]$$

Using the well-nested hypothesis, for every $j \in I^l \cup I^r$, there exist two if-context C'_j, C''_j such that:

$$C_j[\vec{a}_j \diamond \vec{b}_j] =_R \mathsf{if}\ C_0[\vec{a}_0 \diamond \vec{b}_0]\ \mathsf{then}\ C'_j[\vec{a}'_j \diamond \vec{b}'_j]\ \mathsf{else}\ C''_j[\vec{a}''_j \diamond \vec{b}''_j]$$

where $\vec{a}'_j, \vec{a}''_j \subseteq \vec{a}_j \backslash \vec{b}_0$ and $\vec{b}'_j, \vec{b}''_j \subseteq \vec{b}_j$. Similarly, for every $j \in J^l \cup J^r$, there exist D'_j, D''_j such that:

$$D_j[\vec{c}_j \diamond \vec{t}_j] =_R \mathsf{if}\ C_0[\vec{a}_0 \diamond \vec{b}_0]\ \mathsf{then}\ D'_j[\vec{c}'_j \diamond \vec{t}'_j]\ \mathsf{else}\ C''_j[\vec{c}''_j \diamond \vec{t}''_j]$$

where $\vec{c}'_j, \vec{c}''_j \subseteq \vec{a}_j \backslash \vec{b}_0$ and $\vec{t}'_j, \vec{t}''_j \subseteq \vec{t}_j$. Then:

$$s \equiv \mathsf{if}\ C_0[\vec{a}_0 \diamond \vec{b}_0]\ \mathsf{then}\ \boxed{B^l\left[\left(C'_i[\vec{a}'_i \diamond \vec{b}'_i]\right)_{i \in I^l} \diamond \left(D'_j[\vec{c}'_j \diamond \vec{t}'_j]\right)_{j \in J^l}\right]}\ s_{\mathsf{true}}$$
$$\mathsf{else}\ \boxed{B^r\left[\left(C''_i[\vec{a}''_i \diamond \vec{b}''_i]\right)_{i \in I^r} \diamond \left(D''_j[\vec{c}''_j \diamond \vec{t}''_j]\right)_{j \in J^r}\right]}\ s_{\mathsf{false}}$$

We want to show that for all $j \in J^l \cup J^r$, $\exists t \in \vec{t}_j . t \in \mathsf{leave\text{-}st}(s \downarrow_R)$. Let $j \in J^l$ (the proof for $j \in J^r$ is similar). We are going to apply the induction hypothesis to s_{true} (for $j \in J^r$, we apply the induction hypothesis to s_{false}). Lets check that the premises hold for s_{true}:

- (i) and (ii) trivially hold.
- For (iii), we use the fact that we know that the property holds in s for every positions $\epsilon < p < p'$ in if $[]$ then B^l else B^r, and the fact that for every $i \in I^l \cup I^r$, $\vec{b'}^i \subseteq \vec{b}^i$.
- Checking that (iv) holds is straightforward. Assume that there exists $i, j \in I^l$ such that $\vec{b}'_i \cap \vec{b}'_j \neq \emptyset$. Since $\vec{b}'_i \subseteq \vec{b}_i$ and $\vec{b}'_j \subseteq \vec{b}_j$ we know that $\vec{b}_i \cap \vec{b}_j \neq \emptyset$. Therefore $C_i[\vec{a}_i \diamond \vec{b}_i] \equiv C_j[\vec{a}_j \diamond \vec{b}_j]$. Hence:

$$C'_i[\vec{a}'_i \diamond \vec{b}'_i] \equiv C'_j[\vec{a}'_j \diamond \vec{b}'_j] \qquad\qquad C''_i[\vec{a}''_i \diamond \vec{b}''_i] \equiv C''_j[\vec{a}''_j \diamond \vec{b}''_j]$$

- Using the inductive property of well-nested couples (item (iv)) we know that the following couple of sets is well-nested:

$$\left(\left\{C'_i[\vec{a}'_i \diamond \vec{b}'_i] \mid i \in I^l \cup I^r \cup \{0\}\right\}, \left\{D'_j[\vec{c}'_j \diamond \vec{t}'_j] \mid j \in J^l \cup J^r\right\}_j\right)$$

  Since, for every $(\mathcal{C}, \mathcal{D}), (\mathcal{C}', \mathcal{D}')$, if $(\mathcal{C}, \mathcal{D})$ is well-nested and $\mathcal{C}' \subseteq \mathcal{C} \wedge \mathcal{D}' \subseteq \mathcal{D}$ then $(\mathcal{C}', \mathcal{D}')$ is well-nested, we know that the following couple of sets is well-nested:

$$\left(\left\{C'_i[\vec{a}'_i \diamond \vec{b}'_i] \mid i \in I^l \cup \{0\}\right\}, \left\{D'_j[\vec{c}'_j \diamond \vec{t}'_j] \mid j \in J^l\right\}_j\right)$$

We can apply the induction hypothesis to s_{true}, which shows that for all $j \in J^l$, there exists $t \in \vec{t}'_j$ such that $t \in \mathsf{leave\text{-}st}(s_{\mathsf{true}} \downarrow_R)$. To conclude, we have to lift this to $\mathsf{leave\text{-}st}(s \downarrow_R)$.

Let $S = I^l \cup I^r \cup \{0\} \cup J^l \cup J^r$, and $S' = S \backslash \{0\}$. We apply Proposition 5.16 to show that $t \in \mathsf{leave\text{-}st}(s \downarrow_R)$. The only difficulty lies in showing that:

$$\vec{b}_0 \cap \left(\bigcup_{i \in S'} \vec{a}'_i, \vec{a}''_i, \vec{b}'_i, \vec{b}''_i, \vec{c}'_i, \vec{c}''_i\right) = \emptyset$$

We know that $b_0 \cap \left(\bigcup_{i \in S'} \vec{a}'_i, \vec{a}''_i, \vec{c}'_i, \vec{c}''_i\right) = \emptyset$ (since $\vec{a}'_i \subseteq \vec{a}_i \backslash \vec{b}_0, \dots$), so it only remains to show that:

$$\vec{b}_0 \cap \bigcup_{i \in S'} \vec{b}'_i, \vec{b}''_i = \emptyset \tag{5.17}$$

For every $i \in S'$, we know that $\vec{b}'_i \subseteq \vec{b}_i$ and $\vec{b}''_i \subseteq \vec{b}_i$. Hence, if $\vec{b}_0 \cap \vec{b}'_i \neq \emptyset$ or $\vec{b}_0 \cap \vec{b}''_i \neq \emptyset$ then $\vec{b}_i \cap \vec{b}_0 \neq \emptyset$. Since $C_0[]$ is at the root of s, we know using (iii) that $\vec{b}_i \cap \vec{b}_0 = \emptyset$. Hence (5.17) holds.

Part 2 For the general case, we just observe that we can take:

$$B[] \equiv A[([]_d)_{d \in \vec{d}} \diamond (B_l[])_l]$$

We only need to check that the property (i)-(v) are verified for $B[]$. Properties (i)-(iv) are straightforward. For (v), we only observe that, since $\vec{d}$ are if-free and in R-normal form, if $(\mathcal{C}, \mathcal{D})$ is well-nested then $(\mathcal{C} \cup \vec{d}, \mathcal{D})$ is well-nested. ■

5.11 Proof Cut Elimination

Consider a proof $P \vdash^{\mathsf{npf}} t \sim t'$. Lemma 5.16 shows that, under some conditions, any normalized basic term $\gamma \leq_{\mathsf{l}}^{\epsilon,l} (t, P)$ has a persistent leaf in t, i.e. $\mathsf{leave\text{-}st}(\gamma \downarrow_R) \cap \mathsf{leave\text{-}st}(t \downarrow_R) \neq \emptyset$. To apply this lemma, we need to have a proof P satisfying the hypothesis of Lemma 5.16. We give simplified version of these conditions below:

(i) for every $\beta, \beta' \leq_{\mathsf{c}}^{\epsilon,l} (t, P)$, we have $\mathsf{cond\text{-}st}(\beta \downarrow_R) \cap \mathsf{leave\text{-}st}(\beta \downarrow_R) = \emptyset$.

(ii) $\left(\bigcup_{\beta \leq_{\mathsf{c}}^{\epsilon,l}(t,P)} \mathsf{leave\text{-}st}(\beta \downarrow_R)\right) \cap \{\mathsf{true}, \mathsf{false}\} = \emptyset$.

(iii) For every $\beta, \beta' \leq_{\mathsf{c}}^{\epsilon,l} (t, P)$ and positions $p < p'$ in t such that $t_{|p} \equiv \beta$ and $t_{|p'} \equiv \beta'$, we have:

$$\mathsf{leave\text{-}st}(\beta \downarrow_R) \cap \mathsf{leave\text{-}st}(\beta' \downarrow_R) = \emptyset$$

(iv) For every $\beta, \beta' \leq_{\mathsf{c}}^{\epsilon,l} (t, P)$, if $\mathsf{leave\text{-}st}(\beta \downarrow_R) \cap \mathsf{leave\text{-}st}(\beta' \downarrow_R) \neq \emptyset$ then $\beta \equiv \beta$.

(v) The following couple of sets is well-nested:

$$\left(\left\{\beta \downarrow_R |\ \beta \leq_{\mathsf{c}}^{\epsilon,l} (t,P)\right\}, \left\{\gamma \downarrow_R |\ \gamma \leq_{\mathsf{l}}^{\epsilon,l} (t,P)\right\}\right)$$

For each property above, we give the proposition or lemma proving that it holds, or we announce in which section we will prove it.

(i) In other word, this means that every normalized basic terms has disjoint conditionals and leaves. We will prove this in Section 5.11.2.

(ii) For this to hold, we need to prove that, w.l.o.g., we can assume that true and false do not appear in the leaves of normalized basic terms. We will show this in Section 5.11.1.

(iii) This requires two non-trivial proof cut, which we explain in Section 5.11.3. It relies on Lemma 5.2.

(iv) This is a consequence of Proposition 5.11, which we already proved.

(v) We showed that these sets are well-nested in Lemma 5.14.

The rest of this section is organized as follows: in Section 5.11.1 we deal with (ii), by showing that we can assume that true and false do not appear in proof in normal proof form; in Section 5.11.2 we prove that conditionals and leaves of basic terms are disjoints, which we need for (i); in Section 5.11.3, we give examples of proof cut elimination used to obtain (iii); finally, in Section 5.11.4, we use Lemma 5.16 to prove that we can assume, w.l.o.g., that every leaf term appearing t has a persistent leaf in t.

5.11.1 Removing True and False From Basic Terms

In this section, we prove that we can assume, w.l.o.g., that true and false do not appear in the leaves of normalized basic terms.

Key Observation Let s be an if-free in R-normal form, s can be rewritten into a complex term u:

$$u \equiv C\left[\left(D_i[\vec{a}_i \diamond \vec{b}_i]\right)_i \diamond \vec{t}\right]$$

that is not if-free. Basically, the following proposition shows that as long as the term u does not contain true and false conditionals, the term s will always appear in the right-most and left-most branches of C. This is actually an invariant preserved by the term rewriting system R without the rules:

$$\mathsf{if\ true\ then}\ v\ \mathsf{else}\ w \to w \qquad\qquad \mathsf{if\ false\ then}\ v\ \mathsf{else}\ w \to w$$

Proposition 5.17. *For all if-free term s in R-normal form, if $s =_R C\left[\left(D_i[\vec{a}_i \diamond \vec{b}_i]\right)_i \diamond \vec{t}\right]$ where:*

- *$\vec{t} \cup \bigcup_i (\vec{a}_i \cup \vec{b}_i)$ are if-free and in R-normal form.*
- *For every i such that $D_i[\vec{a}_i \diamond \vec{b}_i]$ is a term appearing on the left-most (resp. right-most) branch of C, we have that $\mathsf{false} \notin \vec{a}_i \cup \vec{b}_i$ (resp. $\mathsf{true} \notin \vec{a}_i \cup \vec{b}_i$).*

Then the left-most (resp. right-most) element of $\vec{t}$ is s.

Proof. If suffices to show that the existence of a decomposition satisfying these two properties is preserved by $\to_R$, which is simple. We conclude by observing that since s is if-free, the only decomposition of $s \downarrow_R$ into $C\left[\left(D_i\left[\vec{a}_i \diamond \vec{b}_i\right]\right)_i \diamond \vec{t}\right]$ is such that $C \equiv []$. Hence $\vec{t}$ is a single element u, and $u \equiv s \downarrow_R \equiv s$. ■

We would like to prove that for every b, there exists no derivation of $b \sim \mathsf{true}$ or $b \sim \mathsf{false}$. Such derivations would be problematic since true and false are conditionals of constant size, but b could be of any size (and we are trying to bound all conditionals appearing in a proof). Also, the else branch of a true conditional can contain anything and is, a priori, not bounded by the proof conclusion. Using Proposition 5.17 we proved above, we show that there exists no proof of $b \sim \mathsf{true}$ or false, as long as b is if-free and the proof is in the fragment $\mathcal{A}_{\succ}$.

Proposition 5.18. *Let b an if-free conditional in R-normal, with $b \not\equiv \mathsf{false}$ (resp. $b \not\equiv \mathsf{true}$). Then there exists no derivation of $b \sim \mathsf{false}$ (resp. $b \sim \mathsf{true}$) in $\mathcal{A}_{\succ}$.*

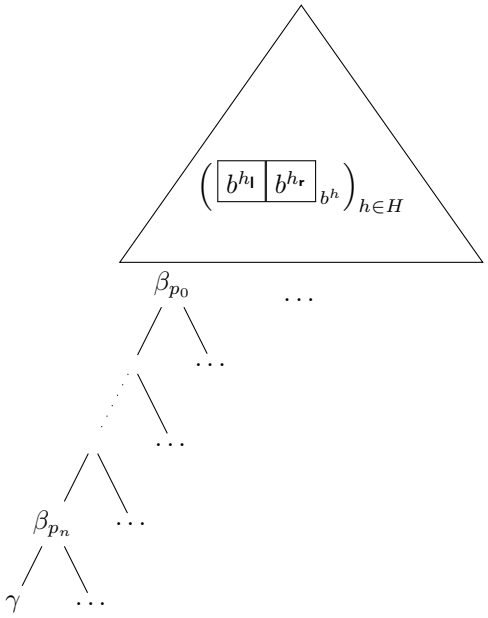


Figure 5.11: Shape of the Term in the Proof of Proposition 5.18

Proof. We prove only that there is no derivation of $b \sim \mathsf{false}$ in $\mathcal{A}_\succ$ (the proof that there is no derivation of $b \sim \mathsf{true}$ in $\mathcal{A}_\succ$ is exactly the same). We prove this by contradiction. Let b an if-free conditional in R-normal form which is not true and false, and let P be such that $P \vdash^{\mathsf{npf}} b \sim \mathsf{false}$. We choose the conditional b such that its proof P is of minimal size.

First the minimality of the derivation implies that for all $\mathsf{h} \in \mathsf{index}(P)$, for all b_0 such that $b_0 \leq^{\mathsf{h}}_{\mathsf{cs}} (b, P)$ or $b_0 \leq^{\mathsf{h}}_{\mathsf{cs}} (\mathsf{false}, P)$, $b_0 \neq_R \mathsf{false}$. Let $H = \mathsf{cs\text{-}pos}(P)$. We now focus on the left-most branch of the proof.

First Part Let $l \in \mathsf{label}(P)$. First we show that for all $\beta \leq^{\epsilon,l}_{\mathsf{c}} (b, P)$, $\beta \neq_R \mathsf{false}$. Assume that this is not the case, let $\beta =_R \mathsf{false}$ and β' be such that $(\beta, \beta') \leq^{\epsilon,l}_{\mathsf{c}\sim\mathsf{c}} (b \sim \mathsf{false}, P)$. If $\beta =_R \beta' =_R \mathsf{false}$ then there is an easy proof cut elimination which yields a smaller proof P' of $b \sim \mathsf{false}$.

Hence assume $\beta' \neq_R \mathsf{false}$. If $\beta =_R \mathsf{false}$ then $\mathsf{leave\text{-}st}(\beta \downarrow_R) = \mathsf{leave\text{-}st}(\mathsf{false} \downarrow_R) = \{\mathsf{false}\}$. Since β is a normalized basic conditional (for the CCA_2 trace $\mathcal{S}$ of its branch in P), and since false is a normalized basic conditional, using Proposition 5.11 we have $\beta \equiv \mathsf{false}$.

There exists a derivation of $\beta \sim \beta'$ in $\mathfrak{F}(\mathsf{FA_s}^* \cdot \mathsf{Dup}^* \cdot \mathrm{CCA}_2)$. Since $\beta \equiv \mathsf{false}$, no rules in $\mathsf{FA_s}$ are applied. Therefore the derivation is only an application of CCA_2, which is not possible. Similarly we do not have $\beta \neq_R \mathsf{false}$ and $\beta' =_R \mathsf{false}$.

Part 2 Using Proposition 5.11 we know that $\beta \neq_R \mathsf{false}$ implies that for all $u \in \mathsf{leave\text{-}st}(\beta \downarrow_R)$, $u \not\equiv \mathsf{false}$. Moreover, for any term w, $w \downarrow_R$ does not contain false in its conditionals (or we could apply $\mathsf{if\ false\ then}\ x\ \mathsf{else}\ y \to y$). Hence for every $a \in \mathsf{cond\text{-}st}(\beta \downarrow_R)$, $a \not\equiv \mathsf{false}$.

We let $(\gamma, \gamma') \leq^{\epsilon,l}_{\mathsf{l}} (b \sim \mathsf{false}, P)$ be the left-most elements (as shown in Figure 5.11). For all $a \in \mathsf{cond\text{-}st}(\gamma \downarrow_R)$, $a \not\equiv \mathsf{false}$. Hence if we let $u_0 \in \mathsf{leave\text{-}st}(\gamma \downarrow_R)$ be the left-most leave element of $\gamma \downarrow_R$, then by Proposition 5.17 we know that $u_0 \equiv b$. Recall that $b \neq_R \mathsf{false}$.

Similarly, by applying the exact same reasoning to the other side, we know that the left-most leaf element u'_0 of $\gamma' \downarrow_R$ is false, and by Proposition 5.11 we get that $\gamma' \equiv \mathsf{false}$. Since there exists a derivation of $\gamma \sim \gamma'$ in $\mathfrak{F}(\mathsf{FA_s}^* \cdot \mathsf{Dup}^* \cdot \mathrm{CCA}_2)$, no rule in $\mathsf{FA_s}$ is applied. Therefore the derivation is only an application of CCA_2. Contradiction. ■

Thanks to this proposition, we can ensure that any proof P of $t \sim t'$ does not contain a $\mathsf{CS}_\square$ or $\overline{\mathsf{BFA}}$ application on true or false: if we have a $\mathsf{CS}_\square$ or $\overline{\mathsf{BFA}}$ application on $(\mathsf{true}, \mathsf{true})$ or $(\mathsf{false}, \mathsf{false})$ then there is an easy proof cut elimination, and the previous proposition ensures that there are no $\mathsf{CS}_\square$ or $\overline{\mathsf{BFA}}$ applications on $(\mathsf{true}, b), (b, \mathsf{true}), (\mathsf{false}, b)$ or (b, false) (with $b \neq_R \mathsf{false}, \mathsf{true}$).

Proposition 5.19. *For all $P \vdash^{npf} t \sim t'$, there exists P' such that $P' \vdash^{npf} t \sim t'$ and for all $l \in \mathit{label}(P'), h \in \mathit{index}(P'), x \in \{l, r\}$ we have:*

$$\forall \beta \in \left((\leq_{c}^{h_x,l} \cup \leq_{cs}^{h_x})(t, P')\right) \cup \left((\leq_{c}^{h_x,l} \cup \leq_{cs}^{h_x})(t', P')\right), \quad \{\mathit{false}, \mathit{true}\} \cap \mathsf{leave\text{-}st}(\beta \downarrow_R) = \emptyset$$

Proof. Through simple proof cut eliminations, We can construct a proof P' from P such that:

$$\{(\mathsf{true}, \mathsf{true}), (\mathsf{false}, \mathsf{false})\} \cap (\leq_{c\sim c}^{h_x,l} (t \sim t', P) \cup \leq_{cs\sim cs}^{h_x} (t \sim t', P)) = \emptyset$$

Then using Proposition 5.18 we know that for all:

$$(\beta, \beta') \in (\leq_{c\sim c}^{h_x,l} (t \sim t', P) \cup \leq_{cs\sim cs}^{h_x} (t \sim t', P))$$

the conditionals β and β' are such that $\beta \neq_R \mathsf{false}$ and $\beta' \neq_R \mathsf{false}$ (same with true). Finally if $\beta \neq_R \mathsf{false}$ then using Proposition 5.11 we know that for every $u \in \mathsf{leave\text{-}st}(\beta \downarrow_R)$, $u \not\equiv \mathsf{false}$ (idem with true). ■

5.11.2 Basic Terms have Disjoints Conditionals and Leaves

We now prove that every normalized basic terms has disjoint conditionals and leaves.Let β be a normalized basic terms. First. we show that every conditional term b in $\mathsf{cond\text{-}st}(\beta \downarrow_R)$ is the leaf of another normalized basic term β', which is a strict subterm of β. Therefore, if $\mathsf{cond\text{-}st}(\beta \downarrow_R) \cap \mathsf{leave\text{-}st}(\beta \downarrow_R) \neq \emptyset$ then there exists β' such that $\mathsf{leave\text{-}st}(\beta \downarrow_R) \cap \mathsf{leave\text{-}st}(\beta' \downarrow_R) \neq \emptyset$. Using Proposition 5.11, we deduce that $\beta \equiv \beta'$, which contradicts the fact that β' is a strict sub-term of β.

First, we define the set of normalized basic conditionals appearing in a term t.

Definition 5.53. For all term t, we let $<_{\mathsf{bc}}^{\mathcal{S}} t$ be the set of $\mathcal{S}$-normalized basic conditional appearing in t, defined inductively by:

- If t is a $\mathcal{S}$-normalized simple term $C[\vec{b} \diamond \vec{u}]$, then:

$$<_{\mathsf{bc}}^{\mathcal{S}} t \;=\; \vec{b} \;\cup\; \left(<_{\mathsf{bc}}^{\mathcal{S}} \vec{b}\right) \;\cup\; \left(<_{\mathsf{bc}}^{\mathcal{S}} \vec{u}\right)$$

- If t is a $\mathcal{S}$-normalized basic term $B[\vec{w}, (\alpha_i)_i, (\mathsf{dec}_j)_j]$, then:

$$<_{\mathsf{bc}}^{\mathcal{S}} t \;=\; \bigcup_i <_{\mathsf{bc}}^{\mathcal{S}} \alpha_i \;\cup\; \bigcup_j <_{\mathsf{bc}}^{\mathcal{S}} \mathsf{dec}_j$$

- For every $\mathcal{S}$-encryption oracle call $t \equiv \{u\}_{\mathsf{pk}}^{r}$, then:

$$<_{\mathsf{bc}}^{\mathcal{S}} t \;= <_{\mathsf{bc}}^{\mathcal{S}} u$$

- For every $\mathcal{S}$-decryption oracle call $C[\vec{b} \diamond \vec{u}]$, let s, sk be such that s is if-free and the terms in $\vec{u}$ are of the form $\mathbf{0}(\mathsf{dec}(s[(\alpha_i), (\mathsf{dec}_j)_j], \mathsf{sk}))$ or $\mathsf{dec}(s[(\alpha_i), (\mathsf{dec}_j)_j], \mathsf{sk})$. Then:

$$<_{\mathsf{bc}}^{\mathcal{S}} t \;=\; \vec{b} \;\cup\; \left(<_{\mathsf{bc}}^{\mathcal{S}} \vec{b}\right) \;\cup\; \bigcup_i <_{\mathsf{bc}}^{\mathcal{S}} \alpha_i \;\cup\; \bigcup_j <_{\mathsf{bc}}^{\mathcal{S}} \mathsf{dec}_j$$

We show that the over-approximated set of conditionals $\overline{\mathsf{cond\text{-}st}}(\beta)$ is exactly the over-approximated set of leaves of the normalized basic conditionals that are subterm of β.

Proposition 5.20. *For every term β such that β is a $\mathcal{S}$-normalized basic term, $\mathcal{S}$-normalized simple term, $\mathcal{S}$-decryption oracle call or $\mathcal{S}$-encryption oracle call:*

$$\overline{\mathsf{cond\text{-}st}}(\beta) = \bigcup_{u <_{\mathsf{bc}}^{\mathcal{S}} \beta} \overline{\mathsf{leave\text{-}st}}(u)$$

Proof. We prove this by induction on the order $<_{\mathsf{ind}}^{\mathcal{S}}$, which, we recall, is the order stemming from $\mathcal{S}$-normalized basic terms, $\mathcal{S}$-normalized simple terms, $\mathcal{S}$-decryption oracle calls or $\mathcal{S}$-encryption oracle calls mutually inductive definitions.

Base Case If β is minimal for $<^{\mathcal{S}}_{\mathsf{ind}}$, then we have the following cases:

- $\mathcal{S}$-decryption oracle call: β is of the form $C[\vec{b} \diamond \vec{u}]$, and there exists s, sk such that terms in $\vec{u}$ are of the form $\mathbf{0}(\mathsf{dec}(s, \mathsf{sk}))$ or $\mathsf{dec}(s, \mathsf{sk})$, and s is if-free. Moreover by minimality of β the vector of terms $\vec{b}$ must be empty, since for all $b \in \vec{b}$, b is a $\mathcal{S}$-normalized basic term.
  Hence $\overline{\mathsf{cond\text{-}st}}(\beta) = \emptyset$. Finally since β is minimal there are no u such that $u <^{\mathcal{S}}_{\mathsf{bc}} \beta$.
- $\mathcal{S}$-encryption oracle call case cannot happen, as β would not be minimal.
- $\mathcal{S}$-normalized basic term: β contains no $\mathsf{if\ then\ else}$ symbol, hence $\overline{\mathsf{cond\text{-}st}}(\beta) = \emptyset$. Moreover since β is minimal there are no u such that $u <^{\mathcal{S}}_{\mathsf{bc}} \beta$.
- $\mathcal{S}$-normalized simple term case cannot happen, as β would not be minimal.

Inductive Case Let β be such that for all $\beta' \neq \beta$, if $\beta' <^{\mathcal{S}}_{\mathsf{ind}} \beta$ then the property holds for β'.

- $\mathcal{S}$-normalized basic term: β is of the form $B[\vec{w}, (\alpha_i)_i, (\mathsf{dec}_j)_j]$. The result is then immediate by induction hypothesis and using the definition of $\overline{\mathsf{cond\text{-}st}}(\cdot)$ and $<^{\mathcal{S}}_{\mathsf{bc}}$:

$$\begin{aligned}
\overline{\mathsf{cond\text{-}st}}(\beta) &= \bigcup_i \overline{\mathsf{cond\text{-}st}}(\alpha_i) \quad \cup \quad \bigcup_j \overline{\mathsf{cond\text{-}st}}(\mathsf{dec}_i) && \text{(By definition of } \overline{\mathsf{cond\text{-}st}}(\cdot)) \\
&= \bigcup_i \bigcup_{u <^{\mathcal{S}}_{\mathsf{bc}} \alpha_i} \overline{\mathsf{leave\text{-}st}}(u) \quad \cup \quad \bigcup_j \bigcup_{u <^{\mathcal{S}}_{\mathsf{bc}} \mathsf{dec}_j} \overline{\mathsf{leave\text{-}st}}(u) && \text{(By induction hypothesis)} \\
&= \bigcup_{u <^{\mathcal{S}}_{\mathsf{bc}} \beta} \overline{\mathsf{leave\text{-}st}}(u) && \text{(By definition of } <^{\mathcal{S}}_{\mathsf{bc}})
\end{aligned}$$

- $\mathcal{S}$-decryption oracle call: t is of the form $C[\vec{g} \diamond \vec{u}]$, where there exists s, sk such that terms in $\vec{u}$ are of the form $\mathbf{0}(\mathsf{dec}(s[(\alpha_i), (\mathsf{dec}_j)_j], \mathsf{sk}))$ or $\mathsf{dec}(s[(\alpha_i), (\mathsf{dec}_j)_j], \mathsf{sk})$, and s is if-free. Then:

$$\begin{aligned}
\overline{\mathsf{cond\text{-}st}}(\beta) &= \bigcup_i \overline{\mathsf{cond\text{-}st}}(\alpha_i) \quad \cup \bigcup_j \overline{\mathsf{cond\text{-}st}}(\mathsf{dec}_i) \quad \cup \overline{\mathsf{cond\text{-}st}}(\vec{g}) \quad \cup \overline{\mathsf{leave\text{-}st}}(\vec{g}) \\
& && \text{(By definition of } \overline{\mathsf{cond\text{-}st}}(\cdot)) \\
&= \bigcup_i \bigcup_{u <^{\mathcal{S}}_{\mathsf{bc}} \alpha_i} \overline{\mathsf{leave\text{-}st}}(u) \cup \bigcup_j \bigcup_{u <^{\mathcal{S}}_{\mathsf{bc}} \mathsf{dec}_j} \overline{\mathsf{leave\text{-}st}}(u) \cup \bigcup_{u <^{\mathcal{S}}_{\mathsf{bc}} \vec{g}} \overline{\mathsf{leave\text{-}st}}(u) \cup \overline{\mathsf{leave\text{-}st}}(\vec{g}) \\
&\text{(By induction hypothesis: remark that guards in } \vec{g} \text{ are } \mathcal{S}\text{-normalized basic terms s.t. } \vec{g} \leq^{\mathcal{S}}_{\mathsf{bt}} \beta) \\
&= \bigcup_{u <^{\mathcal{S}}_{\mathsf{bc}} \beta} \overline{\mathsf{leave\text{-}st}}(u) && \text{(By definition of } <^{\mathcal{S}}_{\mathsf{bc}})
\end{aligned}$$

- $\mathcal{S}$-encryption oracle call: t is of the form $\{s\}^r_{\mathsf{pk}}$, then:

$$\begin{aligned}
\overline{\mathsf{cond\text{-}st}}(\beta) &= \overline{\mathsf{cond\text{-}st}}(s) && \text{(By definition of } \overline{\mathsf{cond\text{-}st}}(\cdot)) \\
&= \bigcup_{u <^{\mathcal{S}}_{\mathsf{bc}} s} \overline{\mathsf{leave\text{-}st}}(u) && \text{(By induction hypothesis)} \\
&= \bigcup_{u <^{\mathcal{S}}_{\mathsf{bc}} \beta} \overline{\mathsf{leave\text{-}st}}(u) && \text{(By definition of } <^{\mathcal{S}}_{\mathsf{bc}})
\end{aligned}$$

- $\mathcal{S}$-normalized simple term: t is of the form $C[\vec{b} \diamond \vec{v}]$. Then:

$$\begin{aligned}
\overline{\mathsf{cond\text{-}st}}(\beta) &= \overline{\mathsf{cond\text{-}st}}(\vec{b}) \quad \cup \overline{\mathsf{cond\text{-}st}}(\vec{v}) \quad \cup \overline{\mathsf{leave\text{-}st}}(\vec{b}) && \text{(By definition of } \overline{\mathsf{cond\text{-}st}}(\cdot)) \\
&= \bigcup_{u <^{\mathcal{S}}_{\mathsf{bc}} \vec{b}} \overline{\mathsf{leave\text{-}st}}(u) \cup \bigcup_{u <^{\mathcal{S}}_{\mathsf{bc}} \vec{v}} \overline{\mathsf{leave\text{-}st}}(u) \cup \overline{\mathsf{leave\text{-}st}}(\vec{b}) && \text{(By induction hypothesis)} \\
&= \bigcup_{u <^{\mathcal{S}}_{\mathsf{bc}} \beta} \overline{\mathsf{leave\text{-}st}}(u) && \text{(By definition of } <^{\mathcal{S}}_{\mathsf{bc}}) \quad \blacksquare
\end{aligned}$$

We can now prove that every normalized basic terms has disjoint conditionals and leaves, using Proposition 5.11 and the result above.

Proposition 5.21. *Let $P \vdash^{npf} t \sim t'$. Then for all h, l for all $\beta \leq^{\mathsf{h},l}_{bt} (t, P)$, $\overline{\mathsf{cond\text{-}st}}(\beta) \cap \overline{\mathsf{leave\text{-}st}}(\beta) = \emptyset$.*

Proof. Let h, l and $\beta \leq^{\mathsf{h},l}_{\mathsf{bt}} (t, P)$ be such that $\overline{\mathsf{cond\text{-}st}}(\beta) \cap \overline{\mathsf{leave\text{-}st}}(\beta) \neq \emptyset$. By Proposition 5.20 this means that there exists a $\mathcal{S}_l^P$-normalized basic term $u <^{\mathcal{S}_l}_{\mathsf{bc}} \beta$ such that $\overline{\mathsf{leave\text{-}st}}(u) \cap \overline{\mathsf{leave\text{-}st}}(\beta) \neq \emptyset$.

By Proposition 5.11, $u \equiv \beta$. But $u <^{\mathcal{S}_l}_{\mathsf{bc}} \beta$ implies that u is a strict subterm of β. Absurd. ■

5.11.3 Proof Cuts on Branches

For the hypothesis (iii) of Lemma 5.16 to hold, we need to make sure that the same conditional never appear twice in the same branch[4]. Therefore, we need to show that if some normalized basic term β appears twice in the same branch, then there is a proof cut. We have three possibilities:

- The two occurrences of β are involved in $\overline{\mathsf{BFA}}$ applications.
- The two occurrences of β are involved in $\mathsf{CS}_\Box$ applications.
- One occurrence of β is with an $\overline{\mathsf{BFA}}$ application, the other with a $\mathsf{CS}_\Box$ applications.

We only present proof cut eliminations for the first two cases. We deal with the cross case later.

$\overline{\mathsf{BFA}}$ Rule We already used this cut elimination to deal with Example 5.7 for conditionals involved in $\overline{\mathsf{BFA}}$ applications. The cuts we want to eliminate are of the form:

$$\dfrac{a_1, a_2, u_3, v_4, w_5 \sim b_1, c_2, r_3, s_4, t_5}{\begin{array}{ccc} a_1 & & b_1 \\ a_2 \quad w_5 & \sim & c_2 \quad t_5 \\ u_3 \quad v_4 & & r_3 \quad s_4 \\ \underbrace{\qquad}_{\sigma} & & \underbrace{\qquad}_{\tau} \end{array}}\ \overline{\mathsf{BFA}}^{(2)} \tag{5.18}$$

Using Lemma 5.1, we extract a proof of $a_1, a_2 \sim b_1, c_2$, which, thanks to the ordered strategy, is in $\mathfrak{F}(\mathsf{FA_s}^* \cdot \mathsf{Dup}^* \cdot \mathrm{CCA}_2)$. From Lemma 5.2 we get that $b \equiv c$. We then replace (5.18) by:

$$\dfrac{\dfrac{a_1, u_3, w_5 \sim b_1, r_3, t_5}{\begin{array}{ccc} a_1 & & b_1 \\ u_3 \quad w_5 & \sim & r_3 \quad t_5 \end{array}}\ \overline{\mathsf{BFA}}}{\sigma \sim \tau}\ R$$

We retrieve a proof in $\mathcal{A}_\succ$ by pulling R to the beginning of the proof.

$\mathsf{CS}_\Box$ Rule The $\mathsf{CS}_\Box$ case is more complicated. E.g., take two boxed $\mathsf{CS}_\Box$ conditionals for the same if-free conditional a, and two arbitrary $\mathsf{CS}_\Box$ conditionals for the right side:

$$a_i^\Box \equiv \boxed{a_i^l}\boxed{a_i^r}_a \ (i \in \{1,2\}) \quad b_1^\Box \equiv \boxed{b_1^l}\boxed{b_1^r}_b \quad c_2^\Box \equiv \boxed{c_2^l}\boxed{c_2^r}_c$$

Consider the following cut:

$$\dfrac{\begin{array}{ccc} \vdots\ (A) & \vdots\ (B) & \vdots\ (C) \\ a_1^l, a_2^l, u_3 \sim b_1^l, c_2^l, r_3 & a_1^l, a_2^r, v_4 \sim b_1^l, c_2^r, s_4 & a_1^r, w_5 \sim b_1^r, t_5 \end{array}}{\begin{array}{ccc} a_1^\Box & & b_1^\Box \\ a_2^\Box \quad w_5 & \sim & c_2^\Box \quad t_5 \\ u_3 \quad v_4 & & r_3 \quad s_4 \\ \underbrace{\qquad}_{\sigma} & & \underbrace{\qquad}_{\tau} \end{array}}\ \mathsf{CS}_\Box^{(2)}$$

[4]Indeed, we recall that Proposition 5.11 shows that if $\mathsf{leave\text{-}st}(\beta \downarrow_R) \cap \mathsf{leave\text{-}st}(\beta' \downarrow_R) \neq \emptyset$ then $\beta \equiv \beta'$.

As we did for $\overline{\mathsf{BFA}}$, we can extract from (A), using Lemma 5.1, a proof of $a_1^l, a_2^l \sim b_1^l, c_2^l$. But using the ordered strategy, we get that this proof is in $\mathcal{A}_{\mathsf{CS}_\square}$, which we recall is the fragment:

$$\mathfrak{F}(\mathsf{CS}_\square^* \cdot \{\overline{\mathsf{BFA}}(b, b')\}^* \cdot \mathsf{UnF} \cdot {\mathsf{FA}_\mathsf{s}}^* \cdot \mathsf{Dup}^* \cdot \mathrm{CCA}_2)$$

Therefore we cannot apply Lemma 5.2. To deal with this cut, we generalize Lemma 5.2 to the case where the proof is in $\mathcal{A}_{\mathsf{CS}_\square}$. For this, we need the extra assumptions that $a_1^l, a_2^l, b_1^l, c_2^l$ are if-free, which is a side-condition of $\mathsf{CS}_\square$.

Lemma 5.17. *For every terms a, a', b, c with if-free R-normal forms, if $a =_R a'$ and $P \vdash^{\mathsf{npf}} a, a' \sim b, c$ then $b =_R c$.*

Proof. Let $t \equiv \langle a\,,\, a\rangle$ and $t' \equiv \langle b\,,\, c\rangle$, we know that there exists P' such that $P' \vdash^{\mathsf{npf}} t \sim t'$ since $P \vdash^{\mathsf{npf}} a, a' \sim b, c$. Using Proposition 5.19, we can assume that for every $h \in \mathsf{index}(P), l, \mathsf{x}$:

$$\forall \beta \in \left((\leq_{\mathsf{c}}^{h_\mathsf{x}, l} \cup \leq_{\mathsf{cs}}^{h_\mathsf{x}, l})(t, P')\right) \cup \left((\leq_{\mathsf{c}}^{h_\mathsf{x}, l} \cup \leq_{\mathsf{cs}}^{h_\mathsf{x}, l})(t', P')\right), \quad \{\mathsf{false}, \mathsf{true}\} \cap \mathsf{leave\text{-}st}(\beta \downarrow_R) = \emptyset$$

Let $(\gamma, \gamma') \leq_{\mathsf{l}}^{\epsilon, l} (t \sim t', P)$ be the left-most elements of t and t'. By Proposition 5.17 we know that $\langle a\,,\, a\rangle \downarrow_R \in \mathsf{leave\text{-}st}(\gamma \downarrow_R)$ and $\langle b\,,\, c\rangle \downarrow_R \in \mathsf{leave\text{-}st}(\gamma' \downarrow_R)$. More precisely we know that $\langle b\,,\, c\rangle$ is the left-most element of $\gamma' \downarrow_R$.

Since $\gamma \sim \gamma'$ is provable in $\mathfrak{F}(\mathsf{FA_s}^* \cdot \mathsf{Dup}^* \cdot \mathrm{CCA}_2)$, we know that there exist $\mathcal{S}_l^P$-normalized basic terms γ_1, γ_2 and $\mathcal{S}_l'^P$-normalized basic terms γ_1', γ_2' such that $\gamma =_R \langle \gamma_1\,,\, \gamma_2\rangle$, $\gamma' =_R \langle \gamma_1'\,,\, \gamma_2'\rangle$, and $\gamma_1, \gamma_2 \sim \gamma_1', \gamma_2'$ is provable in $\mathfrak{F}(\mathsf{FA_s}^* \cdot \mathsf{Dup}^* \cdot \mathrm{CCA}_2)$.

Moreover $a \in \mathsf{leave\text{-}st}(\gamma_1 \downarrow_R)$ and $a \in \mathsf{leave\text{-}st}(\gamma_2 \downarrow_R)$, hence $\mathsf{leave\text{-}st}(\gamma_1 \downarrow_R) \cap \mathsf{leave\text{-}st}(\gamma_2 \downarrow_R) \neq \emptyset$. By Proposition 5.11 we deduce that $\gamma_1 \equiv \gamma_2$.

Therefore there exists a proof of $\gamma_1, \gamma_1 \sim \gamma_1', \gamma_2'$ in $\mathfrak{F}(\mathsf{FA_s}^* \cdot \mathsf{Dup}^* \cdot \mathrm{CCA}_2)$. By Lemma 5.2, $\gamma_1' \equiv \gamma_2'$. We conclude by observing that since $\langle b\,,\, c\rangle$ is the let-most element of $\gamma' \downarrow_R$, b and c are the left-most element of, respectively, γ_1' and γ_2'. Therefore $b \equiv c$. ■

We now deal with the cut above. Using Lemma 5.17, we know that $b =_R c$. Since b, c are in R-normal form, $b \equiv c$ and therefore $b_1^\square =_{R_\square} b =_{R_\square} c_2^\square$ (using well-formedness). Similarly $a_1^\square =_{R_\square} a =_{R_\square} a_2^\square$. This yields the (cut-free) proof:

$$\dfrac{\dfrac{\vdots\ (A') \qquad\qquad \vdots\ (C)}{a_1^l, u_3 \sim b_1^l, r_3 \qquad a_1^r, w_5 \sim b_1^r, t_5}\ \mathsf{CS}_\square}{\begin{array}{ccc} a_1^\square & & b_1^\square \\ u_3 \quad w_5 & \sim & r_3 \quad t_5 \end{array}\ R_\square}$$
$$\sigma \sim \tau$$

where (A') is extracted from (A) by Lemma 5.7. Finally, to get a proof in $\mathcal{A}_{\succ}$, we commute the $R_\square$ rewriting to the beginning.

5.11.4 Main Lemma

Definition 5.54. A *directed path* ${}^\delta\vec{\rho}$ is a sequence $(b_0, d_0), \ldots, (b_n, d_n)$ where $b_0, \ldots, b_n$ are conditionals and $d_0, \ldots, d_n$ (the directions) are in $\{\mathsf{then}, \mathsf{else}\}$.

Two directed paths ${}^\delta\vec{\rho}$ and ${}^\delta\vec{\rho}'$ have the same directions if:

- they have the same length.
- the sequences of *directions* $d_0, \ldots, d_n$ and $d_0', \ldots, d_n'$ extracted from, resp., ${}^\delta\vec{\rho}$ and ${}^\delta\vec{\rho}'$, are equal.

Given a directed path ${}^\delta\vec{\rho}$, we let $\vec{\rho}$ stands for the sequence of *conditionals* extracted from ${}^\delta\vec{\rho}$.

Example 5.19. Let s be the term of Example 5.3, which we recall below:

$$\mathsf{if}\ b_1\ \mathsf{then}\ \mathsf{if}\ b_2\ \mathsf{then}\ t_1\ \mathsf{else}\ t_2$$
$$\mathsf{else}\ t_3$$

Then ${}^\delta\vec{\rho} = (b_1, \mathsf{then}), (b_2, \mathsf{else})$ is the directed path corresponding to the branch starting at the root of s and ending at the term t_2. Moreover, $\vec{\rho} = b_1, b_2$. □

Definition 5.55. Let $P \vdash^{\mathsf{npf}} t \sim t'$, we know that t is of the form:

$$t \equiv C\left[\left(\boxed{b^{h_{\mathsf{l}}}}\boxed{b^{h_{\mathsf{r}}}}_{b^h}\right)_{h\in H} \diamond \left(D_l\left[(\beta)_{\beta \leq^{\epsilon,l}_{\mathsf{c}}(t,P)} \diamond (\gamma)_{\gamma \leq^{\epsilon,l}_{\mathsf{l}}(t,P)}\right]\right)_{l\in L}\right]$$

For all l, we let:

- ${}^{\delta}\mathsf{cs\text{-}path}^{\epsilon,l}(t,P)$ be the directed path of conditional occurring from the root of t to $D_l[]$ in P.
- ${}^{\delta}\mathsf{cs\text{-}path}^{\epsilon,l}_{\sim}(t \sim t', P)$ be the directed path of pairs of conditionals occurring from the root of (t,t') to $D_l[]$ in P.

We extend this to all $h \in \mathsf{index}(P), \mathsf{x} \in \{\mathsf{l},\mathsf{r}\}$ by having:

$$\begin{aligned} & {}^{\delta}\mathsf{cs\text{-}path}^{h_{\mathsf{x}},l}(t,P) && = {}^{\delta}\mathsf{cs\text{-}path}^{\epsilon,l}(b, \mathsf{extract}_{\mathsf{x}}(h,P)) \\ \text{and} \quad & {}^{\delta}\mathsf{cs\text{-}path}^{h_{\mathsf{x}},l}_{\sim}(t \sim t', P) && = {}^{\delta}\mathsf{cs\text{-}path}^{\epsilon,l}_{\sim}(b \sim b', \mathsf{extract}_{\mathsf{x}}(h,P)) \end{aligned}$$

where $\mathsf{extract}_{\mathsf{x}}(h,P)$ is a proof of $b \sim b'$.

We let the depth of a position h in P to be the number of nested applications of the $\mathsf{CS}_{\square}$ rule to h.

Definition 5.56. Let $P \vdash^{\mathsf{npf}} t \sim t'$. For every $h \in \mathsf{index}(P)$, we let $\mathsf{if\text{-}depth}_P(h)$ be the depth of h in P, defined by:

$$\mathsf{if\text{-}depth}_P(h) = \begin{cases} 0 & \text{if } h \in \mathsf{cs\text{-}pos}(P) \\ 1 + \mathsf{if\text{-}depth}_{P^{\mathsf{l}}}(h) & \text{if } \exists g \in \mathsf{cs\text{-}pos}(P) \text{ s.t. } h \in \mathsf{index}(P^{\mathsf{l}}) \text{ where } P^{\mathsf{l}} = \mathsf{extract}_{\mathsf{l}}(g,P) \\ 1 + \mathsf{if\text{-}depth}_{P^{\mathsf{r}}}(h) & \text{if } \exists g \in \mathsf{cs\text{-}pos}(P) \text{ s.t. } h \in \mathsf{index}(P^{\mathsf{r}}) \text{ where } P^{\mathsf{r}} = \mathsf{extract}_{\mathsf{r}}(g,P) \end{cases}$$

Lemma 5.18. *Let $P \vdash^{\mathsf{npf}} t \sim t'$. There exists P' such that $P' \vdash^{\mathsf{npf}} t \sim t'$ and for all $h \in \mathsf{index}(P')$ with $h \neq \epsilon$, for all $\mathsf{x} \in \{\mathsf{l},\mathsf{r}\}$, if we let $\mathsf{h} = h_{\mathsf{x}}$ and $P^{\mathsf{h}} = \mathsf{extract}_{\mathsf{x}}(h,P')$ be the proof of $b^{\mathsf{h}} \sim b'^{\mathsf{h}}$ then for all $l \in \mathsf{label}(P^{\mathsf{h}})$:*

(a) The proof P^{h} does not use the $\{\overline{\mathsf{BFA}}(b,b')\}$ rules.

(b) $\mathsf{cs\text{-}path}^{\mathsf{h},l}(t,P)$ (resp. $\mathsf{cs\text{-}path}^{\mathsf{h},l}(t',P)$) does not contain two occurrences of the same conditional.

(c) For all $\gamma \leq^{\mathsf{h},l}_{\mathsf{l}} (t,P')$, $(b^{\mathsf{h}} \downarrow_R) \in \mathsf{leave\text{-}st}(\gamma \downarrow_R)$ and for all $\gamma' \leq^{\mathsf{h},l}_{\mathsf{l}} (t',P')$, $(b'^{\mathsf{h}} \downarrow_R) \in \mathsf{leave\text{-}st}(\gamma' \downarrow_R)$.

(d) For all $\beta \leq^{\epsilon,l}_{\mathsf{c}} (t,P')$, $\mathsf{leave\text{-}st}(\beta \downarrow_R) \cap \mathsf{cs\text{-}path}^{\epsilon,l}(t,P) = \emptyset$ (same for t').

(e) For all $\gamma \leq^{\epsilon,l}_{\mathsf{l}} (t,P')$, $\mathsf{leave\text{-}st}(t \downarrow_R) \cap \mathsf{leave\text{-}st}(\gamma \downarrow_R) \neq \emptyset$ (same for t').

Proof. Using Proposition 5.19, we know that we have P such that $P \vdash^{\mathsf{npf}} t \sim t'$ and for all $l \in \mathsf{label}(P), h \in \mathsf{index}(P), \mathsf{x} \in \{\mathsf{l},\mathsf{r}\}$ we have:

$$\forall \beta \in \left((\leq^{h_{\mathsf{x}},l}_{\mathsf{c}} \cup \leq^{h_{\mathsf{x}},l}_{\mathsf{cs}})(t,P)\right) \cup \left((\leq^{h_{\mathsf{x}},l}_{\mathsf{c}} \cup \leq^{h_{\mathsf{x}},l}_{\mathsf{cs}})(t',P)\right), \quad \{\mathsf{false},\mathsf{true}\} \cap \mathsf{leave\text{-}st}(\beta \downarrow_R) = \emptyset \tag{5.19}$$

First, we rewrite the proof P so that all $\mathsf{CS}^{\mathsf{no}}_{\mathsf{if}}$ applications are of the form:

$$\frac{b, (u_i)_i \sim b', (u'_i)_i \quad b, (v_i)_i \sim b', (v'_i)_i}{(\mathsf{if}\ b\ \mathsf{then}\ u_i\ \mathsf{else}\ v_i)_i \sim (\mathsf{if}\ b'\ \mathsf{then}\ u'_i\ \mathsf{else}\ v'_i)_i}\ \mathsf{CS}^{\mathsf{no}}_{\mathsf{if}} \tag{5.20}$$

We prove by induction on n, starting with the inner-most $\mathsf{CS}^{\mathsf{no}}_{\mathsf{if}}$ conditionals, that there exists P such that $P \vdash^{\mathsf{npf}} t \sim t'$, (5.19) is true for P and the following properties hold for all $h, h' \in \mathsf{index}(P)$:

(i) If $\mathsf{if\text{-}depth}_P(h) \geq n$ then the $\mathsf{extract}_{\mathsf{l}}(h,P)$ and $\mathsf{extract}_{\mathsf{r}}(h,P)$ do not use the $\{\overline{\mathsf{BFA}}(b,b')\}$ rules.

(ii) If $\mathsf{if\text{-}depth}_P(h) \geq n$ then for all x, l, $\mathsf{cs\text{-}path}^{h_{\mathsf{x}},l}(t,P)$ and $\mathsf{cs\text{-}path}^{h_{\mathsf{x}},l}(t',P)$ do not contain two occurrences of the same conditional.

(iii) If $\mathsf{if\text{-}depth}_P(h) \geq n$ then for all x, if $\mathsf{extract}_{\mathsf{x}}(h,P)$ is the proof of $b \sim b'$ then for all l, for all $\gamma \leq^{h_{\mathsf{x}},l}_{\mathsf{l}} (t,P)$, $(b \downarrow_R) \in \mathsf{leave\text{-}st}(\gamma \downarrow_R)$ and for all $\gamma' \leq^{h_{\mathsf{x}},l}_{\mathsf{l}} (t',P)$, $(b' \downarrow_R) \in \mathsf{leave\text{-}st}(\gamma' \downarrow_R)$.

(iv) If $\mathsf{if\text{-}depth}_P(h) < n$ then for all $h, h' \in \mathsf{index}(P)$ such that $h \leq h'$, if we let h'' be such that $h' = h \cdot h''$ and x be such that $h'' \in \mathsf{index}(\mathsf{extract}_{\mathsf{x}}(h,P))$, then for all x', for all $l \in \mathsf{label}(\mathsf{extract}_{\mathsf{x}'}(h',P))$, we have

$${}^{\delta}\mathsf{cs\text{-}path}^{h_{\mathsf{x}},l}(t,P) \supseteq {}^{\delta}\mathsf{cs\text{-}path}^{h'_{\mathsf{x}'},l}(t,P)$$

Let $n_{\max}$ be the maximal $\mathsf{if\text{-}depth}$ in the proof of $t \sim t'$:

$$n_{\max} = \max_{h \in \mathsf{index}(P)} \mathsf{if\text{-}depth}_P(h)$$

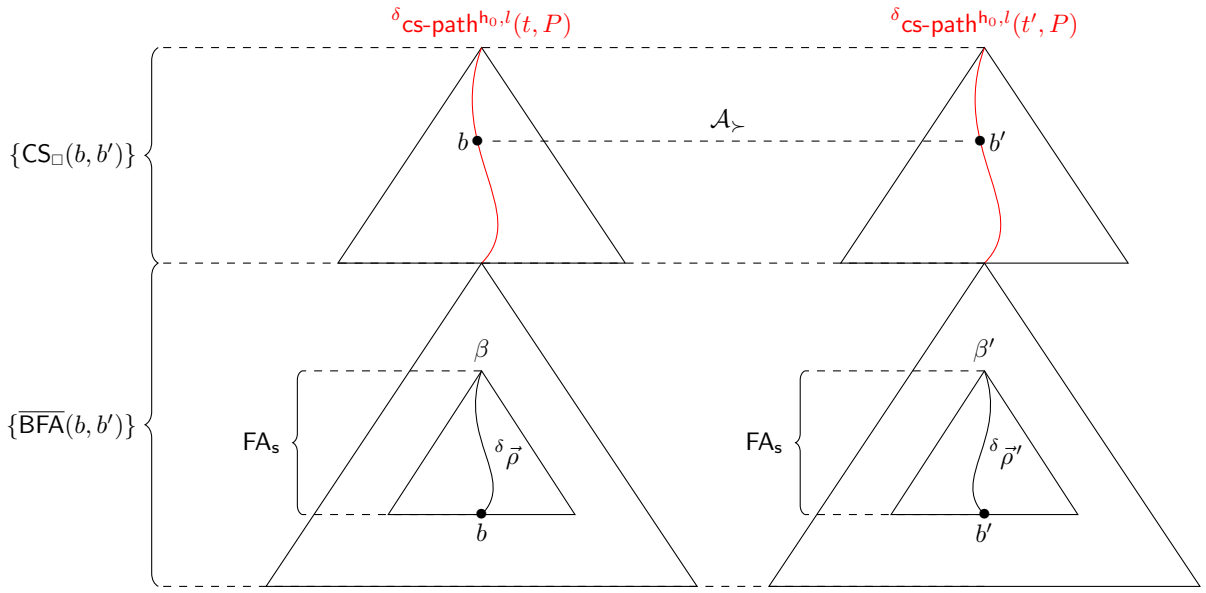


Figure 5.12: Corresponding occurrences of b and b' in the proof of Lemma 5.18

Base Case We are going to show that the invariants hold at $n_{\max}+1$. Invariants (i), (ii) and (iii) are obvious, since there exists no h such that $\mathsf{if\text{-}depth}_P(h) \geq n_{\max}+1$; and invariant (iv) is a consequence of the rewriting done in (5.20).

Inductive Case: Assume that the property holds for $n+1$ and let us show that it holds for n.

Step 1 Let $l \in \mathsf{label}(P)$ and $h_0 \in \mathsf{h\text{-}branch}(l)$ such that $\mathsf{if\text{-}depth}_P(h_0) = n$. Let $\mathsf{x}_0 \in \{\mathsf{l},\mathsf{r}\}$ and $\mathsf{h}_0 = h_{0\mathsf{x}_0}$. We start by showing that for all l, for all $\beta \leq_{\mathsf{c}}^{\mathsf{h}_0,l} (t,P)$, if there exists $b \in \mathsf{cs\text{-}path}^{\mathsf{h}_0,l}(t,P)$ such that $b \in \mathsf{leave\text{-}st}(\beta\downarrow_R)$ then there exists $(b,b') \in \mathsf{cs\text{-}path}_{\sim}^{\mathsf{h}_0,l}(t,P)$ and $(\beta,\beta') \leq_{\mathsf{c}\sim\mathsf{c}}^{\mathsf{h}_0,l} (t\sim t',P)$ s.t.:

- $b' \in \mathsf{leave\text{-}st}(\beta'\downarrow_R)$.
- There exists a directed path $^{\delta}\vec{\rho}$ (resp. $^{\delta}\vec{\rho}'$) of the conditionals occurring from the root of $\beta\downarrow_R$ (resp. $\beta'\downarrow_R$) to a leave b (resp. b') such that $^{\delta}\vec{\rho} \subseteq {}^{\delta}\mathsf{cs\text{-}path}^{\mathsf{h}_0,l}(t,P)$ (resp. $^{\delta}\vec{\rho}' \subseteq {}^{\delta}\mathsf{cs\text{-}path}^{\mathsf{h}_0,l}(t,P)$).

This is described in Figure 5.12.

Let $\beta \leq_{\mathsf{c}}^{\mathsf{h}_0,l} (t,P)$ and $b \in \mathsf{cs\text{-}path}^{\mathsf{h}_0,l}(t,P)$ such that $b \in \mathsf{leave\text{-}st}(\beta\downarrow_R)$. We know that there exists b' and β' such that $(b,b') \in \mathsf{cs\text{-}path}_{\sim}^{\mathsf{h}_0,l}(t,P)$ and $(\beta,\beta') \leq_{\mathsf{c}\sim\mathsf{c}}^{\mathsf{h}_0,l} (t\sim t',P)$.

Let $h \in \mathsf{cs\text{-}pos}(\mathsf{extract}_{x_0}(h_0,P))$ and x be the direction taken in l at h be such that $\mathsf{extract}(h,P)$ is the rule $\mathsf{CS}_{\square}(b,b')$. We know that $\mathsf{extract}_{\mathsf{x}}(h,P)$ is a proof of $a \sim a'$, where $a =_R b$ and $a' =_R b'$. As $\mathsf{if\text{-}depth}(h) = n+1$ we know by induction hypothesis (i) that $\mathsf{extract}_{\mathsf{x}}(h,P)$ does not uses $\{\overline{\mathsf{BFA}}(b,b')\}$. Hence the set $\leq_{\mathsf{l}}^{\epsilon,l} (a,\mathsf{extract}_{\mathsf{x}}(h,P))$ is the singleton $\{\gamma_l\}$ and the set $\leq_{\mathsf{l}}^{\epsilon,l} (a',\mathsf{extract}_{\mathsf{x}}(h,P))$ is the singleton $\{\gamma'_l\}$. Let $H = \mathsf{index}(\mathsf{extract}_{\mathsf{x}}(h,P))$, we have:

$$a \equiv C\big[(b^g)_{g\in H} \diamond (\gamma_{l_a})_{l_a}\big] \qquad a' \equiv C\big[(b'^g)_{g\in H} \diamond (\gamma'_{l_a})_{l_a}\big]$$

By induction hypothesis (iii) we know that $b \in \mathsf{leave\text{-}st}(\gamma_l\downarrow_R)$ and $b' \in \mathsf{leave\text{-}st}(\gamma'_l\downarrow_R)$. γ_l and β are $\mathcal{S}_l$-normalized basic terms, hence using Proposition 5.11 we know that $\beta \equiv \gamma_l$. We can extract from the branch l of P a proof of $\gamma_l, \beta \sim \gamma'_l, \beta'$ in $\mathfrak{F}(\mathsf{FA_s}^* \cdot \mathsf{Dup}^* \cdot \overline{\mathsf{CCA_2}})$. Therefore, using Lemma 5.2, we get that $\beta' \equiv \gamma'_l$. Since $b' \in \mathsf{leave\text{-}st}(\gamma'_l\downarrow_R)$, we deduce that $b' \in \mathsf{leave\text{-}st}(\beta'\downarrow_R)$. This concludes the proof of the first bullet point.

We now prove the second bullet point. By induction hypothesis (iv) we know that:

$$^{\delta}\mathsf{cs\text{-}path}^{\mathsf{h}_0,l}(t,P) \supseteq {}^{\delta}\mathsf{cs\text{-}path}^{h_{\mathsf{x}},l}(t,P) \qquad {}^{\delta}\mathsf{cs\text{-}path}^{\mathsf{h}_0,l}(t',P) \supseteq {}^{\delta}\mathsf{cs\text{-}path}^{h_{\mathsf{x}},l}(t',P)$$

By definition of $\vec{\rho}$, $\mathsf{cond\text{-}st}(\gamma_l\downarrow_R) = \mathsf{cond\text{-}st}(\beta\downarrow_R) \supseteq \vec{\rho}$. We can do better, and obtained an inclusion in the *directed* conditional path. First, we know that:

- $a \equiv C\left[(b^g)_{g\in H} \diamond (\gamma_{l_a})_{l_a}\right]$, $a =_R b$ and b is if-free and in R-normal form.
- Invariant (ii) holds, hence ${}^{\delta}\mathsf{cs\text{-}path}^{h_{\mathsf{x}},l}(t,P)$ does not contain two occurrences of the same conditional.
- ${}^{\delta}\mathsf{cs\text{-}path}^{h_{\mathsf{x}},l}(t,P)$ does not contain true and false.

The existence of a decomposition as described above is invariant by (chunks of) $\rightarrow_{R^{\succ_u}}$ reductions, for a well-chosen ordering $\succ_u$. At the end of the reduction, we have b. By looking at the reduction backward, we see that b is a leaf of $\gamma_l \downarrow_{R^{\succ_u}}$, such that the directed path ${}^{\delta}\vec{\rho}$ from the root of $\gamma_l \downarrow_{R^{\succ_u}}$ to b is included in the path from the root of a to γ_l.

We deduce that ${}^{\delta}\vec{\rho} \subseteq {}^{\delta}\mathsf{cs\text{-}path}^{h_{\mathsf{x}},l}(t,P)$. By consequence, ${}^{\delta}\vec{\rho} \subseteq {}^{\delta}\mathsf{cs\text{-}path}^{\mathsf{h}_0,l}(t,P)$. Similarly we show that ${}^{\delta}\vec{\rho}' \subseteq {}^{\delta}\mathsf{cs\text{-}path}^{\mathsf{h}_0,l}(t',P)$.

Step 2 By doing some proof cut elimination, we can guarantee that for all l, for all $\beta \leq_{\mathsf{c}}^{\mathsf{h}_0,l} (t,P)$:

$$\mathsf{leave\text{-}st}(\beta\downarrow_R) \cap \mathsf{cs\text{-}path}^{\mathsf{h}_0,l}(t,P) = \emptyset$$

Assume this is not the case: using **Step 1** we have:

$${}^{\delta}\vec{\rho} \subseteq {}^{\delta}\mathsf{cs\text{-}path}^{\mathsf{h}_0,l}(t,P) \qquad\qquad {}^{\delta}\vec{\rho}' \subseteq {}^{\delta}\mathsf{cs\text{-}path}^{\mathsf{h}_0,l}(t',P)$$

Therefore we can rewrite β and β' into, respectively, b and b' (this is possible because we have an inclusion between the *directed paths*, not just the paths). We can then rewrite b and b' into true if we are on the then branch of b and b' (i.e. $\mathsf{x} = \mathsf{l}$), and false if we are on the else branch (i.e. $\mathsf{x} = \mathsf{r}$). Finally we get rid of true and false using R, and check that the resulting proof verifies (5.19) and the induction invariants.

Step 2 b. Then we show that we can assume that (ii) holds through some proof rewriting, while maintaining invariant (iv).

Let $(a,a'),(b,b') \leq_{\mathsf{cs}\sim\mathsf{cs}}^{\mathsf{h}_0} (t,P)$ such that $a \equiv b$ and they are on the same branch l. Since they are on the same branch, we can extract a proof $Q \vdash^{\mathsf{npf}} a, a\sim a', b'$. Moreover $a\downarrow_R, a'\downarrow_R, b'\downarrow_R$ are if-free, therefore by Lemma 5.17 we have $a' \equiv b'$. We then do our standard proof cut elimination to get rid of the duplicate. We need to make sure that this preserves invariant (iv): this follows from the fact that invariant (iv) holds for P at depth $n+1$ and that the cut takes place at depth n.

Step 3 We then show that (iii) holds. Let $b^{\mathsf{h}_0}, b'^{\mathsf{h}_0}$ be such that $\mathsf{extract}_{\mathsf{x}_0}(h,P) \vdash^{\mathsf{npf}} b^{\mathsf{h}_0} \sim b'^{\mathsf{h}_0}$. We know that:

$$b^{\mathsf{h}_0} \equiv C\left[\left(\boxed{b^{h_{\mathsf{l}}}\,|\,b^{h_{\mathsf{r}}}}_{b^h}\right)_{h\in H^{\mathsf{h}_0}} \diamond \left(D_l^{\mathsf{h}_0}\left[(\beta)_{\beta\leq_{\mathsf{c}}^{\mathsf{h}_0,l}(t,P)} \diamond (\gamma)_{\gamma\leq_{\mathsf{l}}^{\mathsf{h}_0,l}(t,P)}\right]\right)_{l\in L^{\mathsf{h}_0}}\right]$$

where $H^{\mathsf{h}_0} = \mathsf{cs\text{-}pos}(\mathsf{extract}_{\mathsf{x}_0}(h_0,P))$ and $L^{\mathsf{h}_0} = \mathsf{label}(\mathsf{extract}_{\mathsf{x}_0}(h_0,P))$.

To prove that for all l, for all $\gamma \leq_{\mathsf{l}}^{\mathsf{h}_0,l} (t,P)$, we have $b^{\mathsf{h}_0}\downarrow_R \in \mathsf{leave\text{-}st}(\gamma\downarrow_R)$, we only need to show that the hypotheses of Lemma 5.16 hold for b^{h_0} (then we do the same thing with b'^{h_0} to show that for all $\gamma' \leq_{\mathsf{l}}^{\mathsf{h}_0,l} (t',P)$ we have $b'^{\mathsf{h}_0}\downarrow_R \in \mathsf{leave\text{-}st}(\gamma'\downarrow_R)$):

- (5.16.i): the only difficulty lies in proving that for all $\beta \leq_{\mathsf{c}}^{\mathsf{h}_0,l} (t,P)$, $\mathsf{cond\text{-}st}(\beta\downarrow_R) \cap \mathsf{leave\text{-}st}(\beta\downarrow_R) = \emptyset$, which is shown in Proposition 5.21.
- (5.16.ii): this is a consequence of the fact that (5.19) holds for P.
- (5.16.iii): for pairs in $(\mathsf{cs\text{-}path}^{\mathsf{h}_0,l}(t,P))^2$ this was shown in **Step 2 b**. For couples of positions in $D_l^{\mathsf{h}_0} \times D_l^{\mathsf{h}_0}$ we have a proof cut elimination (which we already described in Section 5.11.3): let $p < p'$ be the positions in b^{h_0} of $\beta_0, \beta_1 \leq_{\mathsf{c}}^{\mathsf{h}_0,l} (t,P)$ on the same branch such that $\mathsf{leave\text{-}st}(\beta_0) \cap \mathsf{leave\text{-}st}(\beta_1) \neq \emptyset$. By Proposition 5.11 we know that $\beta_0 \equiv \beta_1$. Let β_0', β_1' be the conditionals at positions, respectively, p and p' in b'^{h_0}. We know that $(\beta_0,\beta_0'),(\beta_1,\beta_1') \leq_{\mathsf{c}}^{\mathsf{h}_0,l} (t\sim t',P)$. We can extract from P a proof of:

  $$\beta_0, \beta_0 \sim \beta_0', \beta_1'$$

  in $\mathfrak{F}(\mathsf{FA_s}^* \cdot \mathsf{Dup}^* \cdot \overline{\mathsf{CCA}_2})$, hence using Lemma 5.2 we get that $\beta_0' \equiv \beta_1'$. Therefore we can do the following proof cut elimination: if p' is on the then branch of p then we can rewrite β_1 and β_1' into true in, respectively, b^{h_0} and b'^{h_0}. We then rewrite the two terms using R to remove the true conditionals. This yields a new proof Q in proof normal form, such that (5.19) and the induction invariants hold. We do a similar cut elimination with false if p' is in the else of p.

  Finally the result proven at **Step 2** shows that we do not have cross cases $\mathsf{cs\text{-}path}^{\mathsf{h}_0,l}(t,P) \times D_l^{\mathsf{h}_0}$.

- (5.16.iv): this is a consequence of Corollary 5.2.(i).
- (5.16.v): this is a consequence of Lemma 5.14.

Step 4 We conclude by showing that we can get rid of the $\{\overline{\mathsf{BFA}}(b,b')\}$ applications.

Using Corollary 5.2.(ii) and the proof Q constructed at **Step 3**, we know that for all $\gamma,\gamma' \leq_{\mathsf{l}}^{\mathsf{h}_0,l} (t,Q)$, $\gamma \equiv \gamma'$ (and the same holds for (t',Q)). Therefore there is a proof cut elimination that allows us to remove all $\{\overline{\mathsf{BFA}}(b,b')\}$ applications, by rewriting:

$$D_l\Big[\_ \diamond (\gamma)_{\gamma \leq_{\mathsf{l}}^{\mathsf{h}_0,l}(t,Q)}\Big] \quad \text{and} \quad D_l\Big[\_ \diamond (\gamma')_{\gamma \leq_{\mathsf{l}}^{\mathsf{h}_0,l}(t',Q)}\Big]$$

into, respectively, γ_0 and γ_0' (where $\gamma_0 \leq_{\mathsf{l}}^{\mathsf{h}_0,l} (t,Q)$ and $\gamma_0' \leq_{\mathsf{l}}^{\mathsf{h}_0,l} (t',Q)$).

Conclusion To conclude, we can first observe that the properties (a),(b) and (c) are implied by, respectively, (i), (ii) and (iii) for $n=0$. The proof that (d) (resp. (e)) holds is exactly the same than the one we did at **Step 2** (resp. **Step 3**). ■

5.12 Bounding the Basic Terms

5.12.1 α-Bounded Conditionals

We are ready to do the final proof cut eliminations, which will yield derivation of bounded size w.r.t. $|t\downarrow_R|+|t'\downarrow_R|$. To bound the size of cut-free derivations, we are going to bound the size of all normalized basic terms and case-study conditionals appearing in such derivations. To do this, we first introduce the notion of (t,P)-α-bounded terms, where $P \vdash^{\mathsf{npf}} t \sim t'$, and then prove that (t,P)-α-bounded terms are of bounded size w.r.t. $|t\downarrow_R|+|t'\downarrow_R|$. Basically, a term β in $\leq_{\mathsf{bt}}^{\mathsf{h},l}(t,P)$ or $\mathsf{cs\text{-}path}^{\mathsf{h},l}(t,P)$ is (t,P)-α-bounded if we are in one of the following case:

- β is a normalized basic term, and β has a leaf term appearing in $\mathsf{st}(t\downarrow_R)$. Since β is uniquely characterized by its leaf terms, this bound β.
- Let β' be the term matching β on the *right*. If β' shares a leaf term with $\mathsf{st}(t'\downarrow_R)$, then, by the previous observation, β' is bounded. Since β and β' differ only by the content of their encryptions, this also bound β.
- If β is a case-study conditional (i.e. in $\mathsf{cs\text{-}path}^{\mathsf{h},l}(t,P)$), and if there exists a (t,P)-α-bounded normalized basic term ε such that β appears in ε's leaf terms. Indeed, since ε is bounded, it has finitely many leaf terms, which are of bounded size. Hence β is also of bounded size.
- If β is a normalized basic terms used in the sub-proof of $b \sim b'$, where b and b' are (t,P)-α-bounded case-study conditionals, and if b appears in β's leaf terms. Again, since β is uniquely characterized by any of its leaf terms, and since b is bounded, we know that β is bounded.
- Finally, if β is a decryption guard of some decryption oracle call d, where d appears in a (t,P)-α-bounded normalized basic term ζ. Since ζ is bounded, and since β is a sub-term of ζ, the term β is also bounded.

We formally define what is a (t,P)-α-bounded terms.

Definition 5.57. For all $P \vdash^{\mathsf{npf}} t \sim t'$, the set of (t,P)-α-bounded terms is the smallest subset of:

$$\left\{\beta \mid \exists \mathsf{h}, l.\, \beta \leq_{\mathsf{bt}}^{\mathsf{h},l} (t,P)\right\} \cup \left\{b \mid \exists \mathsf{h}.\, b \in \mathsf{cs\text{-}path}^{\mathsf{h},l}(t,P)\right\}$$

such that for all h, l, for all β $(\leq_{\mathsf{bt}}^{\mathsf{h},l} \cup \mathsf{cs\text{-}path}^{\mathsf{h},l})$ (t,P), β is (t,P)-α-bounded if:

- **Base case:** $\mathsf{h} = \epsilon$ and $\mathsf{leave\text{-}st}(\beta\downarrow_R) \cap \mathsf{st}(t\downarrow_R) \neq \emptyset$.
- **Base case:** $\mathsf{h} = \epsilon$ and there exists β' such that:

  $$(\beta,\beta')\ (\leq_{\mathsf{l}\sim\mathsf{l}}^{\epsilon,l} \cup \leq_{\mathsf{c}\sim\mathsf{c}}^{\epsilon,l} \cup \mathsf{cs\text{-}path}^{\epsilon,l})\ (t \sim t', P)$$

  and $\mathsf{leave\text{-}st}(\beta'\downarrow_R) \cap \mathsf{st}(t'\downarrow_R) \neq \emptyset$.

- **Inductive case, same label:** $\beta \in \mathsf{cs\text{-}path}^{\mathsf{h},l}(t,P)$ and there exists $\varepsilon \leq_{\mathsf{bt}}^{\mathsf{h},l} (t,P)$ such that ε is (t,P)-α-bounded and $\beta \in \mathsf{leave\text{-}st}(\varepsilon \downarrow_R)$.
- **Inductive case, different labels:** $\beta \leq_{\mathsf{bt}}^{\mathsf{h},l} (t,P)$, there exists h' such that $\mathsf{h} \in \mathsf{cs\text{-}pos}(\mathsf{h}')$ and $b \in \mathsf{cs\text{-}path}^{\mathsf{h}',l}(t,P)$ such that b is (t,P)-α-bounded and $b \in \mathsf{leave\text{-}st}(\beta \downarrow_R)$.
- **Inductive case, guard:** $\beta \leq_{\mathsf{bt}}^{\mathsf{h},l} (t,P)$, there exists $\varepsilon \leq_{\mathsf{bt}}^{\mathsf{h},l} (t,P)$ such that:
  - $\varepsilon \equiv B[\vec{w}, (\alpha_i)_i, (\mathsf{dec}_j)_j]$ is (t,P)-α-bounded.
  - β is a guard of a $\mathcal{S}_l^P$-decryption oracle call $d \in (\mathsf{dec}_j)_j$.

We continue our proof cut eliminations, starting from the derivations constructed in Lemma 5.18. We let $P \vdash_\alpha^{\mathsf{npf}} t \sim t'$ be the restriction of $\vdash^{\mathsf{npf}}$ to derivations satisfying the properties guaranteed by Lemma 5.18 which use only (t,P)-α-bounded terms. Moreover, we require that no basic conditionals appears twice on the same branch.

Definition 5.58. For all proof P, term t, t', we write $P \vdash_\alpha^{\mathsf{npf}} t \sim t'$ if:

(I) $P \vdash^{\mathsf{npf}} t \sim t'$ and the properties (a) to (e) of Lemma 5.18 hold.

(II) The following sets are sets of, respectively, (t,P)-α-bounded and (t',P)-α-bounded terms:

$$\left\{\beta \mid \exists \mathsf{h}, l.\, \beta \leq_{\mathsf{bt}}^{\mathsf{h},l} (t,P')\right\} \cup \left\{b \mid \exists \mathsf{h}.\, b \leq_{\mathsf{cs}}^{\mathsf{h}} (t,P')\right\}$$
$$\left\{\beta' \mid \exists \mathsf{h}, l.\, \beta' \leq_{\mathsf{bt}}^{\mathsf{h},l} (t',P')\right\} \cup \left\{b' \mid \exists \mathsf{h}.\, b' \leq_{\mathsf{cs}}^{\mathsf{h}} (t',P')\right\}$$

(III) For every $l \in \mathsf{label}(\epsilon)$, for every path $\vec{\rho}$ of $\mathcal{S}_l^P$-normalized basic conditional from the root of t to some leave, $\vec{\rho}$ does not contain any duplicates. The same property must hold for t'.

We now prove the last proof cut elimination lemma.

Lemma 5.19. *$\vdash_\alpha^{npf}$ is complete for $\vdash^{npf}$.*

Proof. Let P be such that $P \vdash^{\mathsf{npf}} t \sim t'$, where P is obtained using Lemma 5.18. Therefore P satisfies the item (I) of Definition 5.58. Now, we are going to build from P a proof P' of $t \sim t'$ that satisfies the item (II) and (III) of Definition 5.58.

We are going to show that, if there exists β in:

$$\left\{\beta \mid \exists \mathsf{h}, l.\, \beta \leq_{\mathsf{bt}}^{\mathsf{h},l} (t,P')\right\} \cup \left\{b \mid \exists \mathsf{h}.\, b \leq_{\mathsf{cs}}^{\mathsf{h}} (t,P')\right\}$$

such that β is not (t,P)-α-bounded, then there is a cut elimination removing β (we describe the cut elimination used later in the proof). Moreover, the resulting proof will have a smaller number of basic terms which are not (t,P)-α-bounded, hence we will conclude by induction. First, we want to pick a term β maximal for a carefully chosen relation.

Order $<_g$ Let $<_g$ be the transitive closure of the relation $\ll_g$ on:

$$\bigcup_{\mathsf{h} \in \mathsf{index}(P)} \left\{(\beta, \mathsf{h}) \mid \exists l. \beta \leq_{\mathsf{bt}}^{\mathsf{h},l} (t,P)\right\} \cup \bigcup_{\mathsf{h} \in \mathsf{index}(P)} \left\{(b, \mathsf{h}) \mid \exists l. b \in \mathsf{cs\text{-}path}^{\mathsf{h},l}(t,P)\right\}$$

defined by:

$$(\zeta, \mathsf{h}) \ll_g (\zeta', \mathsf{h}') \text{ iff } \begin{cases} \mathsf{h} = \mathsf{h}' \wedge \zeta, \zeta' \leq_{\mathsf{bt}}^{\mathsf{h},l} (t,P) \wedge\ \zeta \text{ is a guard of some decryption oracle call } d \in \mathsf{st}(\zeta') \\ \mathsf{h} = \mathsf{h}' \wedge \zeta \in \mathsf{cs\text{-}path}^{\mathsf{h},l}(t,P) \wedge \zeta' \leq_{\mathsf{bt}}^{\mathsf{h},l} (t,P) \wedge \zeta \in \mathsf{leave\text{-}st}(\zeta' \downarrow_R) \\ \mathsf{h} > \mathsf{h}' \wedge \zeta \leq_{\mathsf{bt}}^{\mathsf{h},l} (t,P) \wedge \zeta' \in \mathsf{cs\text{-}path}^{\mathsf{h}',l}(t,P) \wedge \zeta' \in \mathsf{leave\text{-}st}(\zeta \downarrow_R) \end{cases}$$

First we show that $<_g$ is a strict order. As it is transitive, we just need to show that it is an antisymmetric relation. For all h, the restriction $<_g^{\mathsf{h}}$ of $<_g$ to:

$$\left\{(\beta, \mathsf{h}) \mid \exists l. \beta \leq_{\mathsf{bt}}^{\mathsf{h},l} (t,P)\right\} \cup \left\{(b, \mathsf{h}) \mid \exists l. b \in \mathsf{cs\text{-}path}^{\mathsf{h},l}(t,P)\right\}$$

is a strict order, as it is included in the embedding relation. To show that $<_g$ is a strict order on its full domain, we simply use the facts that for all h, $<_g^{\mathsf{h}}$ is a strict order and that when we go from the domain of $<_g^{\mathsf{h}}$ to the domain of $<_g^{\mathsf{h}'}$, we have $\mathsf{h}' > \mathsf{h}$.

W.l.o.g. we assume that (β, h) is maximal for $<_g$ among the set of terms that are not (t, P)-α-bounded. Consider an arbitrary l such that $\mathsf{h} \in \mathsf{h\text{-}branch}(l)$. Since β is not (t, P)-α-bounded, we know that if β is a guard of some decryption oracle call $d \in \mathsf{st}(\zeta)$ with $\zeta \leq_{\mathsf{bt}}^{\mathsf{h},l} (t, P)$, then ζ is not (t, P)-α-bounded. By maximality of β, it follows that if $\beta \leq_{\mathsf{bt}}^{\mathsf{h},l} (t, P)$ then β is not a decryption guard of any $\zeta \leq_{\mathsf{bt}}^{\mathsf{h},l} (t, P)$.

Case $\mathsf{h} = \epsilon$ First we are going to describe what to do for $\mathsf{h} = \epsilon$. From Lemma 5.18.(e), we know that for every $l \in \mathsf{label}(P)$, for all $\gamma \leq_{\mathsf{l}}^{\epsilon,l} (t, P)$, the basic term γ is (t, P)-α-bounded. Therefore $\beta \not\leq_{\mathsf{l}}^{\epsilon,l} (t, P)$. Moreover, from Lemma 5.18.(d) we get that $\beta \leq_{\mathsf{c}}^{\epsilon,l} (t, P)$ and $\beta \in \mathsf{cs\text{-}path}^{\epsilon,l}(t, P)$ are mutually exclusive. Putting everything together, we have three cases:

(i) either $\beta \left(\not\leq_{\mathsf{l}}^{\epsilon,l} \cup \leq_{\mathsf{c}}^{\epsilon,l}\right) (t, P)$ and $\beta \notin \mathsf{cs\text{-}path}^{\epsilon,l}(t, P)$.

(ii) or $\beta \left(\not\leq_{\mathsf{l}}^{\epsilon,l} \cup \not\leq_{\mathsf{c}}^{\epsilon,l}\right) (t, P)$ and $\beta \in \mathsf{cs\text{-}path}^{\epsilon,l}(t, P)$.

(iii) $\beta \left(\not\leq_{\mathsf{l}}^{\epsilon,l} \cup \not\leq_{\mathsf{c}}^{\epsilon,l}\right) (t, P)$ and $\beta \notin \mathsf{cs\text{-}path}^{\epsilon,l}(t, P)$.

We first focus on case (i). We explain how to deal with (ii) and (iii) later.

- **(i), Part 1** Assume that we are in case i). Let β' be such that $(\beta, \beta') \left(\leq_{\mathsf{c}\sim\mathsf{c}}^{\epsilon,l}\right) (t \sim t', P)$. Since β is not (t, P)-α-bounded we know that for all $u \in \mathsf{leave\text{-}st}(\beta \downarrow_R)$, for all $u' \in \mathsf{leave\text{-}st}(\beta' \downarrow_R)$, u and u' are spurious in, respectively, t and t'. We let:

$$t \equiv C\left[\vec{b}_{cs} \diamond D_l\left[(\beta_i)_{i\in J} \diamond (\gamma_m)_{m\in M}\right], \Delta\right]$$
$$t' \equiv C\left[\vec{b}'_{cs} \diamond D_l\left[(\beta'_i)_{i\in J} \diamond (\gamma'_m)_{m\in M}\right], \Delta'\right]$$

  where, for every $i \in J$, $(\beta_i, \beta'_i) \leq_{\mathsf{c}\sim\mathsf{c}}^{\epsilon,l} (t \sim t', P)$, and for every $m \in M$, $(\gamma_m, \gamma'_m) \leq_{\mathsf{l}\sim\mathsf{l}}^{\epsilon,l} (t \sim t', P)$. Moreover, we assume that for every $i \in J$, the hole $[]_i$ (which is mapped to β_i) appears exactly once in D_l. We define the set of indices $I = \{i \in J \mid \beta \equiv \beta_i\}$. Using Corollary 5.2.(i), we know that:

$$I = \{i \in J \mid \mathsf{leave\text{-}st}(\beta \downarrow_R) \cap \mathsf{leave\text{-}st}(\beta_i \downarrow_R) \neq \emptyset\}$$

  We know that we have a proof of $(\beta_i)_{i\in I} \sim (\beta'_i)_{i\in I}$ in the fragment $\mathfrak{F}(\mathsf{FA_s}^* \cdot \mathsf{Dup}^* \cdot \overline{\mathrm{CCA}_2})$. Therefore:

$$\forall b, b' \in \{\beta'_i \mid i \in I\}, b \equiv b' \equiv \beta' \tag{5.21}$$

  Indeed, if $|I| = 1$ then this is obvious, and if $|I| > 1$ we use Lemma 5.2 (since all the terms on the left are the same). We let $I' = \{i \in J \mid \beta' \equiv \beta'_i\}$. Using the same proof than for I, we know that $I' = \{i \in J \mid \mathsf{leave\text{-}st}(\beta' \downarrow_R) \cap \mathsf{leave\text{-}st}(\beta'_i \downarrow_R) \neq \emptyset\}$. We deduce from this that:

$$\forall b, b' \in \{\beta_i \mid i \in I'\}, b \equiv b' \equiv \beta \tag{5.22}$$

  From (5.21) we get that $I \subseteq I'$ and conversely from (5.22) we get that $I' \subseteq I$. Therefore we have the equality $I = I'$.

- **(i), Part 2** For every $i \notin I$, using Lemma 5.12 on β we know that there exists $\tilde{\beta}_i[]$ such that:

$$\tilde{\beta}_i[\beta] \equiv \beta_i \qquad \text{and} \qquad \mathsf{leave\text{-}st}(\beta \downarrow_R) \cap \mathsf{cond\text{-}st}(\tilde{\beta}_i[] \downarrow_R) = \emptyset$$

  Similarly, for every $m \in M$, there exists $\tilde{\gamma}_m[]$ such that:

$$\tilde{\gamma}_m[\beta] \equiv \gamma_m \qquad \text{and} \qquad \mathsf{leave\text{-}st}(\beta \downarrow_R) \cap \mathsf{cond\text{-}st}(\tilde{\gamma}_m[] \downarrow_R) = \emptyset$$

  Then we have:

$$\begin{aligned} t &\equiv C\left[\vec{b}_{cs} \diamond \left(D_l\left[(\beta_i)_{i\in J} \diamond (\gamma_m)_{m\in M}\right], \Delta\right)\right] \\ &\equiv C\left[\vec{b}_{cs} \diamond \left(D_l\left[\left((\beta)_{i\in I}, (\tilde{\beta}_i[\beta])_{i\notin I}\right) \diamond (\tilde{\gamma}_m[\beta])_{m\in M}\right], \Delta\right)\right] \end{aligned}$$

Let $C_\beta[\vec{b}_\beta \diamond \vec{u}_\beta] \equiv \beta\downarrow_R$. We have:

$$\begin{aligned} & D_l\left[\left((\beta)_{i\in I}, (\tilde{\beta}_i[\beta])_{i\notin I}\right) \diamond (\tilde{\gamma}_m[\beta])_{m\in M}\right] \\ =_R\ & \mathsf{if}\ C_\beta[\vec{b}_\beta \diamond \vec{u}_\beta]\ \mathsf{then}\ D_l\left[\left((\mathsf{true})_{i\in I}, (\tilde{\beta}_i[\mathsf{true}])_{i\notin I}\right) \diamond (\tilde{\gamma}_m[\mathsf{true}])_{m\in M}\right] \\ & \qquad \mathsf{else}\ D_l\left[\left((\mathsf{false})_{i\in I}, (\tilde{\beta}_i[\mathsf{false}])_{i\notin I}\right) \diamond (\tilde{\gamma}_m[\mathsf{false}])_{m\in M}\right] \end{aligned}$$

Since $\vec{u}_\beta = \mathsf{leave\text{-}st}(\beta\downarrow_R)$, for every $u \in \vec{u}_\beta$, $i \in J$ and $m \in M$, we know that $u \notin \mathsf{cond\text{-}st}(\tilde{\beta}_i[]\downarrow_R)$ and $u \notin \mathsf{cond\text{-}st}(\tilde{\gamma}_m[]\downarrow_R)$. Let $\vec{\rho}$ be the conditionals appearing on the path from the root of t to $D_l[\_]$. Using Lemma 5.18.(d), we know that $\vec{u}_\beta \cap \vec{\rho} = \emptyset$. Let $(u_o)_{o\in O}$ be such that $\vec{u} \equiv (u_o)_{o\in O}$. By applying Lemma 5.15 to all u we know that:

$$\begin{aligned} & C\left[\vec{b}_{cs} \diamond \begin{pmatrix} \mathsf{if}\ C_\beta\left[\vec{b}_\beta \diamond \vec{u}_\beta\right]\ \mathsf{then}\ D_l\left[\left((\mathsf{true})_{i\in I}, (\tilde{\beta}_i[\mathsf{true}])_{i\notin I}\right) \diamond (\tilde{\gamma}_i[\mathsf{true}])_m\right] \\ \qquad\qquad \mathsf{else}\ D_l\left[\left((\mathsf{false})_{i\in I}, (\tilde{\beta}_i[\mathsf{false}])_{i\notin I}\right) \diamond (\tilde{\gamma}_i[\mathsf{false}])_m\right], \Delta \end{pmatrix}\right] \\ =_R\ & C\left[\vec{b}_{cs} \diamond \begin{pmatrix} \mathsf{if}\ C_\beta\left[\vec{b}_\beta \diamond (\mathsf{true})_o\right]\ \mathsf{then}\ D_l\left[\left((\mathsf{true})_{i\in I}, (\tilde{\beta}_i[\mathsf{true}])_{i\notin I}\right) \diamond (\tilde{\gamma}_i[\mathsf{true}])_m\right] \\ \qquad\qquad \mathsf{else}\ D_l\left[\left((\mathsf{false})_{i\in I}, (\tilde{\beta}_i[\mathsf{false}])_{i\notin I}\right) \diamond (\tilde{\gamma}_i[\mathsf{false}])_m\right], \Delta \end{pmatrix}\right] \\ =_R\ & C\left[\vec{b}_{cs} \diamond \left(D_l\left[\left((\mathsf{true})_{i\in I}, (\tilde{\beta}_i[\mathsf{true}])_{i\notin I}\right) \diamond (\tilde{\gamma}_i[\mathsf{true}])_m\right], \Delta\right)\right] \end{aligned} \tag{5.23}$$

- **(i), Part 2.b** We do exactly the same thing on the other side: for all $i \notin I$ we know that there exists $\tilde{\beta}'_i[]$ such that:

$$\tilde{\beta}'_i[\beta'] \equiv \beta'_i \qquad \text{and} \qquad \mathsf{leave\text{-}st}(\beta'\downarrow_R) \cap \mathsf{cond\text{-}st}(\tilde{\beta}'_i[]\downarrow_R) = \emptyset$$

And, for every $m \in M$, there exists $\tilde{\gamma}'_m[]$ such that:

$$\tilde{\gamma}'_m[\beta'] \equiv \gamma'_m \qquad \text{and} \qquad \mathsf{leave\text{-}st}(\beta'\downarrow_R) \cap \mathsf{cond\text{-}st}(\tilde{\gamma}'_m[]\downarrow_R) = \emptyset$$

Then by the same reasoning we have:

$$\begin{aligned} t' \equiv\ & C\left[\vec{b}'_{cs} \diamond \left(D_l\left[(\beta'_i)_i \diamond (\gamma'_m)_{m\in M}\right], \Delta'\right)\right] \\ \equiv\ & C\left[\vec{b}'_{cs} \diamond \left(D_l\left[\left((\beta')_{i\in I}, (\tilde{\beta}'_i[\beta'])_{i\notin I}\right) \diamond (\tilde{\gamma}'_m[\beta'])_{m\in M}\right], \Delta'\right)\right] \\ =_R\ & C\left[\vec{b}'_{cs} \diamond \left(D_l\left[\left((\mathsf{true})_{i\in I}, (\tilde{\beta}'_i[\mathsf{true}])_{i\notin I}\right) \diamond (\tilde{\gamma}'_m[\mathsf{true}])_{m\in M}\right], \Delta'\right)\right] \end{aligned} \tag{5.24}$$

Observe that corresponding sub-terms of (5.23) and (5.24) can be matched to corresponding sub-terms of t and t'. It is straightforward to build a proof of the equivalence of (5.23) and (5.24) using P, except for the CCA_2 applications side-conditions. We argue why the side-conditions carry over from the derivation P later in the proof.

- **(ii) and (iii)** The case (ii) works similarly to the case (i), except that we use Lemma 5.17 instead of Lemma 5.2. The case (iii) is exactly like the case (i) when taking $I = \emptyset$.

Case $\mathsf{h} \neq \epsilon$ In that case, thanks to Lemma 5.18.(a), we know that $\beta \not\leq_{\mathsf{c}}^{\mathsf{h},l} (t, P)$. We have three cases:

(a) either $\beta \leq_{\mathsf{i}}^{\mathsf{h},l} (t, P)$: using Lemma 5.18.(c), there exists h_0, b^h such that $\mathsf{h} \in \mathsf{cs\text{-}pos}(\mathsf{h}_0)$, $b^h \in \mathsf{cs\text{-}path}^{\mathsf{h}_0,l}(t, P)$ and $(b^h\downarrow_R) \in \mathsf{leave\text{-}st}(\beta\downarrow_R)$. Since $\mathsf{h} \in \mathsf{cs\text{-}pos}(\mathsf{h}_0)$ implies that $\mathsf{h}_0 < \mathsf{h}$, we know that $\beta <_g b^h$. We then have two cases. Either b^h is (t, P)-α-bounded, and then using the inductive case for different labels of the definition of (t, P)-α-bounded terms, we know that β is (t, P)-abounded. Absurd. Or b^h is not (t, P)-α-bounded, which contradicts the maximality of β among the set of terms which are not (t, P)-abounded. Absurd.

(b) either $\beta \not\leq_{\mathsf{i}}^{\mathsf{h},l} (t, P)$ and $\beta \in \mathsf{cs\text{-}path}^{\mathsf{h},l}(t, P)$: this case is done exactly like case (ii).

(c) either $\beta \not\leq_{\mathsf{i}}^{\mathsf{h},l} (t, P)$ and $\beta \notin \mathsf{cs\text{-}path}^{\mathsf{h},l}(t, P)$: this case is done exactly like case (iii).

Valid Proof Rewriting We do the rewritings described above for every h such that (β, h) is maximal for $<_g$, and for every l such that $\beta \leq^{\mathsf{h},l}_{\mathsf{bt}} (t, P)$ or $\beta \in \mathsf{cs\text{-}path}^{\mathsf{h},l}(t, P)$, *simultaneously*. It remains to check that this is a valid cut elimination. The only difficulty lies in checking that all the side-conditions of the CCA_2 axiom hold. This is tedious, but here are the key ingredients:

- β is not a guard, and the encryptions that need to be guarded in a decryption are invariant by our proof cut elimination. Therefore decryptions that were well-guarded before are still well-guarded after the cut.
- We did the proof rewriting simultaneously for all h such that (β, h) is maximal for $<_g$. Consider h' such that (β, h') is not maximal for $<_g$: then there exists h such that (β, h) is maximal for $<_g$ and $\mathsf{h} < \mathsf{h}'$. Therefore, the sub-proof at index h' is removed by the proof rewriting. This ensure that, for all branch l where a rewriting occurred, we removed all occurrences of β. Therefore, if an encryption used to contain β then all occurrences of this encryption have been rewritten in the same way. This guarantees that the freshness condition on encryption randomness still holds.
- The length constraints on encryption oracle calls still holds thanks to the branch invariance property of the length predicate $\mathsf{EQL}(\_, \_)$.

Conclusion This concludes the proof of the second bullet point of the definition $\vdash^{\mathsf{npf}}_{\alpha}$. The third bullet point is much simpler. We want to show that for all $l \in \mathsf{label}(\epsilon)$, for every path $\vec{\rho}$ of $\mathcal{S}^P_l$-normalized basic conditional from the root of t to some leave, $\vec{\rho}$ does not contain any duplicates. We show this by proof cut elimination as follows: let $(\beta, \beta'_0) \leq^{\epsilon,l}_{\mathsf{c}\sim\mathsf{c}} (t, P)$ and $(\beta, \beta'_1) \leq^{\epsilon,l}_{\mathsf{c}\sim\mathsf{c}} (t, P)$, using Lemma 5.2 we have $\beta'_0 \equiv \beta'_1$. Since they are on the same branch, one may rewrite the lowest occurrence of β and β'_0 into their then branch (we could also use the else branch). This yield a smaller proof, and one can check that all the other properties are invariant of this proof cut elimination. We directly concludes by induction. ■

5.12.2 Bounding the Number of Nested Basic Conditionals

We use the previous lemma to bound the number of basic conditionals appearing in a proof $P \vdash^{\mathsf{npf}}_{\alpha} t \sim t'$. Looking at the definition of (t, P)-α-bounded terms, one may try to show that for every $\beta \in (\leq^{\mathsf{h},l}_{\mathsf{bt}} (t, P) \cup \mathsf{cs\text{-}path}^{\mathsf{h},l}(t, P))$, if β is (t, P)-α-bounded then there exists $u \in \mathsf{leave\text{-}st}(\beta \downarrow_R)$ such that $u \in \mathsf{st}(t \downarrow_R) \cup \mathsf{st}(t' \downarrow_R)$. Since $\mathsf{st}(t \downarrow_R) \cup \mathsf{st}(t' \downarrow_R)$ is finite, and since a basic term is uniquely characterized by any of its leaves, this would allow us to bound the number of basic terms appearing in $P \vdash^{\mathsf{npf}}_{\alpha} t \sim t'$.

Unfortunately, this is not always the case. Indeed, consider $(\beta, \beta') \leq^{\mathsf{h},l}_{\mathsf{c}} (t \sim t', P)$ such that β' has a leaf term appearing in t', but β shares no leaf term with β' nor t:

$$\mathsf{leave\text{-}st}(\beta \downarrow_R) \cap \mathsf{leave\text{-}st}(\beta' \downarrow_R) = \emptyset \qquad \mathsf{leave\text{-}st}(\beta \downarrow_R) \cap \mathsf{st}(t \downarrow_R) = \emptyset \qquad \mathsf{leave\text{-}st}(\beta' \downarrow_R) \cap \mathsf{st}(t' \downarrow_R) \neq \emptyset$$

β' is α-bounded since it shares a leaf term with t', and using the second case, β is α-bounded too. But β shares no leaf term with t and t'.

Still, we can bound β. Since $(\beta, \beta') \leq^{\mathsf{h},l}_{\mathsf{c}} (t \sim t', P)$, we observe that $\beta \equiv B[\vec{w}, (\alpha_i)_i, (\mathsf{dec}_j)_j]$ and $\beta' \equiv B[\vec{w}, (\alpha'_i)_i, (\mathsf{dec}'_j)_j]$. Using the fact that $\mathsf{leave\text{-}st}(\beta' \downarrow_R) \cap \mathsf{st}(t' \downarrow_R)$ and that β is a $\mathcal{S}_l$-normalized basic term, we know that every leaf $u \in \mathsf{leave\text{-}st}(\beta \downarrow_R)$ is in $\mathsf{st}(t' \downarrow_R)$, *modulo the content of the $\mathcal{S}_l$-encryption oracle calls*. This motivate the introduction of the notion of *leaf frame*.

Leaf frame Let β be a $\mathcal{S}_l$-normalized basic term, and $u, v \in \mathsf{leave\text{-}st}(\beta \downarrow_R)$ be leaf terms of β. Then u and v only differ by their encryptions. That is, if one replace all the zero decryptions $\mathbf{0}(\mathsf{dec}(\_, \mathsf{sk}))$ by $\mathsf{dec}(\_, \mathsf{sk})$, and all the leaves of encryptions $\{m\}^{\mathsf{n}}_{\mathsf{pk}}$ by $\{[]_\alpha\}^{\mathsf{n}}_{\mathsf{pk}}$ (where α is the unique term of $\mathcal{E}_l$ such that $\alpha \equiv \{\_\}^{\mathsf{n}}_{\mathsf{pk}}$) in u and in v then you get the same context. We formalize this below, and use it to generalize Proposition 5.11.

Definition 5.59. Let $P \vdash^{\mathsf{npf}}_{\alpha} t \sim t'$ and l be a branch label in $\mathsf{label}(P)$. We define the left *leaf frame* $\mathsf{l\text{-}frame}^P_l$ of $\beta \in (\leq^{\mathsf{h},l}_{\mathsf{bt}} (t, P) \cup \mathsf{cs\text{-}path}^{\mathsf{h},l}(t, P))$ inductively as follows:

$$\mathsf{l\text{-}frame}^P_l(s) \equiv \begin{cases} \{[]_\alpha\}^{\mathsf{n}}_{\mathsf{pk}} & \text{if } \exists \alpha \equiv \{m\}^{\mathsf{n}}_{\mathsf{pk}} \in \mathcal{E}^P_l \wedge s \equiv \{\_\}^{\mathsf{n}}_{\mathsf{pk}} \\ \mathsf{dec}(\mathsf{l\text{-}frame}^P_l(s), \mathsf{sk}) & \text{if } \mathsf{sk} \in \mathcal{K}^P_l \wedge s \equiv \mathbf{0}(\mathsf{dec}(s, \mathsf{sk})) \\ \mathsf{l\text{-}frame}^P_l(v) & \text{if } s \equiv \mathsf{if}\ b\ \mathsf{then}\ u\ \mathsf{else}\ v \\ f((\mathsf{l\text{-}frame}^P_l(u_i))_i) & \text{otherwise} \end{cases}$$

We also let $\underline{\mathsf{l\text{-}frame}}_l^P(\beta)$ be $\mathsf{l\text{-}frame}_l^P(\beta)$ where we make every hole variable appear at most once, by replacing a hole variable $[]_\alpha$ occurring at position p in β by $[]_{\alpha,p}$.

We define the right *leaf frame* $\mathsf{r\text{-}frame}_l^P$ (and its underlined version $\underline{\mathsf{r\text{-}frame}}_l^P$) of $\beta \in (\leq_{\mathsf{bt}}^{\mathsf{h},l}(t', P) \cup \mathsf{cs\text{-}path}^{\mathsf{h},l}(t', P))$, using $\mathcal{E}_l'^P$ instead of $\mathcal{E}_l^P$.

Remark 5.12. We have two remarks:

- We state some results only for $\mathsf{l\text{-}frame}$. The corresponding results for $\mathsf{r\text{-}frame}$ are obtained by symmetry.
- The hole variables in $\underline{\mathsf{l\text{-}frame}}_l^P(\beta)$ are annotated by both the position p of the hole *and* the encryption α that appears at p in β. By consequence, if two normalized basic terms β and β' are such that $\underline{\mathsf{l\text{-}frame}}_l^P(\beta)$ and $\underline{\mathsf{l\text{-}frame}}_l^P(\beta')$ share a hole variable $[]_{\alpha,p}$, it means that β and β' contain the *same encryption* α *at the same position* p. This is crucial, as we want $\underline{\mathsf{l\text{-}frame}}_l^P$ to uniquely characterize normalized basic terms. □

Example 5.20. For all $\mathcal{S}_l^P$-decryption oracle call dec guarding $\mathsf{dec}(s[(\alpha_i)_i, (\mathsf{dec}_j)_j], \mathsf{sk})$, if for all i, $\alpha_i \equiv \{\_\}_{\mathsf{pk}_i}^{\mathsf{n}_i}$ then:

$$\mathsf{l\text{-}frame}_l^P(\mathsf{dec}) \equiv \mathsf{dec}\Big(s\Big[\big(\{[]_{\alpha_i}\}_{\mathsf{pk}_i}^{\mathsf{n}_i}\big)_i, \big(\mathsf{l\text{-}frame}_l^P(\mathsf{dec}_j)\big)_j\Big], \mathsf{sk}\Big)$$

We also give an example of $\underline{\mathsf{l\text{-}frame}}_l^P$. Assuming that $\alpha_0 \equiv \{A\}_{\mathsf{pk}}^{\mathsf{n}_0}$ and $\alpha_1 \equiv \{B\}_{\mathsf{pk}}^{\mathsf{n}_1}$ are encryptions in $\mathcal{E}_l^P$:

$$\underline{\mathsf{l\text{-}frame}}_l^P\left(\langle \alpha_0\,,\, \langle \alpha_1\,,\, \alpha_0\rangle\rangle\right) \equiv \langle \{[]_{\alpha_0,00}\}_{\mathsf{pk}}^{\mathsf{n}_0}\,,\, \langle \{[]_{\alpha_1,100}\}_{\mathsf{pk}}^{\mathsf{n}_1}\,,\, \{[]_{\alpha_0,110}\}_{\mathsf{pk}}^{\mathsf{n}_0}\rangle\rangle$$ □

Proposition 5.22. *Let $P \vdash_\alpha^{npf} t \sim t'$ and $l \in \mathsf{label}(P)$. Let b be an if-free term in R-normal form. For every $\mathcal{S}_l$-normalized basic terms γ, if $b \in \mathsf{leave\text{-}st}(\gamma \downarrow_R)$ then $\mathsf{l\text{-}frame}_l^P(b) \equiv \mathsf{l\text{-}frame}_l^P(\gamma)$.*

Proof. This is by induction on the size of γ. ■

Proposition 5.23. *Let $P \vdash_\alpha^{npf} t \sim t'$ and $l \in \mathsf{label}(P)$. For every $\mathcal{S}_l$-normalized basic terms β, β', if $\mathsf{l\text{-}frame}_l^P(\beta) \equiv \mathsf{l\text{-}frame}_l^P(\beta')$ then $\beta \equiv \beta'$.*

Proof. The proof is exactly the same than for Proposition 5.11. ■

Proposition 5.24. *Let $P \vdash_\alpha^{npf} t \sim t'$ and $l \in \mathsf{label}(P)$. For all h, if $(b, b') \leq_{\mathsf{cs}\sim\mathsf{cs}}^{\mathsf{h},l} (t \sim t', P)$ then there exists h' and $(\gamma, \gamma')\,(\leq_{\mathsf{c}\sim\mathsf{c}}^{\mathsf{h}',l} \cup \leq_{\mathsf{l}\sim\mathsf{l}}^{\mathsf{h}',l})\,(t \sim t', P)$ such that $b \in \mathsf{leave\text{-}st}(\gamma \downarrow_R)$ and $b' \in \mathsf{leave\text{-}st}(\gamma' \downarrow_R)$.*

Proof. Let h, x be such that $\mathsf{h} = h_{\mathsf{x}}$. Let $h_0 \in \mathsf{cs\text{-}pos}(\mathsf{extract}_{\mathsf{x}}(h, P))$ and x_0 be such that x_0 is the direction taken in l at position h_0, and such that $Q = \mathsf{extract}_{\mathsf{x}_0}(h_0, P)$ is a proof of $b \sim b'$.

Using the fact that the sub-proofs of $\mathsf{CS}_\square$ conditionals of P do not use the $\overline{\mathsf{BFA}}$ rule, we know that Q lies in the fragment:

$$\mathfrak{F}(\mathsf{CS}_\square \cdot \mathsf{FA_s}^* \cdot \mathsf{Dup}^* \cdot \overline{\mathrm{CCA}_2})$$

Let $(\gamma, \gamma') \leq_{\mathsf{l}\sim\mathsf{l}}^{\epsilon,l} (b \sim b', Q)$. Using the property (c) of Lemma 5.18 (which holds thanks to $\vdash_\alpha^{\mathsf{npf}}$), we know that $b \in \mathsf{leave\text{-}st}(\gamma \downarrow_R)$ and $b \in \mathsf{leave\text{-}st}(\gamma' \downarrow_R)$. ■

Proposition 5.25. *Let $P \vdash_\alpha^{npf} t \sim t'$ and $l \in \mathsf{label}(P)$. For all h, if $(\beta, \beta')\,(\leq_{\mathsf{c}\sim\mathsf{c}}^{\mathsf{h},l} \cup \leq_{\mathsf{l}\sim\mathsf{l}}^{\mathsf{h},l} \cup \mathsf{cs\text{-}path}_\sim^{\mathsf{h},l})\,(t \sim t', P)$ then $\mathsf{l\text{-}frame}_l^P(\beta) \equiv \mathsf{r\text{-}frame}_l^P(\beta')$.*

Proof. First we deal with the case $(\beta, \beta')\,(\leq_{\mathsf{c}\sim\mathsf{c}}^{\mathsf{h},l} \cup \leq_{\mathsf{l}\sim\mathsf{l}}^{\mathsf{h},l})\,(t \sim t', P)$. We know that we can extract a proof Q (from P) such that $Q \vdash_\alpha^{\mathsf{npf}} \beta \sim \beta'$ and Q is in the fragment $\mathfrak{F}(\mathsf{FA_s}^* \cdot \mathsf{Dup}^* \cdot \overline{\mathrm{CCA}_2})$. The result follows from the definitions of $\mathsf{l\text{-}frame}_l^P$ and $\mathsf{r\text{-}frame}_l^P$.

Now we deal with the case $(\beta, \beta')\,(\mathsf{cs\text{-}path}_\sim^{\mathsf{h},l})\,(t \sim t', P)$. Using Proposition 5.24 we know that there exists h' and $(\gamma, \gamma')\,(\leq_{\mathsf{c}\sim\mathsf{c}}^{\mathsf{h}',l} \cup \leq_{\mathsf{l}\sim\mathsf{l}}^{\mathsf{h}',l})\,(t \sim t', P)$ such that $\beta \in \mathsf{leave\text{-}st}(\gamma \downarrow_R)$ and $\beta' \in \mathsf{leave\text{-}st}(\gamma' \downarrow_R)$. Since β is if-free and in R-normal form, we obtain that $\mathsf{l\text{-}frame}_l^P(\beta) \equiv \mathsf{l\text{-}frame}_l^P(\gamma)$ by applying Proposition 5.22. Similarly $\mathsf{r\text{-}frame}_l^P(\beta') \equiv \mathsf{r\text{-}frame}_l^P(\gamma')$. Moreover, from the previous case, we get that $\mathsf{l\text{-}frame}_l^P(\gamma) \equiv \mathsf{r\text{-}frame}_l^P(\gamma')$. Hence $\mathsf{l\text{-}frame}_l^P(\beta) \equiv \mathsf{r\text{-}frame}_l^P(\beta')$. ■

Proposition 5.26. *Let $P \vdash_\alpha^{npf} t \sim t'$ and $l \in \mathsf{label}(P)$. For every $\mathcal{S}_l$-normalized basic terms β, β', $\mathsf{l\text{-}frame}_l^P(\beta) \equiv \mathsf{l\text{-}frame}_l^P(\beta')$ if and only if $\underline{\mathsf{l\text{-}frame}}_l^P(\beta) \equiv \underline{\mathsf{l\text{-}frame}}_l^P(\beta')$.*

Proof. This is obvious, since the hole variable annotations in $\underline{\mathsf{l\text{-}frame}}_l^P$ uniquely characterize both the position of the hole and the encryption appearing at this position. ■

Proposition 5.27. *Let $P \vdash^{\mathsf{npf}}_{\alpha} t \sim t'$ and $l \in \mathsf{label}(P)$. For every $\mathcal{S}_l$-normalized basic terms β, β' and substitutions θ, θ', if $\underline{\mathsf{l\text{-}frame}}_l^P(\beta)\theta \equiv \underline{\mathsf{l\text{-}frame}}_l^P(\beta')\theta'$ then $\underline{\mathsf{l\text{-}frame}}_l^P(\beta) \equiv \underline{\mathsf{l\text{-}frame}}_l^P(\beta')$.*

Proof. We prove this by induction on the size of β. The base case is trivial, lets deal with the inductive case. Let β and β' be $\mathcal{S}_l^P$-normalized basic terms, we know that $\beta \equiv B[\vec{w}, (\alpha_i)_i, (\mathsf{dec}_j)_j]$ where:

- for every i, $\alpha_i \equiv \{m_i\}^{\mathsf{n}_i}_{\mathsf{pk}_i} \in \mathcal{E}_l^P$.
- for every j, dec_j is a decryption oracle call for $\mathsf{dec}(s_j, \mathsf{sk}_j)$ in $\mathcal{D}_l^P$.

Similarly, we have a decomposition of β' into $B'[\vec{w}', (\alpha'_i)_i, (\mathsf{dec}'_j)_j]$. By definition of $\mathsf{l\text{-}frame}_l^P$, and using the fact that $\mathsf{fresh}(\mathcal{R}_l^P; \vec{w})$, we have:

$$\mathsf{l\text{-}frame}_l^P(\beta) \equiv B[\vec{w}, (\{[]_{\alpha_i}\}^{\mathsf{n}_i}_{\mathsf{pk}_i})_i, \mathsf{dec}(\mathsf{l\text{-}frame}_l^P(s_j), \mathsf{sk}_j)]$$

Similarly:

$$\mathsf{l\text{-}frame}_l^P(\beta') \equiv B'[\vec{w}', (\{[]_{\alpha'_i}\}^{\mathsf{n}'_i}_{\mathsf{pk}'_i})_i, \mathsf{dec}(\mathsf{l\text{-}frame}_l^P(s'_j), \mathsf{sk}'_j)]$$

We have three cases:

- Either $\beta \equiv \{m\}^{\mathsf{n}}_{\mathsf{pk}} \in \mathcal{E}_l^P$. Then $\underline{\mathsf{l\text{-}frame}}_l^P(\beta) \equiv \{[]_{\beta,0}\}^{\mathsf{n}}_{\mathsf{pk}}$. By definition of $\mathsf{l\text{-}frame}$, and using the fact that $\underline{\mathsf{l\text{-}frame}}_l^P(\beta)\theta \equiv \underline{\mathsf{l\text{-}frame}}_l^P(\beta')\theta'$, we get that β' is of the form $\{m'\}^{\mathsf{n}}_{\mathsf{pk}}$. We deduce from the freshness side condition of n that $m' \equiv m$.
- Or $\beta \equiv \mathsf{dec}$ where dec is a $\mathcal{S}_l^P$-decryption oracle call guarding $\mathsf{dec}(s, \mathsf{sk})$. Then $\underline{\mathsf{l\text{-}frame}}_l^P(\beta) \equiv \mathsf{dec}(\underline{\mathsf{l\text{-}frame}}_l^P(s), \mathsf{sk})\mu$, where μ is the substitution that lifts positions of s into positions of $\mathsf{dec}(s, \mathsf{sk})$, i.e. for every $\alpha \in \mathcal{E}_l^P$ and position $p \in \mathsf{pos}(s)$:

  $$\mu([]_{\alpha,p}) \equiv []_{\alpha,0\cdot p}$$

  By definition of $\mathsf{l\text{-}frame}$, and using the fact that $\underline{\mathsf{l\text{-}frame}}_l^P(\beta)\theta \equiv \underline{\mathsf{l\text{-}frame}}_l^P(\beta')\theta'$ and that β' is a $\mathcal{S}_l^P$-normalized basic term, we get that β' is also some dec' where dec' is a $\mathcal{S}_l^P$-decryption oracle call guarding $\mathsf{dec}(s', \mathsf{sk})$.

  Moreover we have $\underline{\mathsf{l\text{-}frame}}_l^P(s)\mu\theta \equiv \mathsf{l\text{-}frame}_l^P(s')\mu\theta$, and s, s' are $\mathcal{S}_l^P$-normalized basic terms. Hence by induction hypothesis $\underline{\mathsf{l\text{-}frame}}_l^P(s) \equiv \underline{\mathsf{l\text{-}frame}}_l^P(s')$, which concludes this case.
- Or we are not in one of the two cases above. Then, there exists $f \in \mathcal{F}_{\setminus \mathsf{if},\mathbf{0}}$ s.t. $\beta \equiv f(u_1, \ldots, u_n)$ and $\beta' \equiv f(u'_1, \ldots, u'_n)$, where $u_1, \ldots, u_n$ and $u'_1, \ldots, u'_n$ are $\mathcal{S}_l^P$-normalized basic term. Hence $\underline{\mathsf{l\text{-}frame}}_l^P(\beta)$ and $\underline{\mathsf{l\text{-}frame}}_l^P(\beta')$ both starts with the function symbol f.

  Moreover, if we let, for very $1 \leq i \leq n$, μ_i be the lifting substitution such that, for every $\alpha \in \mathcal{E}_l^P$ and position p, $\mu_i([]_{\alpha,p}) \equiv []_{\alpha,i\cdot p}$, then:

  $$\underline{\mathsf{l\text{-}frame}}_l^P(\beta) \equiv f(\underline{\mathsf{l\text{-}frame}}_l^P(u_1)\mu_1, \ldots, \underline{\mathsf{l\text{-}frame}}_l^P(u_n)\mu_n)$$

  $$\underline{\mathsf{l\text{-}frame}}_l^P(\beta') \equiv f(\underline{\mathsf{l\text{-}frame}}_l^P(u'_1)\mu_1, \ldots, \underline{\mathsf{l\text{-}frame}}_l^P(u'_n)\mu_n)$$

  We apply θ to the equations above, and use the fact that $\underline{\mathsf{l\text{-}frame}}_l^P(\beta)\theta \equiv \underline{\mathsf{l\text{-}frame}}_l^P(\beta')\theta$:

  $$\begin{aligned} f(\underline{\mathsf{l\text{-}frame}}_l^P(u_1)\mu_1\theta, \ldots, \underline{\mathsf{l\text{-}frame}}_l^P(u_n)\mu_n\theta) &\equiv \underline{\mathsf{l\text{-}frame}}_l^P(\beta)\theta \\ &\equiv \underline{\mathsf{l\text{-}frame}}_l^P(\beta')\theta \\ &\equiv f(\underline{\mathsf{l\text{-}frame}}_l^P(u'_1)\mu_1\theta, \ldots, \underline{\mathsf{l\text{-}frame}}_l^P(u'_n)\mu_n\theta) \end{aligned}$$

  Hence, for every $1 \leq i \leq n$, $\underline{\mathsf{l\text{-}frame}}_l^P(u_i)\mu_i\theta \equiv \underline{\mathsf{l\text{-}frame}}_l^P(u'_i)\mu_i\theta$. By induction hypothesis, we deduce that $\underline{\mathsf{l\text{-}frame}}_l^P(u_i) \equiv \underline{\mathsf{l\text{-}frame}}_l^P(u'_i)$. Therefore $\underline{\mathsf{l\text{-}frame}}_l^P(\beta) \equiv \underline{\mathsf{l\text{-}frame}}_l^P(\beta')$. ■

Definition 5.60. We let $<_{\mathsf{st}}$ be the strict, well-founded, subterm ordering.

Nested Sequences of Basic Conditionals We want to bound the number of nested basic conditional appearing in $P \vdash_\alpha^{\mathsf{npf}} t \sim t'$. Using the contrapositive of Proposition 5.23, we know that when $\beta <_{\mathsf{st}} \beta'$ we have $\mathsf{l\text{-}frame}_l^P(\beta) \not\equiv \mathsf{l\text{-}frame}_l^P(\beta')$. Moreover, using Proposition 5.26 and Proposition 5.27, we know that $\mathsf{l\text{-}frame}_l^P(\beta) \not\equiv \mathsf{l\text{-}frame}_l^P(\beta')$ implies that $\underline{\mathsf{l\text{-}frame}}_l^P(\beta)\theta \not\equiv \underline{\mathsf{l\text{-}frame}}_l^P(\beta')\theta'$ (for every substitutions θ, θ').

Therefore, for any sequence of nested $\mathcal{S}_l^P$-normalized basic conditionals:

$$\beta_1 <_{\mathsf{st}} \cdots <_{\mathsf{st}} \beta_n$$

and for any substitutions $\theta_1, \ldots, \theta_n$, we know that $(\underline{\mathsf{l\text{-}frame}}_l^P(\beta_i)\theta_i)_{1\leq i\leq n}$ is a sequence of pair-wise distinct terms. Tu use this, we prove that there there exists a sequence of substitutions $\theta_1, \ldots, \theta_n$ such that:

$$\left\{\underline{\mathsf{l\text{-}frame}}_l^P(\beta_1)\theta_1, \ldots, \underline{\mathsf{l\text{-}frame}}_l^P(\beta_n)\theta_n\right\} \subseteq \mathcal{B}(t, t')$$

where $\mathcal{B}(t, t')$ is a set of bounded size w.r.t. $|t|+|t'|$. Since the $(\mathsf{l\text{-}frame}_l^P(\beta_i)\theta_i)_{1\leq i\leq n}$ are pair-wise distinct, using a pigeon-hole argument we get that $n \leq |\mathcal{B}(t, t')|$.

We outline the end of this sub-section. First, we define the set of terms $\mathcal{B}(t, t')$, and show the existence of the substitutions $(\theta_i)_i$. Then, we bound the size of $\mathcal{B}(t, t')$. Finally, we bound the number of nested basic conditional n using a pigeon-hole argument.

Definition 5.61. Let u be an if-free term. We let $\zeta_{\mathcal{K}}(u)$ be the set of terms obtained from u by replacing some occurrences of $\mathbf{0}(\mathsf{dec}(w, \mathsf{sk}))$ by $\mathsf{dec}(w, \mathsf{sk})$ (where $\mathsf{sk} \in \mathcal{K}$), non-deterministically stopping at some encryptions. Formally:

$$\zeta_{\mathcal{K}}(u) = \begin{cases} \{\mathsf{dec}(v, \mathsf{sk}) \mid w \in v \in \zeta_{\mathcal{K}}(w)\} & \text{if } u \equiv \mathbf{0}(\mathsf{dec}(w, \mathsf{sk})) \text{ and } \mathsf{sk} \in \mathcal{K} \\ \{u\} \cup \{\{v\}^{\mathsf{n}_r}_{\mathsf{pk}(\mathsf{n})} \mid v \in \zeta_{\mathcal{K}}(m)\} & \text{if } u \equiv \{m\}^{\mathsf{n}_r}_{\mathsf{pk}(\mathsf{n})} \text{ and } \mathsf{sk}(\mathsf{n}) \in \mathcal{K} \\ \{f(v_1, \ldots, v_n) \mid \forall i, v_i \in \zeta_{\mathcal{K}}(u_i)\} & \text{otherwise, where } u \equiv f(u_1, \ldots, u_n) \end{cases}$$

Moreover, given a set of ground terms $\mathcal{S}$, we let $\mathsf{guards}_{\mathcal{K}}(\mathcal{S})$ be an over-approximation of the set of guards of terms in $\mathcal{S}$:

$$\mathsf{guards}_{\mathcal{K}}(\mathcal{S}) = \left\{\mathsf{eq}(s, \alpha) \mid \mathsf{dec}(s, \mathsf{sk}(\mathsf{n})) \in \mathcal{S} \wedge \alpha \equiv \{\_\}^{-}_{\mathsf{pk}(\mathsf{n})} \in \mathsf{st}(s) \wedge \mathsf{sk}(\mathsf{n}) \in \mathcal{K}\right\}$$

Definition 5.62. Let $\mathcal{S}_{\mathsf{k}}(t)$ be the set of private keys appearing in $t \downarrow_R$, i.e. $\mathcal{S}_{\mathsf{k}}(t) = \{\mathsf{sk}(\mathsf{n}) \mid \mathsf{sk}(\mathsf{n}) \in \mathsf{st}(t \downarrow_R)\}$. For every term t, we let $\mathcal{B}(t)$ be the set:

$$\mathcal{B}(t) = \bigcup_{\mathcal{K} \subseteq \mathcal{S}_{\mathsf{k}}(t)} \bigcup_{\substack{u \in \mathsf{st}(\mathsf{leave\text{-}st}(t \downarrow_R)) \\ \vee u \in \mathsf{st}(\mathsf{cond\text{-}st}(t \downarrow_R))}} \zeta_{\mathcal{K}}(u) \cup \mathsf{guards}_{\mathcal{K}}(\zeta_{\mathcal{K}}(u))$$

Moreover, we let $\mathcal{B}(t, t') = \mathcal{B}(t) \cup \mathcal{B}(t')$.

Proposition 5.28. *Let $P \vdash_\alpha^{\mathsf{npf}} t \sim t'$ and $l \in \mathsf{label}(P)$. Let β be a $\mathcal{S}_l^P$-normalized basic conditional. Then, for every $u \in \mathsf{leave\text{-}st}(\beta \downarrow_R)$, there exists θ such that $\underline{\mathsf{l\text{-}frame}}_l^P(\beta)\theta \in \zeta_{\mathcal{K}}(u)$.*

Proof. We show this by induction on $|\beta|$.

- If β is an encryption $\{m\}^{\mathsf{n}}_{\mathsf{pk}} \in \mathcal{E}_l^P$, then $\underline{\mathsf{l\text{-}frame}}_l^P(\beta) \equiv \{[]_{\beta,0}\}^{\mathsf{n}}_{\mathsf{pk}}$ and:

  $$\mathsf{leave\text{-}st}(\beta \downarrow_R) = \left\{\{v\}^{\mathsf{n}}_{\mathsf{pk}} \mid v \in \mathsf{leave\text{-}st}(m \downarrow_R)\right\}$$

  Let $u \in \mathsf{leave\text{-}st}(\beta \downarrow_R)$, there exists $u_m \in \mathsf{leave\text{-}st}(m \downarrow_R)$ such that $u \equiv \{u_m\}^{\mathsf{n}}_{\mathsf{pk}}$. Let θ be the substitution mapping $[]_{\beta,0}$ to u_m. Then:

  $$\underline{\mathsf{l\text{-}frame}}_l^P(\beta)\theta \equiv \{u_m\}^{\mathsf{n}}_{\mathsf{pk}} \equiv u \in \zeta_{\mathcal{K}_l^P}(u)$$

- If β is a decryption oracle call in $\mathcal{D}_l^P$ for $\mathsf{dec}(s, \mathsf{sk})$, the:

  $$\mathsf{leave\text{-}st}(\beta \downarrow_R) \subseteq \{\mathsf{dec}(u_s, \mathsf{sk}) \mid u_s \in \mathsf{leave\text{-}st}(s \downarrow_R)\} \cup \{\mathbf{0}(\mathsf{dec}(u_s, \mathsf{sk})) \mid u_s \in \mathsf{leave\text{-}st}(s \downarrow_R)\}$$

Let $u \in \mathsf{leave\text{-}st}(\beta \downarrow_R)$, there exists $u_s \in \mathsf{leave\text{-}st}(s \downarrow_R)$ such that $u \equiv \mathsf{dec}(u_s, \mathsf{sk})$ or $u \equiv \mathbf{0}(\mathsf{dec}(u_s, \mathsf{sk}))$. Since s is a $\mathcal{S}_l^P$-normalized basic term, by induction hypothesis we have θ such that $\underline{\mathsf{l\text{-}frame}}_l^P(s)\theta \in \zeta_{\mathcal{K}_l^P}(u_s)$. Moreover:

$$\underline{\mathsf{l\text{-}frame}}_l^P(\beta) \equiv \mathsf{dec}(\underline{\mathsf{l\text{-}frame}}_l^P(s)\mu, \mathsf{sk})$$

where μ is a renaming of hole variables. Let $\theta' = \mu^{-1}\theta$, then:

$$\underline{\mathsf{l\text{-}frame}}_l^P(\beta)\theta' \equiv \mathsf{dec}(\underline{\mathsf{l\text{-}frame}}_l^P(s)\mu\mu^{-1}\theta, \mathsf{sk}) \equiv \mathsf{dec}(\underline{\mathsf{l\text{-}frame}}_l^P(s)\theta, \mathsf{sk}) \in \zeta_{\mathcal{K}_l^P}(u)$$

- Otherwise, $\beta \equiv f(\beta_1, \ldots, \beta_n)$ where, for every $1 \leq i \leq n$, β_i is a $\mathcal{S}_l^P$-normalized basic term. Then, using the fact that β is a $\mathcal{S}_l^P$-normalized basic term, we check that:

$$\mathsf{leave\text{-}st}(\beta \downarrow_R) \subseteq \{f(v_1, \ldots, v_n) \mid \forall i, v_i \in \mathsf{leave\text{-}st}(\beta_i \downarrow_R)\}$$

Let $u \in \mathsf{leave\text{-}st}(\beta \downarrow_R)$, there exists $v_1, \ldots, v_n$ such that for every $1 \leq i \leq n$ $v_i \in \mathsf{leave\text{-}st}(\beta_i \downarrow_R)$ and $u \equiv f(v_1, \ldots, v_n)$. By induction hypothesis, there exists $\theta_1, \ldots, \theta_n$ such that for every $1 \leq i \leq n$:

$$\underline{\mathsf{l\text{-}frame}}_l^P(\beta_i)\theta_i \in \zeta_{\mathcal{K}_l^P}(v_i)$$

For very $1 \leq i \leq n$, let μ_i be the lifting substitution such that, for every $\alpha \in \mathcal{E}_l^P$ and position p, $\mu_i([]_{\alpha,p}) \equiv []_{\alpha,i\cdot p}$. Then:

$$\underline{\mathsf{l\text{-}frame}}_l^P(\beta) \equiv f(\underline{\mathsf{l\text{-}frame}}_l^P(\beta_1)\mu_1, \ldots, \underline{\mathsf{l\text{-}frame}}_l^P(\beta_n)\mu_n)$$

Observe that the substitutions $(\mu_i\theta_i)_{1\leq i\leq n}$ have disjoint domains. Let $\theta = \mu_1\theta_1 \ldots \mu_n\theta_n$. Then:

$$\underline{\mathsf{l\text{-}frame}}_l^P(\beta)\theta \equiv f(\underline{\mathsf{l\text{-}frame}}_l^P(\beta_1)\mu_1\theta_1, \ldots, \underline{\mathsf{l\text{-}frame}}_l^P(\beta_n)\mu_n\theta_n)$$

We know that f cannot be the function symbol $\mathbf{0}(\_)$ (since $\mathsf{FA}_{\setminus \mathbf{0}}$ cannot be applied on $\mathbf{0}(\_)$). It follows that:

$$f(\underline{\mathsf{l\text{-}frame}}_l^P(\beta_1)\mu_1\theta_1, \ldots, \underline{\mathsf{l\text{-}frame}}_l^P(\beta_n)\mu_n\theta_n) \in \zeta_{\mathcal{K}_l^P}(u) \qquad \blacksquare$$

We lift the previous result to α-bounded conditionals.

Lemma 5.20. *Let $P \vdash_\alpha^{npf} t \sim t'$, l a branch label in $\mathsf{label}(P)$, h a proof index and $\beta \in (\leq_{\mathsf{bt}}^{\mathsf{h},l}(t, P) \cup \mathsf{cs\text{-}path}^{\mathsf{h},l}(t, P))$. If β is (t, P)-α-bounded then there exists a substitution θ s.t. $\underline{\mathsf{l\text{-}frame}}_l^P(\beta)\theta \in \mathcal{B}(t, t')$.*

Proof. We prove this by induction on the well-founded order underlying the inductive definition of (t, P)-α-bounded terms.

- **Base case:** Assume $\mathsf{h} = \epsilon$ and $\mathsf{leave\text{-}st}(\beta \downarrow_R) \cap \mathsf{st}(t \downarrow_R) \neq \emptyset$. Let $u \in \mathsf{leave\text{-}st}(\beta \downarrow_R) \cap \mathsf{st}(t \downarrow_R)$, we have u in R-normal form and if-free, therefore $u \in \mathsf{st}(\mathsf{leave\text{-}st}(t \downarrow_R) \cup \mathsf{cond\text{-}st}(t \downarrow_R))$. Moreover, by Proposition 5.28, there exists θ such that $\underline{\mathsf{l\text{-}frame}}_l^P(\beta)\theta \in \zeta_{\mathcal{K}_l^P}(u)$. Hence $\underline{\mathsf{l\text{-}frame}}_l^P(\beta)\theta \in \mathcal{B}(t, t')$.
- **Base case:** Assume $\mathsf{h} = \epsilon$ and there exists β' such that:

$$(\beta, \beta')\ (\leq_{\mathsf{l}\sim\mathsf{l}}^{\epsilon,l} \cup \leq_{\mathsf{c}\sim\mathsf{c}}^{\epsilon,l} \cup \leq_{\mathsf{cs}\sim\mathsf{cs}}^{\epsilon})\ (t \sim t', P) \quad \text{and} \quad \mathsf{leave\text{-}st}(\beta' \downarrow_R) \cap \mathsf{st}(t' \downarrow_R) \neq \emptyset$$

By Proposition 5.25 we know that $\mathsf{l\text{-}frame}_l^P(\beta) \equiv \mathsf{r\text{-}frame}_l^P(\beta')$. By Proposition 5.26, we deduce that $\underline{\mathsf{l\text{-}frame}}_l^P(\beta) \equiv \underline{\mathsf{r\text{-}frame}}_l^P(\beta')$. From the previous case we know that there exists θ such that $\underline{\mathsf{r\text{-}frame}}_l^P(\beta')\theta \in \mathcal{B}(t')$. Therefore $\underline{\mathsf{l\text{-}frame}}_l^P(\beta)\theta \in \mathcal{B}(t')$.
- **Inductive case, same label:** Assume $\beta \in \mathsf{cs\text{-}path}^{\mathsf{h},l}(t, P)$ and that there exists $\varepsilon \leq_{\mathsf{bt}}^{\mathsf{h},l}(t, P)$ such that ε is (t, P)-α-bounded and $\beta \in \mathsf{leave\text{-}st}(\varepsilon \downarrow_R)$. By induction hypothesis we have θ such that $\underline{\mathsf{l\text{-}frame}}_l^P(\varepsilon)\theta \in \mathcal{B}(t, t')$. We know that β is if-free and in R-normal form and that ε is a $\mathcal{S}_l^P$-normalized basic term. Therefore, by Proposition 5.22, we have $\mathsf{l\text{-}frame}_l^P(\beta) \equiv \mathsf{l\text{-}frame}_l^P(\varepsilon)$. Hence, using Proposition 5.26, $\underline{\mathsf{l\text{-}frame}}_l^P(\beta)\theta \in \mathcal{B}(t, t')$.
- **Inductive case, different labels:** Similar to the previous case.
- **Inductive case, guard:** If there exists $\varepsilon \leq_{\mathsf{bt}}^{\mathsf{h},l}(t, P)$ such that:
  - $\varepsilon \equiv B[\vec{w}, (\alpha_i)_i, (\mathsf{dec}_j)_j]$ is (t, P)-α-bounded.

– β is a guard of a $\mathcal{S}_l^P$-decryption oracle call $d \in (\mathsf{dec}_j)_j$.

By induction hypothesis there exists θ such that $\underline{\mathsf{l\text{-}frame}}_l^P(\varepsilon)\theta \in \mathcal{B}(t,t')$. Moreover let $(\mathsf{pk}_i)_i$ and $(\mathsf{n}_i)_i$ be such that $\forall i, \alpha_i \equiv \{\_\}^{\mathsf{n}_i}_{\mathsf{pk}_i}$. Then:

$$\mathsf{l\text{-}frame}_l^P(\varepsilon) \equiv B\left[\vec{w}, \left(\{[]_{\alpha_i}\}^{\mathsf{n}_i}_{\mathsf{pk}_i}\right)_i, \left(\mathsf{l\text{-}frame}_l^P(\mathsf{dec}_j)\right)_j\right]$$

Therefore there exists a renaming of hole variables μ such that $\underline{\mathsf{l\text{-}frame}}_l^P(d)\mu\theta \in \mathsf{st}(\underline{\mathsf{l\text{-}frame}}_l^P(\varepsilon)\theta)$. Since $\mathcal{B}(t,t')$ is closed under st, this implies that:

$$\underline{\mathsf{l\text{-}frame}}_l^P(d)\mu\theta \in \mathcal{B}(t,t')$$

d is of the form $\mathsf{dec}(s,\mathsf{sk})$ where $\mathsf{sk} \in \mathcal{K}$. Since members of $\mathsf{guards}_{\mathcal{K}}(\_)$ are of the form $\mathsf{eq}(\_,\_)$, we know that there exists some $u \in \mathsf{st}(\mathsf{leave\text{-}st}(t\downarrow_R) \cup \mathsf{cond\text{-}st}(t\downarrow_R))$ such that $\underline{\mathsf{l\text{-}frame}}_l^P(d)\mu\theta \in \zeta_{\mathcal{K}}(u)$. Since β is a guard of d, β is of the form $\mathsf{eq}(s,\alpha)$ where α is an encryption under key pk (corresponding to sk) and randomness n appearing directly in s. It follows that:

$$\mathsf{l\text{-}frame}_l^P(d) \equiv \mathsf{dec}(\mathsf{l\text{-}frame}_l^P(s),\mathsf{sk}) \qquad \mathsf{l\text{-}frame}_l^P(\beta) \equiv \mathsf{eq}(\mathsf{l\text{-}frame}_l^P(s),\{[]_\alpha\}^{\mathsf{n}}_{\mathsf{pk}})$$

Since α appears directly in s, and since $\underline{\mathsf{l\text{-}frame}}_l^P(d)\mu\theta \in \zeta_{\mathcal{K}}(u)$, there exists θ' such that:

$$\underline{\mathsf{l\text{-}frame}}_l^P(\beta)\theta' \in \mathsf{guards}_{\mathcal{K}}(\zeta_{\mathcal{K}}(u)) \subseteq \mathcal{B}(t,t') \qquad \blacksquare$$

We now bound the size of $\mathcal{B}(t)$.

Proposition 5.29. *For every term t, for every $u \in \mathcal{B}(t)$, we have $|u| \leq |t\downarrow_R|$. Moreover:*

$$|\mathcal{B}(t)| \leq |t\downarrow_R|^2.2^{|t\downarrow_R|}$$

Proof. An over-approximation of the set of terms $\zeta_{\mathcal{K}}(u)$ is obtained from u by choosing a subset of positions of u where decryptions over keys in $\mathcal{K}$ occur, and removing $\mathbf{0}$ before the subterms at these positions (if there is one). Hence each element of $\zeta_{\mathcal{K}}(u)$ is of size at most $|u|$. Moreover, for every $u \in \mathsf{st}(\mathsf{leave\text{-}st}(t\downarrow_R) \cup \mathsf{cond\text{-}st}(t\downarrow_R))$, we have $u \in \mathsf{st}(t\downarrow_R)$, and therefore $|u| \leq |t\downarrow_R|$. Therefore the set $\zeta_{\mathcal{K}}(u)$ contains terms of size at most $|t\downarrow_R|$.

Let $\mathsf{dec}(s,\mathsf{sk}) \in \zeta_{\mathcal{K}}(u)$, then $|\mathsf{dec}(s,\mathsf{sk})| = |s|+3$ and for every α appearing in s:

$$|\mathsf{eq}(s,\alpha)| = |s| + |\alpha| + 1 \leq 2|s| + 1 \leq 2|\mathsf{dec}(s,\mathsf{sk})| \leq 2|t\downarrow_R|$$

Hence the set $\mathsf{guards}_{\mathcal{K}}(\zeta_{\mathcal{K}}(u))$ contains terms of size at most $2|t\downarrow_R|$. We deduce that for every $v \in \mathcal{B}(t)$, $|v| \leq 2|t\downarrow_R|$. Moreover, by upper-bounding the positions of $\mathsf{dec}(s,\mathsf{sk})$ where an encryption might be, there are at most $|s|-1 \leq |t\downarrow_R| - 1$ such α, independently of the set of keys $\mathcal{K}$. It follows that:

$$\left|\bigcup_{\mathcal{K}\subseteq\mathcal{S}_{\mathbf{k}}(t)} \mathsf{guards}_{\mathcal{K}}(\zeta_{\mathcal{K}}(u))\right| \leq |\zeta_{\mathcal{K}}(u)|.(|t\downarrow_R| - 1)$$

Independently of the set of keys $\mathcal{K}$ chosen, we have at most $|\mathsf{st}(t\downarrow_R)| \leq |t\downarrow_R|$ choices for u, and the set $\bigcup_{\mathcal{K}\subseteq\mathcal{S}_{\mathbf{k}}(t)} \zeta_{\mathcal{K}}(u)$ contains at most $2^{|u|} \leq 2^{|t\downarrow_R|}$ elements (we choose the positions where we remove $\mathbf{0}$s). Hence:

$$\begin{aligned}\left|\bigcup_{\mathcal{K}\subseteq\mathcal{S}_{\mathbf{k}}(t)} \zeta_{\mathcal{K}}(u) \cup \mathsf{guards}_{\mathcal{K}}(\zeta_{\mathcal{K}}(u))\right| &\leq \left|\bigcup_{\mathcal{K}\subseteq\mathcal{S}_{\mathbf{k}}(t)} \zeta_{\mathcal{K}}(u)\right| + \left|\bigcup_{\mathcal{K}\subseteq\mathcal{S}_{\mathbf{k}}(t)} \mathsf{guards}_{\mathcal{K}}(\zeta_{\mathcal{K}}(u))\right| \\ &\leq |\zeta_{\mathcal{K}}(u)| + (|t\downarrow_R| - 1).|\zeta_{\mathcal{K}}(u)| \leq |t\downarrow_R|.2^{|t\downarrow_R|}\end{aligned}$$

By consequence:

$$|\mathcal{B}(t)| \leq |t\downarrow_R|.|t\downarrow_R|.2^{|t\downarrow_R|} \leq |t\downarrow_R|^2.2^{|t\downarrow_R|} \qquad \blacksquare$$

Finally, we apply a pigeon-hole argument to bound the number of nested basic terms.

Lemma 5.21. *Let $P \vdash^{\mathsf{npf}}_{\alpha} t \sim t'$. Let l be a branch label in $\mathsf{label}(P)$, h a proof index. Let $(\beta_i)_{i\leq n}$ such that for all i, $\beta_i \leq^{\mathsf{h},l}_{\mathsf{bt}} (t,P)$. If $\beta_1 <_{\mathsf{st}} \cdots <_{\mathsf{st}} \beta_n$ then $n \leq |\mathcal{B}(t,t')|$.*

Proof. For every $i \neq j$, we know, using Proposition 5.23, that $\mathsf{l\text{-}frame}_l^P(\beta_i) \not\equiv \mathsf{l\text{-}frame}_l^P(\beta_j)$. By Proposition 5.26, we deduce that $\underline{\mathsf{l\text{-}frame}}_l^P(\beta_i) \not\equiv \underline{\mathsf{l\text{-}frame}}_l^P(\beta_j)$. Since $P \vdash_\alpha^{\mathsf{npf}} t \sim t'$, we know that for every i, β_i is (t,P)-α-bounded. Using Lemma 5.20, we deduce that for every i, there exists a substitution θ_i such that:

$$\underline{\mathsf{l\text{-}frame}}_l^P(\beta_i)\theta_i \in \mathcal{B}(t,t')$$

Using the contrapositive of Proposition 5.27, we have that for every $i \neq j$:

$$\underline{\mathsf{l\text{-}frame}}_l^P(\beta_i)\theta_i \not\equiv \underline{\mathsf{l\text{-}frame}}_l^P(\beta_j)\theta_j$$

Therefore, by a pigeon-hole argument, $n \leq |\mathcal{B}(t,t')|$. ■

5.12.3 Candidate Sequences

Let $P \vdash_\alpha^{\mathsf{npf}} t \sim t'$. For all $n \leq |\mathcal{B}(t,t')|$, we are going to define the set $\mathcal{U}_n$ of normalized basic terms that may appear in P using n nested basic terms. We then show that these sets are finite and recursive, and give an upper-bound on their size which does not depend on n. This allows us to conclude by showing that the existence of a proof using our (complete) strategy is decidable.

Definition 5.63. An α-context C is a context such that all holes appear below the encryption function symbol, with proper randomness and encryption key. More precisely, for every position $p \in \mathsf{pos}(C)$, if $C_{|p} \equiv \square$ then $p = p' \cdot 0$ and there exist two nonces n, n_r such that $C_{|p'} \equiv \{\square\}^{\mathsf{n}_r}_{\mathsf{pk(n)}}$.

Moreover, we require that every hole appears at most once.

Remark 5.13. For every $\beta \leq_{\mathsf{bt}}^{\mathsf{h},l} (t,P)$, the context $\mathsf{l\text{-}frame}_l^P(\beta)$ is an α-context. □

Let t and t' be two ground terms. We now define what is a *valid candidate sequence* $(\mathcal{U}_n, \mathcal{A}_n)_{n\in\mathbb{N}}$ for t, t'. Basically, $\mathcal{U}_n$ corresponds to basic terms at nested depth n that could appear, on the left, in a proof of $\vdash_\alpha^{\mathsf{npf}} t \sim t'$, while $\mathcal{A}_n$ is the set of left encryptions oracle calls built using basic terms in $\mathcal{U}_{n-1}$.

Definition 5.64. Let t, t' be two terms. A sequence of pairs of sets of ground terms $(\mathcal{U}_n, \mathcal{A}_n)_{n\in\mathbb{N}}$ is a *valid candidate sequence* for t, t' if:

- $\mathcal{U}_0 = \mathcal{B}(t,t')$ and $\mathcal{A}_0 = \emptyset$.
- For $n \geq 0$, $\mathcal{A}_{n+1}$ can be any set of terms that satisfies the following constraints (with the convention that $\mathcal{A}_{-1} = \emptyset$): $\mathcal{A}_{n+1}$ contains $\mathcal{A}_n$, and for all $\alpha \in \mathcal{A}_{n+1} \backslash \mathcal{A}_n$, $\alpha \equiv \{D[\vec{b} \diamond \vec{u}]\}^{\mathsf{n}_r}_{\mathsf{pk}(\mathsf{n}_p)}$ where:
  - $\vec{b} \cup \vec{u}$ are in $\mathcal{U}_{n-1}$ and there exists $\{\_\}^{\mathsf{n}_r}_{\_} \in \mathsf{st}(t\downarrow_R) \cup \mathsf{st}(t'\downarrow_R)$.
  - for every branch $\vec{\rho} \subseteq \vec{b}$ of $D[\vec{b} \diamond \vec{u}]$, $\vec{\rho}$ does not contain duplicates.
  - $\mathcal{A}_n$ does not contain any terms of the form $\{\_\}^{\mathsf{n}_r}_{\_}$.
- For $n > 0$, we let $\mathcal{U}_{n+1}$ is the set of term defined from $\mathcal{U}_n$ and $\mathcal{A}_n$ as follows: $\mathcal{U}_{n+1}$ contains $\mathcal{U}_n$, plus any element that can be obtained through the following construction:
  - Take a α-context C such that there exists θ with $C\theta \in \mathcal{B}(t,t')$.
  - Let $\square_1, \ldots, \square_a$ be the variables of C, and let $\alpha_1, \ldots, \alpha_a$ be encryptions in $\mathcal{A}_n$. For all $1 \leq k \leq a$, let s_i be such that $\{s_i\}_{\_} \equiv \alpha_i \in \mathcal{A}_n$.
  - Let $v_0 \equiv C[(s_i)_{1\leq i\leq a}]$. Then let v be the term obtained from v_0 as follows: take positions $p_1, \cdots, p_o \in \mathsf{pos}(C)$ such that for all $1 \leq i \leq o$, $C_{|p_i} \equiv \mathsf{dec}(\_, \mathsf{sk}_i)$ (where sk_i is a valid private key, i.e. of the form $\mathsf{sk}(\mathsf{n}_i)$); for every $1 \leq i \leq o$, replace in v_0 the subterm $\mathsf{dec}(s, \mathsf{sk})$ at position p by $D[\vec{g} \diamond \vec{w}]$, where $\vec{g}$ are terms in $\mathcal{U}_n$ of the form $\mathsf{eq}(s, \alpha)$ (with $\alpha \equiv \{\_\}^{\mathsf{n}_\alpha}_{\_} \in \mathcal{A}_n$ and α directly appears in s) and $\forall w \in \vec{w}$, $w \equiv \mathsf{dec}(s, \mathsf{sk})$ or $w \equiv \mathbf{0}(\mathsf{dec}(s, \mathsf{sk}))$.

Proposition 5.30. *Let $P \vdash_\alpha^{\mathsf{npf}} t \sim t'$. For $l \in \mathsf{label}(P)$, there exists a valid candidate sequence $(\mathcal{U}_n, \mathcal{A}_n)_{n\in\mathbb{N}}$ for t, t' such that:*

$$\bigcup_{\mathsf{h}} \leq_{\mathsf{bt}}^{\mathsf{h},l} (t,P) \subseteq \bigcup_{n<|\mathcal{B}(t,t')|} \mathcal{U}_n \qquad \text{and} \qquad \bigcup_{\mathsf{h}} \mathsf{cs\text{-}path}^{\mathsf{h},l}(t,P) \subseteq \bigcup_{n<|\mathcal{B}(t,t')|} \mathsf{leave\text{-}st}\,(\mathcal{U}_n \downarrow_R)$$

Proof. First, we show that there exists a valid candidate sequence such that the inclusion holds when taking the union over $\mathbb{N}$ on the right, and s.t. for every n, $\mathcal{A}_n$ contains only valid encryptions in $\mathcal{E}_l^P$, i.e.:

$$\mathcal{S} = \bigcup_{\mathsf{h}} \leq_{\mathsf{bt}}^{\mathsf{h},l} (t, P) \subseteq \bigcup_{n<+\infty} \mathcal{U}_n \qquad \text{and} \qquad \bigcup_{n\in\mathbb{N}} \mathcal{A}_n \subseteq \mathcal{E}_l^P \tag{5.25}$$

Before starting the construction of the valid candidate sequence, we make some observations: if one fixes $(\mathcal{A}_n)_{n\in\mathbb{N}}$, there is at most one sequence $(\mathcal{U}_n)_{n\in\mathbb{N}}$ such that $(\mathcal{U}_n, \mathcal{A}_n)_{n\in\mathbb{N}}$ is a valid candidate sequence.

Moreover this sequence is non-decreasing in $(\mathcal{A}_n)_{n\in\mathbb{N}}$. More precisely, if $(\mathcal{U}_n, \mathcal{A}_n)_{n\in\mathbb{N}}$ and $(\mathcal{U}'_n, \mathcal{A}'_n)_{n\in\mathbb{N}}$ are valid candidate sequences such that for every n, $\mathcal{A}_n \subseteq \mathcal{A}'_n$, then for every n, $\mathcal{U}_n \subseteq \mathcal{U}'_n$.

We now describe a procedure that recursively construct $\mathcal{S}' \subseteq \mathcal{S}$ and a valid candidate sequence $(\mathcal{U}_n, \mathcal{A}_n)_{n\in\mathbb{N}}$ such that $\mathcal{S}'$ is a subset of $\bigcup_{n\leq+\infty}\mathcal{U}_n$ (eventually, we will show that $\mathcal{S}' = \mathcal{S}$). Moreover we require $(\mathcal{A}_n)_{n\in\mathbb{N}}$ to be minimal in the following sense: if $\alpha \equiv C[\vec{b} \diamond \vec{u}]$ is in $\mathcal{A}_{n+1}\backslash\mathcal{A}_n$ then there exists $v \in \vec{b} \cup \vec{u}$ such that $v \in \mathcal{U}_n\backslash\mathcal{U}_{n-1}$ (in other words, we add new encryptions in $\mathcal{A}_n$ as soon as we can).

Initially we take $\mathcal{A}_n = \emptyset$ for every n, $(\mathcal{U}_n)_{n\in\mathbb{N}}$ such that $(\mathcal{U}_n, \mathcal{A}_n)_{n\in\mathbb{N}}$ is a valid candidate sequence and $\mathcal{S}' = \emptyset$. While $\mathcal{S}' \neq \mathcal{S}$, we pick an element β in $\mathcal{S}\backslash\mathcal{S}'$ such that β is minimal for $<_{\mathsf{st}}$ in $\mathcal{S}\backslash\mathcal{S}'$. Then we add β to $\mathcal{S}'$ and update $(\mathcal{A}_n)_{n\in\mathbb{N}}$ as follows:

Case 1 If β is minimal for $<_{\mathsf{st}}$ in $\mathcal{S}$, we have β of the form $B[\vec{w}, (\alpha_i)_{i\in I}, (\mathsf{dec}_j)_{j\in J}]$. By minimality of β, we have $I = \emptyset$ and for all $j \in J$, dec_j has no encryptions in $\mathcal{E}_l^P$, and by consequence no guards. It follows that β is if-free and in R-normal form, hence $\mathsf{l\text{-}frame}_l^P(\beta) \equiv \beta$. By consequence, using Lemma 5.20, we get that $\beta \in \mathcal{B}(t, t') = \mathcal{U}_0$ (since $\mathcal{U}_0$ does not depends on the sets $(\mathcal{A}_n)_{n\in\mathbb{N}}$).

Case 2 Let β such that for all $\beta' <_{\mathsf{st}} \beta$, $\beta' \in \mathcal{S}'$. Since $\mathcal{S}' \subseteq \cup_{n\in\mathbb{N}}\mathcal{U}_n$, and since $\{\beta' \mid \beta' <_{\mathsf{st}} \beta\}$ is finite, there exists n_m such that:

$$\{\beta' \mid \beta' <_{\mathsf{st}} \beta\} \cap \left(\leq_{\mathsf{bt}}^{\mathsf{h},l} (t, P) \cup \mathsf{cs\text{-}path}^{\mathsf{h},l}(t, P)\right) \subseteq \bigcup_{0\leq n\leq n_m} \mathcal{U}_n$$

From Lemma 5.20 we have a substitution θ such that:

$$\underline{\mathsf{l\text{-}frame}}_l^P(\beta)\theta \in \mathcal{B}(t, t')$$

We then just need to show that we can obtain β from $\underline{\mathsf{l\text{-}frame}}_l^P(\beta)$ using the procedure defining $\mathcal{U}_{n_m+1}$:

- For all encryption $\alpha \equiv \{m\}_{\mathsf{pk}}^{\mathsf{n}} \in \mathsf{st}(\beta) \cap \mathcal{E}_l^P$, we know that $m \equiv C[\vec{b} \diamond \vec{u}]$ where $\vec{b}, \vec{u} <_{\mathsf{st}} \beta$. Hence $\vec{b}, \vec{u}$ are in $\cup_{0\leq n\leq n_m}\mathcal{U}_n$. We then have two cases:
  - either $\cup_{n\in\mathbb{N}}\mathcal{A}_n$ already contains an encryption α' with randomness n. Since $\cup_{n\in\mathbb{N}}\mathcal{A}_n \subseteq \mathcal{E}_l^P$, and using the side-condition of the CCA$_2$ application, we know that $\alpha \equiv \alpha' \in \cup_{n\in\mathbb{N}}\mathcal{A}_n$. By minimality of the $(\mathcal{A}_n)_{n\in\mathbb{N}}$ we know that $\alpha \in \mathcal{A}_{n_m+1}$.
  - or $\cup_{n\in\mathbb{N}}\mathcal{A}_n$ does not contain an encryption with randomness n. Then we simply add α to $\mathcal{A}_{n'}$, where $n' \leq n_m + 1$ is the smallest possible: we know that there exists such a n' since adding α to $\mathcal{A}_n$ yields, after completion of the $(\mathcal{U}_n)_{n\in\mathbb{N}}$, a valid candidate sequence (one can check that for all branch $\vec{\rho}$ of $C[\vec{b} \diamond \vec{u}]$, $\vec{\rho}$ does not contain duplicates, using the third bullet point of the definition of $\vdash_\alpha^{\mathsf{npf}}$).

  Then we replace in $\underline{\mathsf{l\text{-}frame}}_l^P(\beta)$ the holes $[]_\alpha, \_$ by $\{C[\vec{b} \diamond \vec{u}]\}_{\mathsf{pk}}^{\mathsf{n}}$. This produce a term v_0.
- Finally we also replace in v_0 every occurrence of $\mathsf{dec}(\_, \mathsf{sk})$ or $\mathbf{0}(\mathsf{dec}(\_, \mathsf{sk}))$ in $\mathsf{st}(\mathsf{l\text{-}frame}_l^P(\beta))$ by the corresponding $\mathcal{S}_l^P$-decryption oracle call, which is possible since the guards $\vec{g}$ of this decryption oracle calls are such that $\vec{g} <_{\mathsf{st}} \beta$, hence are in $\cup_{0\leq n\leq n_m}\mathcal{U}_n$.

Conclusion We show that when $\mathcal{S} = \mathcal{S}'$ we have:

$$\mathcal{S} \cap \bigcup_{n<+\infty} \mathcal{U}_n \;=\; \mathcal{S} \cap \bigcup_{n<|\mathcal{B}(t,t')|} \mathcal{U}_n \tag{5.26}$$

Assume that $\mathcal{S} \cap \mathcal{U}_{|\mathcal{B}(t,t')|-1} \subsetneq \mathcal{S} \cap \mathcal{U}_{|\mathcal{B}(t,t')|}$, take $\beta \in \mathcal{S} \cap (\mathcal{U}_{|\mathcal{B}(t,t')|} \backslash \mathcal{U}_{|\mathcal{B}(t,t')|-1})$. We know that $\beta \equiv B[\vec{w}, (\alpha_i)_i, (\mathsf{dec}_j)_j]$ and that there is an encryption α in $(\alpha_i)_i$ or in the encryptions of the $(\mathsf{dec}_j)_j$ such that $\alpha \in \mathcal{A}_{|\mathcal{B}(t,t')|-1} \backslash \mathcal{A}_{|\mathcal{B}(t,t')|-2}$ (otherwise β would be in $\mathcal{S} \cap \mathcal{U}_{|\mathcal{B}(t,t')|-1}$). Let $\alpha \equiv \{C[\vec{b} \diamond \vec{u}]\}^{\mathsf{n}}_{\mathsf{pk}}$, by minimality of the $(\mathcal{A}_n)_{n\in\mathbb{N}}$ we know that there is some $v \in \vec{b} \cup \vec{u}$ such that $v \in \mathcal{U}_{|\mathcal{B}(t,t')|-1} \backslash \mathcal{U}_{|\mathcal{B}(t,t')|-2}$. Since β is in $\mathcal{S}$ and since v is a $\mathcal{S}_l^P$-normalized basic term appearing in β we know that $v \in \mathcal{S}$. Let $\beta_0 \equiv \beta$, $\beta_1 \equiv v$, we have $v \in \mathcal{S} \cap (\mathcal{U}_{|\mathcal{B}(t,t')|-1} \backslash \mathcal{U}_{|\mathcal{B}(t,t')|-2})$. By induction we can build a sequence of terms β_n, for $n \in \{0, \ldots, |\mathcal{B}(t,t')|\}$ such that for all $0 \leq n \leq |\mathcal{B}(t,t')|$, $\beta_n \in \mathcal{S} \cap (\mathcal{U}_{|\mathcal{B}(t,t')|-i} \backslash \mathcal{U}_{|\mathcal{B}(t,t')|-(i+1)})$ and $\beta_{n+1} <_{\mathsf{st}} \beta_n$ (with the convention $\mathcal{U}_{-1} = \emptyset$). We built a sequence of terms in $\mathcal{S}$, strictly ordered by $<_{\mathsf{st}}$ and of length $|\mathcal{B}(t,t')| + 1$. This contradicts Lemma 5.21. Absurd.

To finish, it remains to show that:

$$\bigcup_{\mathsf{h}} \mathsf{cs\text{-}path}^{\mathsf{h},l}(t,P) \subseteq \bigcup_{n<|\mathcal{B}(t,t')|} \mathsf{leave\text{-}st}\,(\mathcal{U}_n \downarrow_R)$$

Let b in $\bigcup_{\mathsf{h}} \mathsf{cs\text{-}path}^{\mathsf{h},l}(t,P)$. Using Proposition 5.24 we know that there exists $\gamma \leq^{\mathsf{h}',l}_{\mathsf{bt}} (t,P)$ such that $b \in \mathsf{leave\text{-}st}(\gamma \downarrow_R)$. Since $\gamma \in \bigcup_{n<|\mathcal{B}(t,t')|} \mathcal{U}_n \downarrow_R$, we have $b \in \bigcup_{n<|\mathcal{B}(t,t')|} \mathsf{leave\text{-}st}\,(\mathcal{U}_n \downarrow_R)$. ■

Proposition 5.31. *For all terms u, let $\mathcal{C}_u$ be the set of α-contexts:*

$$\mathcal{C}_u = \{C \mid \exists \theta.\, C\theta \equiv u \wedge \textit{every hole appears at most once}\}$$

and $\mathcal{C}_u^\alpha$ be $\mathcal{C}_u$ quotiented by the α-renaming of holes relation. Then $|\mathcal{C}_u^\alpha| \leq 2^{|u|}$.

Proof. The set of contexts $\mathcal{C}_u^\alpha$ can be injected in the subsets of positions of u as follows: for every context C, associate to C the set of positions of u such that $C_{|p}$ is a hole. This is invariant by α-renaming and uniquely characterizes C modulo hole renaming. It follows that there are less element of $\mathcal{C}_u^\alpha$ than subsets of $\mathsf{pos}(u)$, i.e. $2^{|\mathsf{pos}(u)|} = 2^{|u|}$. ■

Proposition 5.32. *Let t and t' be two ground terms, $N = |t \downarrow_R| + |t' \downarrow_R|$. For every valid candidate sequence $(\mathcal{U}_n, \mathcal{A}_n)_{n\in\mathbb{N}}$ and $n \in \mathbb{N}$:*

$$|\mathcal{A}_n| \leq N \qquad\qquad |\mathcal{U}_n| \leq N^2.2^{3.N}$$

Proof. For every n, $\mathcal{A}_n$ contains only terms of the form $\alpha \equiv \{m\}^{\mathsf{n}_r}_{\mathsf{pk}}$, where $\{\_\}^{\mathsf{n}_r}_{\_} \in \mathsf{st}(t \downarrow_R) \cup \mathsf{st}(t' \downarrow_R)$. Moreover, $\mathcal{A}_n$ cannot contain two encryptions using the same randomness. Therefore $|\mathcal{A}_n| \leq N$.

For every n, the only leeway we have while constructing the terms in $\mathcal{U}_n$ is in the choice of the α-context C, as the content of the encryptions is determined by $\mathcal{A}_{n-1}$, and the guards that are added are determined by $\mathcal{U}_{n-1}$. The α-context C is picked in the following set:

$$\bigcup_{u\in\mathcal{B}(t,t')} \mathcal{C}_u^\alpha$$

which, using Proposition 5.29 and Proposition 5.31, we can bound by:

$$\Big|\bigcup_{u\in\mathcal{B}(t,t')} \mathcal{C}_u^\alpha\Big| \leq \sum_{u\in\mathcal{B}(t,t')} |\mathcal{C}_u^\alpha| \leq \sum_{u\in\mathcal{B}(t,t')} 2^{2.N} \leq N^2.2^N.2^{2.N} = N^2.2^{3.N}$$ ■

Proposition 5.33. *Let t, t' be two ground terms and $N = |t \downarrow_R| + |t' \downarrow_R|$. For every valid candidate sequence $(\mathcal{U}_n, \mathcal{A}_n)_{n\in\mathbb{N}}$ and $n \in \mathbb{N}$:*

$$\forall u \in \bigcup_{n<|\mathcal{B}(t,t')|} \mathcal{U}_n,\ |u| \leq 2^{Q(N)\,.\,2^{4.N}}$$

Where $Q(X)$ is a polynomial of degree 4.

Proof. Even though there are at most $|\mathcal{B}(t,t')|.N^2.2^{3.N}$ distinct basic terms appearing in branch l at proof index h, these terms may be much larger. Let U_n (resp. A_n) be an upper bound on the size of a term in $\mathcal{U}_n$ (resp. $\mathcal{A}_n$). Then for every $0 \leq n < |\mathcal{B}(t,t')|$ and $\alpha \in \mathcal{A}_{n+1} \backslash \mathcal{A}_n$, α is of the form $\{C[\vec{b} \diamond \vec{u}]\}^{\mathsf{n}}_{\mathsf{pk}}$,

where $\vec{b}, \vec{u}$ are in $\mathcal{U}_n$ and C is such that no term appears twice on the same branch. Recall that we call branch the ordered list of *inner conditionals*, which does not include the final leaf. If follows that C is of depth at most $|\mathcal{U}_n| + 1$, and therefore has at most $2^{|\mathcal{U}_n|+2} - 1$ conditional and leaf terms. To bound $|C[\vec{b} \diamond \vec{u}]|$, we need to bound the size of each of its internal and leaf terms, which we do using U_n. We get:

$$\left|C[\vec{b} \diamond \vec{u}]\right| \leq |C| + |C| \,.\, U_n \leq 2.|C| \,.\, U_n \leq 2^{|\mathcal{U}_n|+3} \,.\, U_n$$

since U_n is greater than 1 (terms can not be of size 0). Therefore $|\alpha| \leq 4 + 2^{|\mathcal{U}_n|+3} \,.\, U_n$. Using the bound from Proposition 5.32, we can take:

$$A_n = 4 + 2^{N^2.2^{3.N}+3} \,.\, U_n$$

Now let $u \equiv C[(\alpha_i)_{i\in I}, (\mathsf{dec}_j)_{j\in J}]$ in $\mathcal{U}_{n+1} \backslash \mathcal{U}_n$. We know that $\forall i \in I, |\alpha_i| \leq A_n$. There are at most $|C|$ hole occurrences in C, hence $|I| \leq |C|$ and $|J| \leq |C|$. To bound $|u|$, we also need to bound the size of the decryption guards. There are at most N guards for each decryption (since only element of $\mathcal{A}_n$ may be guarded, and $|\mathcal{A}_n| \leq N$), and each guard is in $\mathcal{U}_n$, so of size bounded by U_n. Moreover, guarded decryptions have at most $N + 1$ leaf, where each life is of size at most $|C[(\alpha_i)_{i\in I}, ([])_{j\in J}]| + 1 \leq |C| + |I|.A_n + 1$. Hence every decryption's size is upper-bounded by:

$$N + N.U_n + (N + 1).(|C| + |I|.A_n + 1)$$

Finally $|C|$ is such that there there exists θ such that $C\theta \in \mathcal{B}(t, t')$, hence $|C| \leq 2.N$ using Proposition 5.29. Hence, assuming $U_n \geq N$ (which will be the case):

$$\begin{aligned} |C[(\alpha_i)_{i\in I}, (\mathsf{dec}_j)_{j\in J}]| &\leq |C| + |I|.A_n + |J|.(N + N.U_n + (N + 1).(|C| + |I|.A_n + 1)) \\ &\leq 2N + 2N.A_n + 2N.(N + N.U_n + (N + 1).(2N + 2N.A_n + 1)) \end{aligned}$$

Seen as a multi-variate polynomial in N, A_n and U_n, we have only monomials N, $N.A_n$, N^2, $N^2.U_n$, N^3 and $N^3.A_n$. Hence there exists a constant L such that:

$$u \leq L.N^3(A_n + U_n) \leq L.N^3(4 + 2^{N^2.2^{3.N}+3}.U_n + U_n)$$

Hence there exists some polynomial Q_0 of degree two such that $u \leq 2^{Q_0(N).2^{3N}}.U_n$. We let $U_0 = N$, and $U_{n+1} = 2^{Q_0(N).2^{3N}}.U_n$. Then:

$$U_{|\mathcal{B}(t,t')|-1} \leq 2^{|\mathcal{B}(t,t')|.Q_0(N).2^{3N}}.U_n \leq 2^{N^2.2^N.Q_0(N).2^{3N}}.U_n \leq 2^{N^2.Q_0(N).2^{4N}}.U_n$$

Hence we have a polynomial $Q(N) = N^2.Q_0(N)$, which is of degree four. ■

Corollary 5.3. *Let $P \vdash^{\mathsf{npf}}_{\alpha} t \sim t'$ and $N = |\mathcal{B}(t, t')|$. For $l \in \mathsf{label}(P)$ and for all proof index h:*

$$\forall u \in \left(\leq^{h,l}_{\mathsf{bt}} (t, P) \cup \mathsf{cs\text{-}path}^{h,l}(t, P)\right), |u| \leq 2^{Q(N) \,.\, 2^{4.N}}$$

Proof. Direct consequence of Proposition 5.30 and Proposition 5.33. ■

To conclude, we only need to bound the number of nested $\mathsf{CS}_\Box$ conditionals.

Proposition 5.34. *Let $P \vdash^{\mathsf{npf}}_{\alpha} t \sim t'$ and $(\mathsf{h}_i)_{1\leq i\leq n}$ be a sequence of indices of P such that for every $1 \leq i < n$, $\mathsf{h}_{i+1} \in \mathsf{cs\text{-}pos}_P(\mathsf{h}_i)$ and $\mathsf{h}_1 = \epsilon$. Then $n \leq |\mathcal{B}(t, t')| + 1$. Moreover $|\mathsf{label}(P)| \leq 2^{|\mathcal{B}(t,t')|}$.*

Proof. Let $l \in \mathsf{label}(P)$ be such that $\mathsf{h}_n \in \mathsf{h\text{-}branch}(l)$. The proof consists in building an increasing sequence of $\mathcal{S}^P_l$-normalized basic terms $\beta_1 <_{\mathsf{st}} \cdots <_{\mathsf{st}} \beta_m$ from $(\mathsf{h}_i)_{1\leq i\leq n}$ of length $m \geq n$. We then concludes using Lemma 5.21.

If $\mathsf{h}_n \neq \epsilon$, then h_n is of the form $h^n_{\mathsf{x}_n}$. We know that $\mathsf{extract}_{\mathsf{x}_n}(h^n, P)$ is a proof of $b^n \sim b'^n$ in $\mathcal{A}_{\mathsf{CS}_\Box}$. Moreover $b^n \downarrow_R$ is in $\mathsf{cs\text{-}path}^{\mathsf{h}_{n-1},l}(t, P)$ and is (t, P)-α-bounded. Be definition of (t, P)-α-bounded terms, we know that there exists $(\beta_{n,j})_{1\leq j\leq k_n}$ (with $k_n \geq 1$) such that:

- for all $1 \leq j \leq k_n$, $\beta_{n,j} \leq^{\mathsf{h}_{n-1},l}_{\mathsf{bt}} (t, P)$.
- $b^n \downarrow_R \in \mathsf{leave\text{-}st}(\beta_{n,1} \downarrow_R)$.

- $\beta_{n,k_n} \leq_{\mathsf{l}}^{\mathsf{h}_{n-1},l} (t,P)$.
- for all $1 \leq j < k_n$, $\beta_{n,j}$ is a guard of a decryption in $\beta_{n,j+1}$, and therefore $\beta_{n,j} <_{\mathsf{st}} \beta_{n,j+1}$.

If $\mathsf{h}_{n-1} \neq \epsilon$, then since $\beta_{n,k_n} \leq_{\mathsf{l}}^{\mathsf{h}_{n-1},l} (t,P)$ is (t,P)-α-bounded, and since for any $\beta \leq_{\mathsf{bt}}^{\mathsf{h}_{n-1},l} (t,P)$, $\beta_{n,j}$ is not a guard of β, we know that we are in the inductive case with different labels of the definition of (t,P)-α-bounded terms. Therefore there exists $b^{n-1} \in \mathsf{cs\text{-}path}^{\mathsf{h}_{n-2},l}(t,P)$ such that $b^{n-1} \in \mathsf{leave\text{-}st}(\beta_{n,k_n})$.

We then iterate this process until we reach ϵ, building sequences $(\beta_{i,j})_{1<i\leq n, 1\leq j\leq k_i}$ and $(b^i)_{1<i\leq n}$. Since for all i, $b^{i-1} \in \mathsf{leave\text{-}st}(\beta_{i,k_i} \downarrow_R)$ and $b^{i-1} \in \mathsf{leave\text{-}st}(\beta_{i-1,1} \downarrow_R)$ we know, using Proposition 5.11, that $\beta_{i,k_i} \equiv \beta_{i-1,1}$. Therefore we have:

$$\beta_{n,1} <_{\mathsf{st}} \cdots <_{\mathsf{st}} \beta_{n,k_n} \equiv \beta_{n-1,1} <_{\mathsf{st}} \cdots <_{\mathsf{st}} \beta_{n-1,k_{n-1}} \cdots <_{\mathsf{st}} \beta_{3,k_3} \equiv \beta_{2,1} <_{\mathsf{st}} \cdots <_{\mathsf{st}} \beta_{2,k_2}$$

Moreover, for all i we have $k_i \geq 1$, therefore we built an increasing sequence of $\mathcal{S}_l^P$-normalized basic terms of length at least $n-1$. It follows, using Lemma 5.21, that $n-1 \leq |\mathcal{B}(t,t')|$.

To upper-bound $|\mathsf{label}(P)|$, we only need to observe that we cannot have two $\mathsf{CS}_\square$ applications on the same conditional in a given branch. Consider the binary tree associated to the $\mathsf{CS}_\square$ applications in P, labelled by the corresponding $\mathsf{CS}_\square$ conditionals (say, on the left). Then this tree is of depth at most $|\mathcal{B}(t,t')|$, and therefore has at most $2^{|\mathcal{B}(t,t')|}$ leaves. ■

Theorem (Main Result)**.** *The following problem is decidable in* 3-NExpTime*:*
Input: *A ground formula $\vec{u} \sim \vec{v}$.*
Question: *Is $\mathsf{Ax} \wedge \vec{u} \not\sim \vec{v}$ unsatisfiable?*

Proof. Let $\vec{u} = u_1, \ldots, u_n$, $\vec{v} = v_1, \ldots, v_n$ and:

$$t \equiv \langle u_1 , \langle \ldots , \langle u_{n-1} , u_n \rangle \rangle \rangle \qquad\qquad t' \equiv \langle v_1 , \langle \ldots , \langle v_{n-1} , v_n \rangle \rangle \rangle$$

Using the $\mathsf{FA}_{\langle \_, \_ \rangle}$ axiom, we know that if $\vec{u} \sim \vec{v}$ is derivable then $t \sim t'$ is derivable. Conversely, we show that $t \sim t'$ is derivable then $\vec{u} \sim \vec{v}$ is derivable. For every $3 \leq i \leq n$, let $\rho_i[]$ be the i-th projection defined using π_1 and π_2 by:

$$\forall n > i \geq 1, \rho_i \equiv \pi_1(\pi_2^{i-1}([])) \qquad\qquad \rho_n[] \equiv \pi_2^{n-1}([])$$

Then:

$$\dfrac{\dfrac{t \sim t'}{(\rho_i[t])_{1\leq i\leq n} \sim (\rho_i[t'])_{1\leq i\leq n}}\ \mathsf{FA}^*_{\setminus \mathbf{0}}}{\vec{u} \sim \vec{v}}\ R$$

Hence $t \sim t'$ is derivable iff $\vec{u} \sim \vec{v}$ is derivable. Moreover, the corresponding proof of $\vec{u} \sim \vec{v}$ is of polynomial size in the size of the proof of $t \sim t'$. Therefore w.l.o.g. we can focus on the case $|\vec{u}| = |\vec{v}| = 1$.

Let $N = |\mathsf{st}(t \downarrow_R)| + |\mathsf{st}(t' \downarrow_R)|$. Using Proposition 5.34, we have bounded the number of branches of the proof tree (by $2^{N^2.2^N}$), and the number of nested $\mathsf{CS}_\square$ conditionals. For every branch, we non-deterministically guesses a set of α-bounded basic terms that can appear in a proof P of $P \vdash_\alpha^{\mathsf{npf}} t \sim t'$ using the valid candidate sequence algorithm (in polynomial time in $\mathcal{O}(N.2^{3.N}.2^{Q(N).2^{4.N}})$, using Proposition 5.32 and Proposition 5.33). Then the procedure guesses the rule applications, and checks that the candidate derivation is a valid proof. This is done in polynomial time in the size of the candidate derivation. Remark that to check whether the leaves are valid CCA_2 instances we use the polynomial-time algorithm describe in Proposition 5.3. Finally, since $|t \downarrow_R|$ is at most exponential with respect to $|t|$, this yields a 3-NExpTime decision procedure that shows the decidability of our problem. ■

5.13 Conclusion

We designed a decision procedure for a fragment of the Bana-Comon indistinguishability logic. This allows to automatically verify that a protocol satisfies some security property. Our result can be reinterpreted, in the cryptographic game transformation setting, as a cut elimination procedure that guarantees that all intermediate games introduced in a proof are of bounded size w.r.t. the protocol studied.

A lot of work remains to be done. First, our decision procedure is in 3-NExpTime, which is a high complexity. But, as we do not have any lower-bound, there may exist a more efficient decision procedure.

Finding such a lower-bound is another interesting direction of research. Then, our completeness result was proven for CCA_2 only. We believe it can be extended to more primitives and cryptographic assumptions. For example, signatures and EUF-CMA are very similar to asymmetric encryption and IND-CCA_2, and should be easy to handle (even combined with the CCA_2 axioms).

Chapter 6

Conclusion

There exist many tools for proving reachability or equivalence properties in the Dolev-Yao model, such as ProVerif, Tamarin or Deepsec. These tools are often semi of fully automatic, reasonably efficient and precise, and have been successfully used to analyse several security protocols. Unfortunately, the situation is much less satisfactory in the computational model, where most tools are interactive (e.g. EasyCrypt and F^*) or semi-automatic, such as CryptoVerif. As this model offers stronger guarantees than the Dolev-Yao model, it is crucial that progress be made there.

Therefore, our goal was to develop and study formal method techniques for proving computational indistinguishability of cryptographic protocols that are amenable to proof automation. The Bana and Comon equivalence model seemed to be a promising candidate for this. In this model, the security of a protocol is expressed as the unsatisfiability of a set of first-order logic formulas. This approach provides strong guarantees, as security in the Bana-Comon model implies security in the computational model. Moreover, this is a symbolic approach, in which the protocol execution is modeled using first-order terms. By consequence, it is potentially amenable to automated deduction techniques.

In this thesis, we showed that the Bana-Comon approach indeed fulfills our requirements, i.e. that it can be used to complete security proofs of real-world protocols (in particular of privacy properties), and that proof automation is indeed possible in this model. We did this through three different contributions.

Model and Axioms In Chapter 2, we presented the Bana-Comon model for indistinguishability, with a small extension to allow for protocols with an unbounded number of sessions. Then, we presented our first important contribution, which is the design of axioms of the logic for some frequent protocol functions (e.g. the $\oplus$), and for several cryptographic hypothesis (IND-CCA$_1$, CR-HK, EUF-CMA and PRF).

Case Studies We showed the usefulness of the Bana-Comon approach and of our axioms through several studies. In Chapter 3, we expressed a notion of privacy in the Bana-Comon logic, and proved that two simple RFID protocols, LAK$^+$ and KCL$^+$, provide privacy under the PRF assumption. Moreover, we showed that this assumption is somehow optimal, by providing attacks when the cryptographic hypothesis are weakened.

In Chapter 4, we studied the 5G-AKA authentication protocol, and showed that several unlinkability attacks against older versions of this protocol apply to it. Moreover, we found a new attack against the PRIV-AKA protocol, which is a significantly modified version of the AKA protocol claimed secure by its authors. Then, we proposed a fixed version of the 5G-AKA protocol, and proved that it provides some form of unlinkability. Again, the proof uses the Bana-Comon logic, and is for any number of agents and sessions independent from the security parameter.

Decidability Result Lastly, we argued that the Bana-Comon approach can be used to perform fully automated proofs of security protocols in Chapter 5. In this chapter, we proved the decidability of a set of axioms of the Bana-Comon logic which are computationally sound, though incomplete, under the IND-CCA$_2$ cryptographic assumption. Basically, our result can be interpreted as the decidability of a family of cryptographic game transformations. The proof relies on term rewriting and automated deduction techniques such as proof cut eliminations. This is the most theoretical result of this thesis.

6.1 Future Works

There are many interesting lines of research for future works, and a lot remains to be done.

Extending the Model We presented axioms for four cryptographic hypotheses, but there exist many more cryptographic primitives and hypothesis for which axioms remain to be designed. More interestingly, we would like to find conditions under which a security proof in the Bana-Comon model, which is for any number of sessions *independent from the security parameter*, can be lifted to a security proof for a *polynomial number* of sessions. Fixing this short-coming of the Bana-Comon method would help make this approach more attractive to cryptographers.

Scope of the Decidability Result It would be interesting to study the limits and the scope of the decidability result. First, a limitation of our decidability result is its rigidity: the result is proved for a fixed axiomatization, and in a non-modular way. We would like to design general sufficient conditions under which the satisfiability problem associated to a set of Bana-Comon axioms is decidable. This seems a good way of extending the result to more cryptographic primitives, such as signatures or hash functions.

Second, we only have an upper bound on the complexity of the satisfiability problem. We would like to improve on the current bound, which is pretty high, and to find a matching lower bound. In a similar vein, it would be interesting to prove that if we extend the set of axioms, either the satisfiability problem becomes undecidable, or it has a very high complexity.

The AKA$^+$ Protocol There are several questions related to the AKA$^+$ protocol. First, our security analysis is in a simplified two-party setting. It would be nice to prove that the protocol is secure in the more complex three-party setting, with an honest or dishonest *Serving Network*. Moreover, we would like to prove that the AKA$^+$ protocol is at a sweet spot between privacy and the amount of random number generation on the user side, as we conjectured. In other words, we want to prove an *impossibility result* stating that no protocol with as much random number generation as AKA$^+$ can provide more privacy.

Proof Automation While the security proofs of the two RFID protocols of Chapter 3 remain tractable, the proofs for the AKA$^+$ protocol in Chapter 4 are extremely tedious and lengthy. This highlights the need for mechanized proofs in the Bana-Comon indistinguishability logic, in an automated or interactive fashion. Unfortunately, the proofs of the AKA$^+$ protocol are out-of-scope of the decidability result of Chapter 5: first because the cryptographic hypothesis are different (PRF vs IND-CCA$_2$); and second, because the axiom system used to prove the AKA$^+$ protocol is much more expressive than the axiom system of the decision result. It seems unlikely that the decidability result can be extended to the full axiom system of the AKA$^+$ protocol proofs.

Of course, the obvious and usual solution is to drop either completeness or termination. In the case of the AKA$^+$ protocol, there is one interesting avenue of research, which originates in the observation that around half of the intermediate properties shown in the AKA$^+$ security proofs are correspondence properties. Basically, they are formula schemata of the following form:

$$\forall \tau,\ \psi_\tau \ \rightarrow \bigvee_{\{\tau' | \tau' \preceq \tau \wedge \theta(\tau,\tau')\}} \phi_{\tau,\tau'}$$

where τ an τ' are instants of the protocol execution; ψ_τ and $\phi_{\tau,\tau'}$ are simple formulas, typically conjunctions of literals; and $\theta(\tau,\tau')$ is an instant constraint, e.g. $\tau' = \_, \text{FN}(j) \wedge \tau' \neq \tau$. Moreover, the cryptographic axioms for unforgeability and collision-resistance are of the same form. We believe that, for this fragment of the logic, it should be possible to design a reasonably efficient proof search strategy. If successful, this would allow to mechanize half of the AKA$^+$ proofs, and would be a major first and necessary step in the direction of full automation.

Bibliography

[ABD+15] David Adrian, Karthikeyan Bhargavan, Zakir Durumeric, Pierrick Gaudry, Matthew Green, J. Alex Halderman, Nadia Heninger, Drew Springall, Emmanuel Thomé, Luke Valenta, Benjamin VanderSloot, Eric Wustrow, Santiago Zanella Béguelin, and Paul Zimmermann. Imperfect forward secrecy: How diffie-hellman fails in practice. In *ACM Conference on Computer and Communications Security*, pages 5–17. ACM, 2015.

[ABF18] Martín Abadi, Bruno Blanchet, and Cédric Fournet. The applied pi calculus: Mobile values, new names, and secure communication. *J. ACM*, 65(1):1:1–1:41, 2018.

[ACRR10] Myrto Arapinis, Tom Chothia, Eike Ritter, and Mark Ryan. Analysing unlinkability and anonymity using the applied pi calculus. In *Proceedings of the 23rd IEEE Computer Security Foundations Symposium, CSF 2010*, pages 107–121. IEEE Computer Society, 2010.

[AF04] Martín Abadi and Cédric Fournet. Private authentication. *Theor. Comput. Sci.*, 322(3):427–476, 2004.

[AFP05] Michel Abdalla, Pierre-Alain Fouque, and David Pointcheval. Password-based authenticated key exchange in the three-party setting. In *Public Key Cryptography*, volume 3386 of *Lecture Notes in Computer Science*, pages 65–84. Springer, 2005.

[AMR+12] Myrto Arapinis, Loretta Ilaria Mancini, Eike Ritter, Mark Ryan, Nico Golde, Kevin Redon, and Ravishankar Borgaonkar. New privacy issues in mobile telephony: fix and verification. In *the ACM Conference on Computer and Communications Security, CCS'12*, pages 205–216. ACM, 2012.

[AR02] Martín Abadi and Phillip Rogaway. Reconciling two views of cryptography (the computational soundness of formal encryption). *J. Cryptology*, 15(2):103–127, 2002.

[BAS12] Gergei Bana, Pedro Adão, and Hideki Sakurada. Computationally complete symbolic attacker in action. In *FSTTCS*, volume 18 of *LIPIcs*, pages 546–560. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012.

[BBD+17a] Benjamin Beurdouche, Karthikeyan Bhargavan, Antoine Delignat-Lavaud, Cédric Fournet, Markulf Kohlweiss, Alfredo Pironti, Pierre-Yves Strub, and Jean Karim Zinzindohoue. A messy state of the union: taming the composite state machines of TLS. *Commun. ACM*, 60(2):99–107, 2017.

[BBD+17b] Karthikeyan Bhargavan, Barry Bond, Antoine Delignat-Lavaud, Cédric Fournet, Chris Hawblitzel, Catalin Hritcu, Samin Ishtiaq, Markulf Kohlweiss, Rustan Leino, Jay R. Lorch, Kenji Maillard, Jianyang Pan, Bryan Parno, Jonathan Protzenko, Tahina Ramananandro, Ashay Rane, Aseem Rastogi, Nikhil Swamy, Laure Thompson, Peng Wang, Santiago Zanella Béguelin, and Jean Karim Zinzindohoue. Everest: Towards a verified, drop-in replacement of HTTPS. In *SNAPL*, volume 71 of *LIPIcs*, pages 1:1–1:12. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[BBM00] Mihir Bellare, Alexandra Boldyreva, and Silvio Micali. Public-key encryption in a multi-user setting: Security proofs and improvements. In *EUROCRYPT*, volume 1807 of *LNCS*, pages 259–274. Springer, 2000.

[BC12] G. Bana and H. Comon-Lundh. Towards unconditional soundness: Computationally complete symbolic attacker. In *Principles of Security and Trust, 2012*, volume 7215 of *LNCS*, pages 189–208. Springer, 2012.

[BC16] Gergei Bana and Rohit Chadha. Verification methods for the computationally complete symbolic attacker based on indistinguishability. *IACR Cryptology ePrint Archive*, 2016:69, 2016.

[BCC+15] Julien Bertrane, Patrick Cousot, Radhia Cousot, Jérôme Feret, Laurent Mauborgne, Antoine Miné, and Xavier Rival. Static analysis and verification of aerospace software by abstract interpretation. *Foundations and Trends in Programming Languages*, 2(2-3):71–190, 2015.

[BCE18] Gergei Bana, Rohit Chadha, and Ajay Kumar Eeralla. Formal analysis of vote privacy using computationally complete symbolic attacker. In *ESORICS (2)*, volume 11099 of *LNCS*, pages 350–372. Springer, 2018.

[BCG+13] Gilles Barthe, Juan Manuel Crespo, Benjamin Grégoire, César Kunz, Yassine Lakhnech, Benedikt Schmidt, and Santiago Zanella Béguelin. Fully automated analysis of padding-based encryption in the computational model. In *ACM Conference on Computer and Communications Security*, pages 1247–1260. ACM, 2013.

[BCK09] Mathieu Baudet, Véronique Cortier, and Steve Kremer. Computationally sound implementations of equational theories against passive adversaries. *Inf. Comput.*, 207(4):496–520, 2009.

[BCL14] G. Bana and H. Comon-Lundh. A computationally complete symbolic attacker for equivalence properties. In *2014 ACM Conference on Computer and Communications Security, CCS '14*, pages 609–620. ACM, 2014.

[BDF+14] Karthikeyan Bhargavan, Antoine Delignat-Lavaud, Cédric Fournet, Alfredo Pironti, and Pierre-Yves Strub. Triple handshakes and cookie cutters: Breaking and fixing authentication over TLS. In *IEEE Symposium on Security and Privacy*, pages 98–113. IEEE Computer Society, 2014.

[BDH+18] David A. Basin, Jannik Dreier, Lucca Hirschi, Savsa Radomirovi'c, Ralf Sasse, and Vincent Stettler. A formal analysis of 5G authentication. In *the ACM Conference on Computer and Communications Security, CCS'18*. ACM, 2018.

[BDK+10] Gilles Barthe, Marion Daubignard, Bruce M. Kapron, Yassine Lakhnech, and Vincent Laporte. On the equality of probabilistic terms. In Edmund M. Clarke and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning - 16th International Conference, LPAR-16, Dakar, Senegal, April 25-May 1, 2010, Revised Selected Papers*, volume 6355 of *LNCS*, pages 46–63. Springer, 2010.

[BDPA14] Guido Bertoni, Joan Daemen, Michaël Peeters, and Gilles Van Assche. The making of KECCAK. *Cryptologia*, 38(1):26–60, 2014.

[BDPR98] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In *CRYPTO*, volume 1462 of *LNCS*, pages 26–45. Springer, 1998.

[Ber08] Daniel J. Bernstein. The Salsa20 family of stream ciphers. In *The eSTREAM Finalists*, volume 4986 of *Lecture Notes in Computer Science*, pages 84–97. Springer, 2008.

[BGHB11] G. Barthe, B. Grégoire, S. Heraud, and S. Zanella Béguelin. Computer-aided security proofs for the working cryptographer. In *Advances in Cryptology - CRYPTO, 2011*, volume 6841 of *LNCS*, pages 71–90. Springer, 2011.

[BHO13] Gergei Bana, Koji Hasebe, and Mitsuhiro Okada. Computationally complete symbolic attacker and key exchange. In *ACM Conference on Computer and Communications Security*, pages 1231–1246. ACM, 2013.

[BHP+17] Ravishankar Borgaonkar, Lucca Hirshi, Shinjo Park, Altaf Shaik, Andrew Martin, and Jean-Pierre Seifert. New adventures in spying 3G & 4G users: Locate, track, monitor, 2017. Briefing at BlackHat USA 2017.

[BKR00] Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of the cipher block chaining message authentication code. *J. Comput. Syst. Sci.*, 61(3):362–399, 2000.

[Bla] Bruno Blanchet. ProVerif*: Cryptographic protocols verifier in the formal model.* available at http://proseccco.gforge..inria.fr/personal/bblanchet/proverif/.

[Bla04] Bruno Blanchet. Automatic proof of strong secrecy for security protocols. In *IEEE Symposium on Security and Privacy*, page 86. IEEE Computer Society, 2004.

[Bla08] Bruno Blanchet. A computationally sound mechanized prover for security protocols. *IEEE Trans. Dependable Sec. Comput.*, 5(4):193–207, 2008.

[BMR14] Michael Backes, Esfandiar Mohammadi, and Tim Ruffing. Computational soundness results for proverif - bridging the gap from trace properties to uniformity. In *POST*, volume 8414 of *Lecture Notes in Computer Science*, pages 42–62. Springer, 2014.

[BMU12] Michael Backes, Ankit Malik, and Dominique Unruh. Computational soundness without protocol restrictions. In *ACM Conference on Computer and Communications Security*, pages 699–711. ACM, 2012.

[BP05] Michael Backes and Birgit Pfitzmann. Limits of the cryptographic realization of dolev-yao-style XOR. In *ESORICS*, volume 3679 of *Lecture Notes in Computer Science*, pages 178–196. Springer, 2005.

[BPW06] Michael Backes, Birgit Pfitzmann, and Michael Waidner. Limits of the BRSIM/UC soundness of dolev-yao models with hashes. In *ESORICS*, volume 4189 of *Lecture Notes in Computer Science*, pages 404–423. Springer, 2006.

[BR06] Mihir Bellare and Phillip Rogaway. The security of triple encryption and a framework for code-based game-playing proofs. In *EUROCRYPT*, volume 4004 of *LNCS*, pages 409–426. Springer, 2006.

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *FOCS*, pages 136–145. IEEE Computer Society, 2001.

[CB13] Vincent Cheval and Bruno Blanchet. Proving more observational equivalences with proverif. In *POST*, volume 7796 of *Lecture Notes in Computer Science*, pages 226–246. Springer, 2013.

[CC77] Patrick Cousot and Radhia Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *POPL*, pages 238–252. ACM, 1977.

[CCD15] Rémy Chrétien, Véronique Cortier, and Stéphanie Delaune. Decidability of trace equivalence for protocols with nonces. In *CSF*, pages 170–184. IEEE Computer Society, 2015.

[CCD17] V. Cheval, H. Comon-Lundh, and S. Delaune. A procedure for deciding symbolic equivalence between sets of constraint systems. *Inf. Comput.*, 255:94–125, 2017.

[CCS13] Hubert Comon-Lundh, Véronique Cortier, and Guillaume Scerri. Tractable inference systems: An extension with a deducibility predicate. In *CADE*, volume 7898 of *LNCS*, pages 91–108. Springer, 2013.

[CCZ10] Hubert Comon-Lundh, Véronique Cortier, and Eugen Zalinescu. Deciding security properties for cryptographic protocols. application to key cycles. *ACM Trans. Comput. Log.*, 11(2):9:1–9:42, 2010.

[Cha82] David Chaum. Blind signatures for untraceable payments. In *CRYPTO*, pages 199–203. Plenum Press, New York, 1982.

[Chi07] Hung-Yu Chien. SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity. *IEEE Trans. Dependable Secur. Comput.*, 4(4):337–340, October 2007.

[CK17] H. Comon and A. Koutsos. Formal computational unlinkability proofs of RFID protocols. In *30th Computer Security Foundations Symposium, 2017*, pages 100–114. IEEE Computer Society, 2017.

[CKKW06] Véronique Cortier, Steve Kremer, Ralf Küsters, and Bogdan Warinschi. Computationally sound symbolic secrecy in the presence of hash functions. In *FSTTCS*, volume 4337 of *Lecture Notes in Computer Science*, pages 176–187. Springer, 2006.

[CKR18] Vincent Cheval, Steve Kremer, and Itsaka Rakotonirina. DEEPSEC: deciding equivalence properties in security protocols theory and practice. In *2018 IEEE Symposium on Security and Privacy, SP 2018*, pages 529–546. IEEE, 2018.

[CKW11] Véronique Cortier, Steve Kremer, and Bogdan Warinschi. A survey of symbolic methods in computational analysis of cryptographic systems. *J. Autom. Reasoning*, 46(3-4):225–259, 2011.

[CL73] Chin-Liang Chang and Richard C. T. Lee. *Symbolic logic and mechanical theorem proving*. Computer science classics. Academic Press, 1973.

[CW11] Véronique Cortier and Bogdan Warinschi. A composable computational soundness notion. In Yan Chen, George Danezis, and Vitaly Shmatikov, editors, *Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS 2011, Chicago, Illinois, USA, October 17-21, 2011*, pages 63–74. ACM, 2011.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Trans. Information Theory*, 22(6):644–654, 1976.

[DJ90] Nachum Dershowitz and Jean-Pierre Jouannaud. Rewrite systems. In *Handbook of Theoretical Computer Science, Volume B: Formal Models and Sematics (B)*, pages 243–320. Elsevier and MIT Press, 1990.

[DOT17] Emanuele D'Osualdo, Luke Ong, and Alwen Tiu. Deciding secrecy of security protocols for an unbounded number of sessions: The case of depth-bounded processes. In *CSF*, pages 464–480. IEEE Computer Society, 2017.

[DR02] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Information Security and Cryptography. Springer, 2002.

[DY83] Danny Dolev and Andrew Chi-Chih Yao. On the security of public key protocols. *IEEE Trans. Information Theory*, 29(2):198–207, 1983.

[FOO92] Atsushi Fujioka, Tatsuaki Okamoto, and Kazuo Ohta. A practical secret voting scheme for large scale elections. In *AUSCRYPT*, volume 718 of *Lecture Notes in Computer Science*, pages 244–251. Springer, 1992.

[FOR16] Pierre-Alain Fouque, Cristina Onete, and Benjamin Richard. Achieving better privacy for the 3gpp AKA protocol. *PoPETs*, 2016(4):255–275, 2016.

[FS01] Alain Finkel and Philippe Schnoebelen. Well-structured transition systems everywhere! *Theor. Comput. Sci.*, 256(1-2):63–92, 2001.

[GB01] Shafi Goldwasser and Mihir Bellare. Lecture notes on cryptography, 2001.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807, 1986.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[Gol01] Oded Goldreich. *The Foundations of Cryptography - Volume 1, Basic Techniques*. Cambridge University Press, 2001.

[HBD16] Lucca Hirschi, David Baelde, and Stéphanie Delaune. A method for verifying privacy-type properties: The unbounded case. In *IEEE Symposium on Security and Privacy, SP 2016*, pages 564–581. IEEE Computer Society, 2016.

[HPVP11] Jens Hermans, Andreas Pashalidis, Frederik Vercauteren, and Bart Preneel. A new RFID privacy model. In *ESORICS*, volume 6879 of *Lecture Notes in Computer Science*, pages 568–587. Springer, 2011.

[Hüt02] Hans Hüttel. Deciding framed bisimilarity. *Electr. Notes Theor. Comput. Sci.*, 68(6):1–18, 2002.

[JLM05] Romain Janvier, Yassine Lakhnech, and Laurent Mazaré. Completing the picture: Soundness of formal encryption in the presence of active adversaries. In *ESOP*, volume 3444 of *Lecture Notes in Computer Science*, pages 172–185. Springer, 2005.

[JR12] Charanjit S. Jutla and Arnab Roy. Decision procedures for simulatability. In *ESORICS*, volume 7459 of *LNCS*, pages 573–590. Springer, 2012.

[JW09] Ari Juels and Stephen A. Weis. Defining strong privacy for RFID. *ACM Trans. Inf. Syst. Secur.*, 13(1):7:1–7:23, November 2009.

[KCL07] I. J. Kim, E. Y. Choi, and D. H. Lee. Secure mobile RFID system against privacy and security problems. In *Security, Privacy and Trust in Pervasive and Ubiquitous Computing, 2007. SECPerU 2007. Third International Workshop on*, pages 67–72, July 2007.

[Koc96] Paul C. Kocher. Timing attacks on implementations of diffie-hellman, rsa, dss, and other systems. In *CRYPTO*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996.

[LAK06] Sangshin Lee, Tomoyuki Asano, and Kwangjo Kim. RFID mutual authentication scheme based on synchronized secret information. In *Symposium on cryptography and information security*, 2006.

[LBdM07] Tri Van Le, Mike Burmester, and Breno de Medeiros. Universally composable and forward-secure RFID authentication and authenticated key exchange. In Feng Bao and Steven Miller, editors, *Proceedings of the 2007 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2007, Singapore, March 20-22, 2007*, pages 242–252. ACM, 2007.

[Low95] Gavin Lowe. An attack on the Needham-Schroeder public-key authentication protocol. *Inf. Process. Lett.*, 56(3):131–133, 1995.

[Low97] Gavin Lowe. A hierarchy of authentication specification. In *CSFW*, pages 31–44. IEEE Computer Society, 1997.

[LSWW14] Ming-Feng Lee, Nigel P. Smart, Bogdan Warinschi, and Gaven J. Watson. Anonymity guarantees of the UMTS/LTE authentication and connection protocol. *Int. J. Inf. Sec.*, 13(6):513–527, 2014.

[MSCB13] S. Meier, B. Schmidt, C. Cremers, and D. Basin. The tamarin prover for the symbolic analysis of security protocols. In *25th International Conference on Computer Aided Verification, CAV'13*, pages 696–701. Springer-Verlag, 2013.

[MW04] Daniele Micciancio and Bogdan Warinschi. Soundness of formal encryption in the presence of active adversaries. In *TCC*, volume 2951 of *Lecture Notes in Computer Science*, pages 133–151. Springer, 2004.

[NL15] Y. Nir and A. Langley. ChaCha20 and Poly1305 for IETF protocols. RFC 7539, RFC Editor, May 2015. http://www.rfc-editor.org/rfc/rfc7539.txt.

[NS78] Roger M. Needham and Michael D. Schroeder. Using encryption for authentication in large networks of computers. *Commun. ACM*, 21(12):993–999, 1978.

[OP08] Khaled Ouafi and Raphael C.-W. Phan. Privacy of recent RFID authentication protocols. In Liqun Chen, Yi Mu, and Willy Susilo, editors, *Information Security Practice and Experience, 4th International Conference, ISPEC 2008, Sydney, Australia, April 21-23, 2008, Proceedings*, volume 4991 of *Lecture Notes in Computer Science*, pages 263–277. Springer, 2008.

[PCER08] Pedro Peris-Lopez, Julio César Hernández Castro, Juan M. Estévez-Tapiador, and Arturo Ribagorda. Advances in ultralightweight cryptography for low-cost RFID tags: Gossamer protocol. In Kyo-Il Chung, Kiwook Sohn, and Moti Yung, editors, *Information Security Applications, 9th International Workshop, WISA 2008, Jeju Island, Korea, September 23-25, 2008, Revised Selected Papers*, volume 5379 of *Lecture Notes in Computer Science*, pages 56–68. Springer, 2008.

[Sat89] Mahadev Satyanarayanan. Integrating security in a large distributed system. *ACM Trans. Comput. Syst.*, 7(3):247–280, 1989.

[Sce15] Guillaume Scerri. *Proofs of security protocols revisited.* PhD thesis, École Normale Supérieure de Cachan, 2015.

[Sho04] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. *IACR Cryptology ePrint Archive*, 2004:332, 2004. https://eprint.iacr.org/2004/332.

[SS16] Guillaume Scerri and Ryan Stanley-Oakes. Analysis of key wrapping apis: Generic policies, computational security. In *IEEE 29th Computer Security Foundations Symposium, CSF 2016, Lisbon, Portugal, June 27 - July 1, 2016*, pages 281–295. IEEE Computer Society, 2016.

[SSB+16] Altaf Shaik, Jean-Pierre Seifert, Ravishankar Borgaonkar, N. Asokan, and Valtteri Niemi. Practical attacks against privacy and availability in 4g/lte mobile communication systems. In *23rd Annual Network and Distributed System Security Symposium, NDSS*. The Internet Society, 2016.

[Str07] Daehyun Strobel. IMSI catcher. *Ruhr-Universität Bochum, Seminar Work*, 2007.

[TS318] TS 33.501: Security architecture and procedures for 5G system, September 2018.

[Unr10] Dominique Unruh. The impossibility of computationally sound XOR. *IACR Cryptology ePrint Archive*, 2010:389, 2010.

[Vau07] Serge Vaudenay. On privacy models for RFID. In *ASIACRYPT 2007, 13th International Conference on the Theory and Application of Cryptology and Information Security*, LNCS, pages 68–87. Springer, 2007.

[vdBVdR15] Fabian van den Broek, Roel Verdult, and Joeri de Ruiter. Defeating IMSI catchers. In *ACM Conference on Computer and Communications Security, CCS'15*, pages 340–351. ACM, 2015.

[VDR08] Ton Van Deursen and Sasa Radomirovic. Attacks on RFID protocols. *IACR Cryptology ePrint Archive*, 2008:310, 2008.

[WL93] T. Y. C. Woo and S. S. Lam. A semantic model for authentication protocols. In *Proceedings 1993 IEEE Computer Society Symposium on Research in Security and Privacy*, pages 178–194, May 1993.

General Index

Symbols Index

Symbols

A

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Preuves symboliques de propriétés d'indistinguabilité calculatoire

Mots clés : Protocoles de sécurité, Sécurité calculatoire, Preuves automatiques, Indistinguabilité

Résumé : Notre société utilise de nombreux systèmes de communications. Parce que ces systèmes sont omniprésents et sont utilisés pour échanger des informations sensibles, ils doivent être protégés. Cela est fait à l'aide de protocoles cryptographiques. Il est crucial que ces protocoles assurent bien les propriétés de sécurité qu'ils affirment avoir, car les échecs peuvent avoir des conséquences importantes. Malheureusement, concevoir des protocoles cryptographiques est notoirement difficile, comme le montre la régularité avec laquelle de nouvelles attaques sont découvertes. Nous pensons que la vérification formelle est le meilleur moyen d'avoir de bonnes garanties dans la sécurité d'un protocole: il s'agit de prouver mathématiquement qu'un protocole satisfait une certaine propriété de sécurité.
Notre objectif est de développer les techniques permettant de vérifier formellement des propriétés d'équivalence sur des protocoles cryptographiques, en utilisant une méthode qui fournit de fortes garanties de sécurités, tout en étant adaptée à des procédures de preuve automatique. Dans cette thèse, nous défendons l'idée que le modèle Bana-Comon pour les propriétés d'équivalences satisfait ces objectifs. Nous soutenons cette affirmation à l'aide de trois contributions.
Tout d'abord, nous étayons le modèle Bana-Comon en concevant des axiomes pour les fonctions usuelles des protocoles de sécurités, et pour plusieurs hypothèses cryptographiques. Dans un second temps, nous illustrons l'utilité de ces axiomes et du modèle en réalisant des études de cas de protocoles concrets: nous étudions deux protocoles RFID, KCL et LAK, ainsi que le protocole d'authentification 5G-AKA, qui est utilisé dans les réseaux de téléphonie mobile. Pour chacun de ces protocoles, nous montrons des attaques existentes ou nouvelles, proposons des versions corrigées de ces protocoles, et prouvons que celles-ci sont sécurisées. Finalement, nous étudions le problème de l'automatisation de la recherche de preuves dans le modèle Bana-Comon. Pour cela, nous prouvons la décidabilité d'un ensemble de règles d'inférences qui est une axiomatisation correcte, bien que incomplète, de l'indistingabilité calculatoire, lorsque l'on utilise un schéma de chiffrement IND-CCA$_2$. Du point de vue d'un cryptographe, cela peut être interprété comme la décidabilité d'un ensemble de transformations de jeux.

Title : Symbolic Proofs of Computational Indistinguishability

Keywords : Security protocols, Computational security, Automatic proofs, Indistinguishability

Abstract : Our society extensively relies on communications systems. Because such systems are used to exchange sensitive information and are pervasive, they need to be secured. Cryptographic protocols are what allow us to have secure communications. It is crucial that such protocols do not fail in providing the security properties they claim, as failures have dire consequences. Unfortunately, designing cryptographic protocols is notoriously hard, and major protocols are regularly and successfully attacked. We argue that formal verification is the best way to get a strong confidence in a protocol security. Basically, the goal is to mathematically prove that a protocol satisfies some security property.
Our objective is to develop techniques to formally verify equivalence properties of cryptographic protocols, using a method that provides strong security guarantees while being amenable to automated deduction techniques. In this thesis, we argue that the Bana-Comon model for equivalence properties meets these goals. We support our claim through three different contributions.
First, we design axioms for the usual functions used in security protocols, and for several cryptographic hypothesis. Second, we illustrate the usefulness of these axioms and of the model by completing case studies of concrete protocols: we study two RFID protocols, KCL and LAK, as well as the 5G-AKA authentication protocol used in mobile communication systems. For each of these protocols, we show existing or new attacks against current versions, propose fixes, and prove that the fixed versions are secure. Finally, we study the problem of proof automation in the Bana-Comon model, by showing the decidability of a set of inference rules which is a sound, though incomplete, axiomatization of computational indistinguishability when using an IND-CCA$_2$ encryption scheme. From a cryptographer's point of view, this can be seen as the decidability of a fixed set of cryptographic game transformations.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France