Relay Attacks over the Internet : Anomaly Detection using Time Measurement

Olivier Gimenez

COLLEGE MATHS, TELECOMS
DOCTORAL INFORMATIQUE, SIGNAL
BRETAGNE SYSTEMES, ELECTRONIQUE

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES

THÈSE DE DOCTORAT DE

L'INSTITUT NATIONAL DES SCIENCES APPLIQUÉES DE RENNES

ÉCOLE DOCTORALE N° 601
Mathématiques, Télécommunications, Informatique, Signal, Systèmes, Électronique
Spécialité : *Informatique*

Par

Olivier GIMENEZ

« Relay Attacks over the Internet: Anomaly Detection using Time Measurement »

Thèse présentée et soutenue à IRISA, le 28/02/2023
Unité de recherche : IRISA, Orange Innovation
Thèse N° : 23ISAR 07 / D23 - 07

Rapporteurs avant soutenance :

Pascal LAFOURCADE Professeur, Université Clermont-Auvergne, Clermont-Ferrand, France
Isabelle CHRISMENT Professeure, Université de Lorraine, Nancy, France

Composition du Jury :

Président :	Pierre-Alain FOUQUE	Professeur, Université Rennes 1, Rennes, France
Examinateurs :	Pascal LAFOURCADE	Professeur, Université Clermont-Auvergne Clermont-Ferrand, France
	Isabelle CHRISMENT	Professeure, Université de Lorraine, Nancy, France
	Ioana BOUREANU	Professor, University of Surrey, Guildford, United Kingdom
	Jean LENEUTRE	Maître de Conférences, Telecom Paris, Palaiseau, France
	Ghada ARFAOUI	Ingénieure de Recherche, Orange INNOV/IT-S, Cesson-Sévigné, France
	Jacques TRAORÉ	Ingénieur de Recherche, Orange INNOV/IT-S, Caen, France
Dir. de thèse :	Gildas AVOINE	Professeur, INSA Rennes, Rennes, France

Remerciements

Je tiens en tout premier lieu à exprimer ma gratitude la plus profonde envers mon directeur de thèse et mes encadrants :
À Gildas Avoine, merci pour la qualité de nos discussions, la sagesse de tes conseils, la rigueur de ta supervision, et de façon générale, l'investissement que tu accordes aux travaux de tes doctorants.
À Jacques Traoré, merci de m'avoir aidé à conserver une partie de mes travaux autour de la cryptographie. Nos "courtes" réunions pour parler de preuves de sécurité, dérivant sans cesse sur des sujets tout aussi nombreux que passionnants, font parties des moments qui resteront dans ma mémoire longtemps après ma thèse.
À Ghada Arfaoui, merci pour l'expertise que tu as su apporter au domaine de recherche que je maîtrisais le moins, j'ai appris petit à petit à découvrir les réseaux et suis fier, aujourd'hui, de constater le chemin parcouru. À vous trois, merci votre soutien continu et votre présence tout au long de mon parcours de doctorat.
Je ne saurais poursuivre sans remercier également les membres de mon jury, Ioana Boureanu, Isabelle Chrisment, Pierre-Alain Fouque, Pascal Lafourcade et Jean Leneutre, qui ont généreusement consacré de leur temps et de leur expertise pour évaluer cette thèse. J'apprécie leurs commentaires pertinents, leurs suggestions et leurs critiques constructives.
Un grand merci à mes collègues de travail :
Du côté académique, merci à Gwendal, Daniel et Pierrick, pour votre bonne humeur et votre sympathie, merci à Thomas pour les mille et une fois où je t'ai appelé à l'aide pour m'aider à surmonter mes galères administratives. Mais surtout, merci à Diane, la meilleure collègue de bureau ever ! T'es une personne en or avec qui j'ai adoré discuter, me plaindre, jouer au pendu, faire un rush commun pour respecter nos deadlines respectives. Crois en tes capacités et cesse donc de te mettre tant de pression, tu vas tout déchirer ! !
Du côté industriel, merci à Guillaume et Adel (a.k.a "les bons plans"), mes premiers collègues de bureau, sachez que l'évènement était incroyable. Merci à Maxime, mon fidèle partenaire d'échiquier en pause café, à Anaïs, avec qui je partage l'expérience d'une vie étudiante à Limoges, à Paul, qui est arrivé 3 semaines après moi pour soutenir plusieurs mois avant, et à Ferran, qui m'a fait travailler ma diction en espagnol. Merci bien évidemment à tous mes autres collègues d'Orange Innovation, Sébastien, Loïc, Bastien, Nicolas, Jeremy, j'espère vous recroiser au cours de mes pérégrinations professionnelles futures. Je n'oublierai pas de remercier mes collègues polonais d'Orange Innovation, qui se sont montré disponible et m'ont aidé dans la mise en place de mes expériences de mesures.
Merci à tous mes amis, que je connais, pour certains, depuis ma tendre enfance, Jeremy, Thomas, Hugo, Marine, Noémie, Joana, Paul, Quentin, Axel, Julien, Claire, vous êtes un phare dans mon existence, et malgré la distance qui nous sépare, chaque retrouvaille me donne l'impression de ne vous avoir jamais quitté.

Merci à Maxime, rencontré au master de Limoges, d'abord un simple camarade de classe, tu es très vite devenu un ami sur qui je peux compter, je te souhaite toute la réussite que tu mérites dans la suite de ta carrière.
Merci tout particulièrement à Cyril, le frère que j'ai choisi, avec qui je partage mes réussites, mes doutes, des crises de rires au téléphone ou devant un bon vieil épisode de la carte au trésor. La liste de nos privates jokes doublerait aisément le nombre de pages de ce manuscrit, alors, il vaut mieux ne pas trop en dire.
Bien sûr, comment ne pas parler de ma formidable famille ? Mon cher frère, mes cousins et cousines, mes neveux et nièces, mes oncles et tantes, vous êtes bien trop nombreux pour que je cite tous vos prénoms, mais soyez sûr que chaque moment passé avec vous est une bouffée d'air frais. Merci d'être ces magnifiques personnes. Merci à ma maman, qui est toujours prête à tout pour son fils, merci de traverser la France pour me regarder parler pendant 45 minutes d'un sujet qui t'est complètement opaque. Merci à mon papa qui me communique sa fierté par son simple regard. Je vous aimes !
Merci enfin à ma compagne, Jessica. Merci de me supporter au quotidien, de m'écouter me plaindre de mon syndrome de l'imposteur, de m'excuser lorsque je suis tellement en retard dans mon travail que je n'arrive plus à me concentrer sur rien d'autres. Mais merci surtout de me rendre heureux, tu as été, il y a bien longtemps, une amie, tu es devenue ma compagne, et je suis assurément fier de pouvoir dire que dans un peu plus de 5 mois, tu seras la mère de mon enfant. Je t'aime.

Mentions honorables :
Merci à toutes les personnes reliées de près ou de loin à la distribution du café sur les lieux de travail, sans vous, nous sommes tous perdus. Merci au technicien qui est venu installer la fibre à mon domicile il y a quelques mois, ça commençait à devenir compliqué. Merci à Miguel, mon aspirateur robot, change rien. Merci à Scapin, mon petit bouledogue qui m'apporte une dose nécessaire de mignonnerie quotidienne dans ce monde de brute.
Merci à tous !

Table of Contents

Résumé en Français

Le réseau Internet est structuré en une multitude de plus petits réseaux connectés entre eux et gérés par diverses entités (académiques, gouvernementales, commerciales ou autres). Ces réseaux sont appelés des *Systèmes Autonomes* (AS) et sont identifiables par leurs pairs grâce à un numéro unique appelé numéro d'AS (ASN). Un AS détient des ensembles d'adresses IP. Chaque AS supervise ses propres communications internes et possède une ou plusieurs portes de sortie vers des AS adjacents. Les communications internes, dites intra-AS, sont entièrement gérés par l'autorité en charge de l'AS, alors que les communications externes, dites extra-AS, sont principalement choisies selon des critères d'accords économiques ou d'optimisation du temps de réception. Afin de pouvoir précisément acheminer chacun des paquets vers le bon nœud destinataire, deux protocoles sont utilisés : le protocole *Internet Protocol* (IP) pour le plan de données, et le protocole *Border Gateway Protocol* (BGP) pour le plan de contrôle.

La mission de BGP est de s'assurer que chaque routeur sache de quelle façon faire suivre les paquets qu'ils reçoivent. Succinctement, BGP permet aux Systèmes Autonomes de construire des tables de routage en envoyant des messages contenant les adresses IP qu'ils détiennent, et en faisant suivre ces messages de proche en proche, accompagné de la liste ordonnée de tous les AS déjà traversés à ce stade. Ces messages sont appelés des annoncements, l'AS envoyant un annoncement est appelé l'origine de l'annoncement et la liste ordonnée des AS traversé est appelé le chemin d'AS. À la réception d'un annoncement contenant un ensemble d'adresse IP, l'AS d'origine et le chemin d'AS courant, un AS peut mettre à jour sa table de routage en gardant en mémoire les adresses IP ainsi que le plus proche AS auquel faire suivre des messages à destination de ces IP.

Le problème de cette procédure vient du fait qu'elle repose sur une parfaite confiance quant à la légitimité des informations reçues. Autrement dit, aucune mesure de sécurité n'est prise pour empêcher un AS d'injecter de fausses informations dans un annoncement. Ce genre d'annoncements fallacieux peut engendrer des modifications dans les tables de routage. Lorsqu'elle est réalisée de façon malveillante, une action pouvant causer des changements dans le trajet des données est appelée une *attaque hijacking* ou *attaque de détournement*. Une fausse annonce BGP peut générer divers types de menaces, par exemple :

— Un AS annonçant un ensemble d'IP qu'il ne détient pas vraiment perturberait les choix de routes des autres AS, rendant injoignable le véritable détenteur des IP cibles pour une grande partie du réseau Internet. De plus, le trafic concerné attendrait le mauvais AS, permettant un usage malveillant des données obtenues.

— Un AS faisant suivre un annoncement en y ayant injecté un chemin d'AS modifié peut devenir un relai sur la route entre différents couples d'AS. Ceci permettrait de mener des attaques de l'homme-du-milieu, ou de passivement observer des échanges

sensibles sur de longues périodes.

État de l'Art

Depuis juillet 1994, quand BGP-4 a été présenté [63], de nombreuses recherches ont tenté de contrecarrer les attaques de détournement :

Propositions pour renforcer BGP. Une part importante de ces travaux repose sur l'utilisation de la cryptographie asymétrique pour sécuriser BGP grâce aux signatures numériques permettant d'attester les AS d'origines ainsi que les chemins d'AS pour chaque annoncement. Kent, Lynn, et Seo proposèrent « Secure BGP » (S-BGP) en 2000 [43]. En 2003, White proposa « Secure Origin BGP » (soBGP) [74]. Puis en 2005, Wan, Kranakis et Van Oorschot, suivi une année plus tard, en 2006, par Karlin, Forrest et Rexford, proposèrent respectivement « Pretty Secure BGP » (psBGP) [73] et « Pretty Good BGP » (pgBGP) [42].
En 2005, l'*Internet Engineering Task Force* (IETF) mit en place le groupe de travail *Secure Inter Domain Routing* (SIDR). Depuis sa création, le groupe SIDR a travaillé à la standardisation du protocole BGPSEC [46], basé sur S-BGP [43]. BGPSEC tire avantage d'une infrastructure de clé publique pour certifier les clés des différents AS et leur permettre de générer des signatures numériques. Le principe de base est de demander à chaque AS recevant un annoncement BGP de signer son chemin d'AS courant, ainsi que l'ASN du prochain AS recevant l'annoncement. Bien que BGPSEC soit probablement la proposition recevant le plus d'investissement de la part de la communauté scientifique, elle reçoit également des critiques quant aux problèmes qu'elle peut engendrer, en particulier concernant le grand nombre de calculs nécessaire aux AS pour générer des signatures pour chaque annoncement reçu. Cela pourrait impliquer un besoin de mise à jour du matériel physique.
Dans le but de minimiser le besoin calculatoire induit par les signatures numériques, Hu, Perrig, Johnson et Sirbu publièrent un article [39] en 2003. Cet article propose d'utiliser des Codes d'Authentification de Message (MAC) en remplacement des signatures, et un autre [40] en 2004, présentant le protocole Secure Path Vector (SPV) qui utilise des signatures à usage unique (one-time signatures) permettant de tirer parti d'une phase de pré-calcul hors ligne pour accélérer le processus. Cependant, Raghavan, Panjwani et Mityagin ont montré en 2007 que SPV n'empêche pas un AS de faire suivre un chemin d'AS modifié.
Gersh et Massey [26] publièrent des travaux en 2013 dans lesquels ils s'attaquèrent au problème de la validation de l'AS d'origine en utilisant des serveurs DNS. Malheureusement, ces travaux n'adressent que la validation d'origine et non l'altération des chemins d'AS. En 2003, Goodell et al. présentèrent le protocole Interdomain Route Validation (IRV) [29], qui fut amélioré en 2015 par Chen et Haeberlen [19]. Cette solution propose l'ajout à chaque AS d'un serveur dédié appelé serveur IRV. Les serveurs IRV gardent en mémoire et gèrent les politiques des AS et les adresses IP qu'ils détiennent. À la récep-

tion, la légitimé d'un annoncement peut être vérifiée en contactant le serveur IRV des AS correspondants. Cependant, cette approche est difficile à mettre en pratique à l'échelle de l'Internet.

Propositions de nouvelles architectures de routage. Face à l'actuel manque de solutions satisfaisantes, un nouveau domaine de recherche nommé le Path Aware Networking (PAN) émerge. Les réseaux Path-Aware visent à redéfinir les bases de l'actuelle architecture de routage de sorte que cette nouvelle architecture ne souffre pas des lacunes du protocole BGP. En particulier, une proposition appelé SCION semble sortir du lot avec un fort investissement de recherche [54, 20]. L'idée de SCION est de ne plus regrouper des sous réseaux de nœuds en Systèmes Autonomes, mais en tant que de nouvelles entités appelées des Domaines de Confiance (Trust Domains, TD). Un Domaine de Confiance est administré de façon similaire à un Système Autonome, mais possède une structure englobant un ou plusieurs sous-domaines, ou sub-TD. Comme son nom le suggère, un Domaine de Confiance ne contient que des entités partageant une confiance mutuelle. Cette structure permet de construire d'elle-même des ensembles ou les Domaine de Confiance de plus haut niveau représenteront des groupes clairement identifiables comme des alliances gouvernementales ou des partenariats entre plusieurs grandes entreprises. Il est cependant très complexe de transiter d'une architecture connue et maîtrisée à une proposition totalement nouvelle, notamment pour des raisons de confiance collectives en cette nouvelle solution. L'adoption d'une redéfinition complète de l'architecture de routage n'est pas à envisager dans un futur proche.

Propositions de détection d'anomalies. Si les recherches précédemment citées semblent insuffisantes pour empêcher efficacement des attaques de détournements sur BGP, beaucoup d'autres propositions ont été faites pour fournir un outil de détection plutôt que de contre-mesures. Certains de ces travaux s'orientent vers la supervision d'annoncement suspicieux. Par exemple, lorsque des adresses IP identiques sont déclarées par plus d'un seul AS [59, 45, 58, 69]. D'autres travaux proposent des solutions orientées vers le machine learning [3, 31].

Une approche intéressante a également été abordée par Hiran, Carlson et Shahmehri en 2015 [36]. Les auteurs y présentent un procédé nommé CrowdSec, dans lequel les utilisateurs mesurent passivement les temps d'aller-retour entre leur machine et des adresses IP déclarées par d'autres AS. En supposant les premières mesures rassemblées dans un contexte non-altéré, les utilisateurs peuvent indépendamment utiliser le test statistique de Grubb [30] pour identifier des mesures de temps anormalement éloignées des valeurs attendues. Cependant, ce procédé ne permet pas de détecter un attaquant qui serait capable d'intercepter des messages tout en répondant une confirmation dans un laps de temps satisfaisant le test statistique.

Contributions

Cette thèse est dédiée à la conception d'une nouvelle technique de détection d'attaques par relai. Détecter des attaques par détournement est immédiat dès lors que le message d'origine n'atteint pas son destinataire initial. Pour cette raison, les travaux décrits dans ce manuscrit adressent les attaques par relai discrètes. Nous pensons que ces attaques peuvent causer de sévères conséquences en termes de violation à la vie privée et de répercussions géo-politiques. Nous définissons une « attaque par relai sur le long-terme » par l'altération d'une route authentique séparant un couple de nœuds fixes, de telle façon qu'un nœud attaquant peut écouter tous les messages des nœuds communicants, à leur insu, et sans en modifier le contenu. Dans une telle situation, l'attaquant joue un rôle passif, et cherche à rassembler autant d'informations que possible, aussi longtemps que possible. Ces attaques peuvent ultimement être utilisées à une échelle internationale pour obtenir d'importantes méta-données, menant à d'éventuels chantages sous la menace de dévoiler des informations privées discréditant un gouvernement ou une industrie. Ces attaques peuvent aussi être utilisées dans le but d'endommager certaines structures économiques. Par exemple, en 2017, Apostolaki, Zohar et Vanbever ont montré que relayer des messages contribuant à la blockchain du bitcoin pour retarder la propagation des blocs pouvait engendrer des pertes financières très importantes [5].
Nous présentons le protocole Internet-friendly Cryptographic Relay-detection Protocol (ICRP). ICRP est un protocole bi-partie détectant des attaques par relai sur des grandes distances en tirant avantage des mesures de temps d'aller-retour d'une façon analogue à [36]. Cependant, ICRP joue également le rôle d'un schéma d'authentification interactif, empêchant un attaquant de tromper les mesures de temps par des réponses anticipées. Cet ajout cryptographique est directement inspiré des protocoles de vérification de proximité (distance-bounding protocols), qui sont eux-mêmes une technique de détection d'attaques par relai, appliqué au contexte des communications sans contact.
Un protocole de vérification de proximité est un protocole d'authentification interactif, assurant non seulement que l'appareil communicant est authentique, mais aussi que cet appareil se trouve physiquement plus proche de son interlocuteur qu'une certaine borne de distance [11].
Les contributions de ce manuscrit peuvent être divisées en plusieurs points :
Premièrement, nous souhaitions fournir une quantité satisfaisante de données expérimentales pour obtenir une vue qualitative du comportement des mesures de temps sur Internet. Ce processus expérimental a commencé durant les 6 premiers mois de cette thèse, et ne s'est jamais totalement interrompu jusqu'au commencement de la rédaction de ce manuscrit. Nous avons pris un soin particulier dans l'étude de la prévisibilité des temps mesurés entre deux nœuds donnés, en rassemblant un grand nombre d'échantillons de mesures pour en observer la distribution.
Deuxièmement, Nous voulions extraire de ces observations un critère de caractérisation pour pouvoir définir un processus de décision prenant en entrée un jeu de mesures, et retournant un bit selon que l'échantillon soit accepté (si aucun relai n'est en cours) ou rejeter (dans le cas contraire). Dans le cas des protocoles de vérification de proximité, ce

processus de décision repose sur le grand principe physique selon lequel aucune information ne peut voyager plus rapidement que la lumière. Cette affirmation peut être exploitée grâce à la grande vitesse de transfert d'informations dans les communications sans contact par fréquences radio, suffisamment proche de la vitesse de la lumière pour pouvoir calculer des distances de façon précise. Une telle méthode n'aurait pas de sens dans le contexte des communications sur Internet, car l'information ne voyage pas en ligne droite vers son destinataire, visite de nombreux nœuds intermédiaires, est traitée par chacun de ces intermédiaires, et n'est même pas contrainte à suivre exactement la même route d'un envoi à l'autre.
Enfin, Nous voulions décrire un protocole de détection efficace, le tester dans des conditions réelles, et montrer par l'implémentation d'un prototype (1) que le processus de décision est précis sur des temps réels et (2) que l'utilisation du protocole pour la supervision de l'envoi d'un grand jeu de données ne cause pas d'impact significatif en termes de débit et de latence.
Cette thèse a mené à la publication de 2 articles :

1. Ghada Arfaoui, Gildas Avoine, Olivier Gimenez, and Jacques Traoré, « How Distance-Bounding Can Detect Internet Traffic Hijack- ing », in : Cryptology and Network Security (2021) [6].
2. Ghada Arfaoui, Gildas Avoine, Olivier Gimenez, and Jacques Traoré, « ICRP : Internet-Friendly Cryptographic Relay-Detection Protocol », in : Cryptography 6.4 (2022) [7].

Introduction

"Power resides where men believe it resides...
It's a trick. A shadow on the wall.
And a very small man can cast a very large shadow."
- Lord Varys

The Internet is structured as an inter-connection of smaller networks owned by different entities (academic, governmental, commercial, or else). These networks are called *Autonomous Systems* (AS) and can be identified by their peers thanks to a unique AS number (ASN). An AS owns a set of IP addresses. Each AS handles its internal communications and has one or more gateways linked to adjacent AS. The communication intra-AS is independently managed by the authority in charge of the AS, while the extra-AS communications are mainly decided from economical agreements and time-delivery optimization. In order to accurately deliver each and every packet of data traveling on the Internet to the right destination node, two main protocols are in use: the *Internet Protocol* (IP) for the data plane and the *Border Gateway Protocol* (BGP) for the control plane.
BGP's task is to make sure every router knows how to forward incoming packets. In a nutshell, BGP allows the Autonomous Systems to construct routing tables by advertising the sets of IP addresses they own and spreading those advertisements along with the ordered list of traversed AS so far. Such an announcement is called a claim, the AS sending a claim is called the origin of the claim, and the path updated each time a new AS receives the claim is called the AS-path. By receiving a claim containing a set of IP addresses, the origin AS and the current AS-path, each AS can keep its routing table up to date by storing the aforementioned set of IP addresses along with the next AS to forward incoming messages to.
The issue with that procedure is that it fully relies on trust, which means that there is no security features preventing an AS to advertise a set of IP addresses that it does not truly own. This kind of misleading advertisements can lead to modified routing tables. When performed on purpose, an action causing the modification of the path an information would have normally traveled is called an *hijacking attack*. A bad BGP advertisement can create different types of threats, for instance:

— An AS falsely claiming a set of IP would disrupt the routing choices of other AS, causing the genuine owners of those addresses to be unreachable for a large portion of the Internet. Moreover, the redirected traffic would instead reach the wrong AS, allowing a potentially malicious use of the obtained data.
— An AS forwarding a modified AS-path can become a relay between multiple couples of AS, potentially aiming to perform man-in-the-middle attacks or just to eavesdrop a critical exchange over the long run.

State of the Art

Since July 1994, when BGP-4 was first described [63], many proposals attempted to prevent hijacking attacks:

Attempts to strengthen BGP. An important part of those attempts rely on the use of asymmetric cryptography to improve BGP, using digital signatures to attest the origin and AS-path of each claim. So, Kent, Lynn, and Seo, published "Secure BGP (S-BGP)" in 2000 [43]. In 2003 White proposed "Secure Origin BGP (soBGP)" [74]. Then in 2005, Wan, Kranakis, and Van Oorschot, followed one year later in 2006 by Karlin, Forrest, and Rexford, respectively introduced "Pretty Secure BGP (psBGP)" [73] and "Pretty Good BGP (pgBGP)" [42].
In 2005 the *Internet Engineering Task Force* (IETF) established the *Secure Inter Domain Routing* working group (SIDR). Since its creation, the SIDR group has worked toward the standardization of the BGPSEC protocol [46] based on S-BGP [43]. BGPSEC takes advantage of a public-key infrastructure to certify AS's key pairs for attesting the digital signatures, then, the basic idea is to ask for each AS receiving a claim to sign its own AS-path so far, along with the next AS to receive the claim. While BGPSEC is probably the proposal receiving the greatest amount of work, it also gets its share of criticism, as it requires a lot of computation from the Autonomous Systems and for every received claim. This could imply the need of large and expensive hardware updates.
In an attempt to avoid the heavy computations due to digital signatures, Hu, Perrig, Johnson, and Sirbu, published a paper [39] in 2003, where the general idea is to use nested Message Authentication Codes (MAC) instead of signatures, and [40] in 2004, describing the Secure Path Vector protocol (SPV) that uses one-time signatures allowing off-line pre-computation to speed up the signing process. However, Raghavan, Panjwani, and Mityagin showed in 2007 that SPV does not actually prevent an AS to forward modified AS-paths.
Gersch and Massey [26] published a work in 2013 in which they addressed the origin AS validation using DNS server. Unfortunately, these works only deal with origin validation. In 2003 Goodell et al. introduced the Interdomain Route Validation protocol (IRV) [29], further improved by Chen and Haeberlen in 2015 [19]. This solution proposes the addition of a dedicated server called an IRV server in each AS. IRV servers would store the AS policies and owned IP addresses. Then, a received claim can be checked by asking the related IRV server. However, this approach is limitated by its lack of scalability.

Attempts to change the routing architecture. Facing the lack of fully satisfying solutions so far, an emerging field of research named the Path-Aware Networking (PAN) emerged. PAN aims to redefine the very architecture of Internet routing so that the new architecture does not suffer from the same routing issues due to BGP. Noticeably, a proposal named SCION seems to stands out with a particularly invested work [20, 54]. SCION's idea is to no longer regroup nodes into Autonomous System, but into entities called Trust Domains (TD). A Trust Domain is administrated in the same fashion as

Autonomous Systems, but follow a nested structure of sub-TD. As its name suggests it, a Trust Domain only contains entity trusting each other which ultimately leads to an architecture where the top-level trust domains represents well identified groups like international alliances, or industries in active partnership. Of course, translating from a well known architecture to a completely new one, will take time, if it ever happens, which means that such a solution can not be practical in the near future.

Attempts on anomaly detection. If those researches seem insufficient to efficiently prevent BGP hijacking, many proposals tried to provide detection instead of mitigation. Some works revolve around the monitoring of suspicious BGP claim, for instance, when similar IP addresses are being claimed by more than one AS [59, 45, 58, 69]. Other works are machine learning-oriented, [3, 31]. Finally, an interesting approach was proposed by Hiran, Carlsson, and Shahmehri in 2015 [36]. The authors presented a framework called CrowdSec, where users are passively gathering round trip times between their stationary machines and IP addresses claimed by another AS. Assuming the first measurements to be genuine, the users independantly apply Grubb's statistical test [30] to identify outlying measures. However, this process does not detect an attacker being able to intercept messages and to answer in a time satisfying the Grubb test.

Contributions

This thesis is dedicated to the conception of a new detection technique for relay attacks. By nature, any hijacking causing messages not to be delivered to the intended receiver is very quickly detected. For this reason, the work described in this manuscript will address the issue of stealthier relay attack. We believe those attacks to be extremely severe in terms of privacy violation and geopolitical repercussions. We define a "long term relay attack" by the modification of a genuine route between a couple of fixed nodes, such that the attacker can eavesdrop every message between the two nodes and does so without attempting to modify the content of the exchanges. In this situation, the attacker plays a passive role, and seeks to gather as much information as possible, for the longest possible period of time. Ultimately, such attacks can be used at an international scale to obtain important metadata, allowing blackmail under the threat of unveiling private information, or discrediting a government or an industry. It can also be used to cause economical damages. For instance, in 2017, Apostolaki, Zohar, and Vanbever, showed that relaying bitcoin messages in order to delay block propagation can entail critical financial losses [5]. We present the Internet-friendly Cryptographic Relay-detection Protocol (ICRP). ICRP is a 2-party protocol detecting long-range relay attacks by taking advantage of the measurement of round trip times in a way similar to [36]. However, ICRP also incorporates an interactive authentication, preventing any attacker to trick the measured times. This additional cryptographic layer is directly inspired from the distance-bounding protocols, another relay detection technique applied to a completely different environement, i.e. contactless communications.

Distance-bounding is an interactive authentication protocol, ensuring not only that the device trying to authenticate itself is genuine, but also that this device is physically located closer than a given distance bound [11].
The contributions of this manuscript can be divided in multiple categories:
The first category was to provide a sufficient amount of experimental data to ensure a qualitative view on the behavior of time measurements over the Internet, this experimental process started during the first 6 months of this PhD and never really stopped until the beginning of the redaction of this manuscript. In particular, we took interest in the predictability of the measured time between two fixed nodes by gathering a fair amount of time traces, and observing the distribution of such traces.
The second category was to search for a way to characterize those time samples in order to define a decision process, that would take a time sample as an input and return a bit whether the sample gets accepted (if there is no ongoing relay attack) or rejected (otherwise). In distance-bounding, this process relies on the axiomatic law of physics, stating that nothing travels faster than light. This statement can be used thanks to the very fast transportation of radio frequency contactless communications, that are close enough to the speed of light to allow a direct distance computation. Such a method can not be considered in Internet communications where information does not travel in straight lines, visits multiple intermediary nodes, is processed by each one of these intermediaries, and is not constraint to follow the exact same route from one exchange to another.
The third and last category was to test the protocol with real condition and to show, through the implementation of a prototype, that (1) the decision process was accurate on real time samples, and (2) the use of the protocol does not create significant loss in terms of throughput and latency when supervising the sending of large files.
During this PhD, 2 articles were published:

1. Ghada Arfaoui, Gildas Avoine, Olivier Gimenez, and Jacques Traoré, « How Distance-Bounding Can Detect Internet Traffic Hijack- ing », in: Cryptology and Network Security (2021) [6].
2. Ghada Arfaoui, Gildas Avoine, Olivier Gimenez, and Jacques Traoré, « ICRP: Internet-Friendly Cryptographic Relay-Detection Protocol », in: Cryptography 6.4 (2022) [7].

Outline of the rest of the manuscript

Chapter 1 presents the background needed for the rest of the manuscript. It precisely describes the way information is routed through the Internet and how hijacking attacks can be performed. It provides more specific details about some of the more noticeable countermeasures, and also presents and defines the distance-bounding protocols.
Chapter 2 elaborates on the experiments made on the Internet. The measurements are made between nodes set up on different locations on Earth to observe if the distance (whether geographical or defined by the route length) separating the nodes has an impact on how the collected measures are distributed. Each measurement are grouped depending

on the date and time they were gathered, such groups are called samples. The samples are separated in 3 categories: the punctual samples, containing measurements gathered during a very short time, the continuous samples, containing measurements gathered at regular intervals for a long period, and the spreaded samples which are the concatenation of multiple punctual sample. Each experimental result is presented and discussed.
Chapter 3 formally describes the protocol. ICRP has 2 tasks to handle, the first one is to gather a time sample during the interaction between the 2 party and to test the validity of the times using a so-called decision function. The second one is the authentication, this task is crucial to make sure that the sample of time obtained during the execution corresponds to an exchange between the genuine parties. For practical consideration, a detailed description of a prototype implementation is provided as well, along with an analysis of practical performances.
Finally, Chapter 4 provides a cryptographic proof of security in the Random Oracle Model. This proof mainly relies on the work of Boureanu et al. [15], in which the authors precisely described a complete threat model for distance-bounding protocols.

Chapter 1

Relay Attacks and Countermeasures

"It was the singers who taught the First Men to send messages by raven...
But in those days, the birds would speak the words.
The trees remember, but men forget.
And so now they write the messages on parchment
and tie them round the feet of birds who have never shared their skin"
- Brynden Rivers

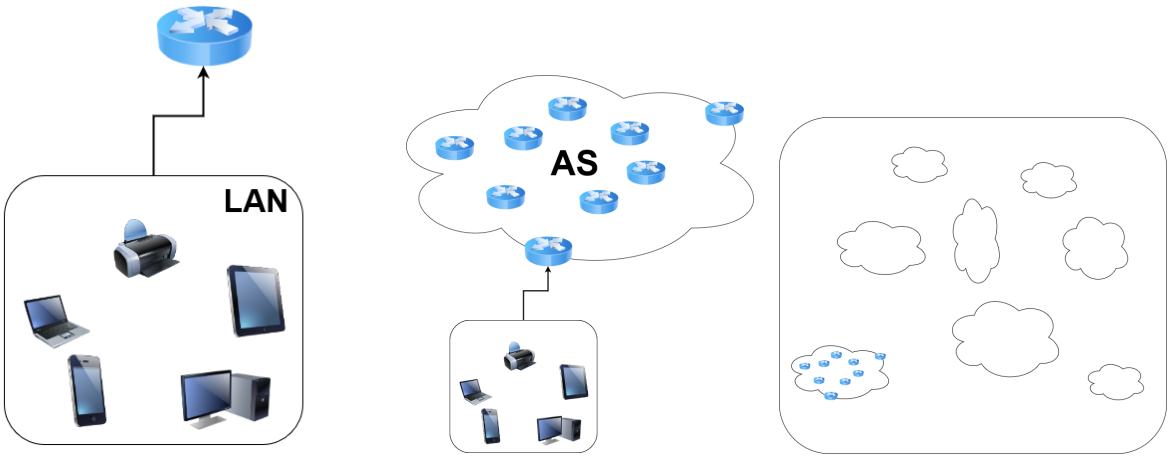
Introduction

This chapter will cover the subject of the Internet routing protocols and the existing methods to deviate traffic, called hijacking attacks, as mentionned in [12]. It will then explore the state of the art regarding mitigation of such attacks. Section 1.1 presents the Internet structure and routing mechanism. Section 1.2 describes a kind of attack for hijacking data over the Internet and how it can be used to set up a long term relay. Section 1.3 presents the state of the art of the countermeasures for those attacks. Section 1.4 elaborates on relay attacks in the context of contactless communications and the solutions suited to counter them in this specific environment. Finally, the conclusion of this chapter considers the challenges to overcome if a contactless relay countermeasure was to be adapted against Internet hijacking.

1.1 Routing over the Internet

1.1.1 Structure Overview

The Internet allows any 2 connected devices to communicate thanks to an organized topology based on a sub-network structure. Figure 1.1 illustrates this structure, which is then described in the next sections.



(a) Local Area Network (b) Autonomous System (c) The Internet

Figure 1.1 – Overview of the Internet structure from LAN to AS level

Local Area Network

A Local Area Network (LAN) is the biggest possible zoom on the Internet structure, it involves typical user devices such as personal computer, smartphones, smart TVs, or other connected devices. Each machine of a LAN can join another LAN thanks to a dedicated networking equipment called router. Roughly speaking, a router is a gateway to whatever lives outside the LAN. Its purpose is to forward any piece of data to the right path leading to the targeted receiver LAN. The routers directly linked to a LAN are often qualified to live "on the edge" of the Internet, as they are the very first routing equipment to go through for reaching another LAN.

Autonomous System

Each LAN's router itself belongs to a wider network called an Autonomous System (AS). Each AS is referenced by a unique identifier called the AS Number (ASN), it is handled by an authority, which is most of the time an Internet Service Provider (e.g. AT&T, Orange), or other telecommunication related companies (e.g. Amazon, Cisco), or even governmental institutions. More precisely, an AS is a network with a cohesive and independent routing policy. Autonomous Systems are the widest individual entities present on the Internet,

and they also possess one or multiple gateways allowing them to communicate one to another. According to the Number Resource Organization (NRO) and their quarterly Internet Number Resource status report [53], the number of allocated ASN in June 2022 was around 119700. Together, all the existing ASes form the Internet.

Routers

The routers are the devices transmitting information from one node to another. They can be classified in 2 categories:

— Edge routers: they are the equipment available for the general public, it usually has 2 interfaces, one interface for the LAN it is linked to, and another one to access the related AS. These have processing capabilities suited to a recreational use of Internet. Other edge routers can have much better processing capabilities and are generally distributed to industries needing to connect multiple offices and, thus, to manage a lot more of traffic. In both cases, they are the most external routing equipment of the Internet, which is why they are designated as "edges".

— Core routers: they are the fastest equipment in terms of throughput, and form the so-called "Internet Backbone" or "Core Network". They are used specifically to deal with the forwarding of information when it gets to the areas of the web where the demand of throughput and bandwidth is the highest.

1.1.2 Routing Overview

While some AS owns only a few hundreds of routers like ASN 44495, some are managing over several millions, for instance ASN 3356, owned by LEVEL3, is the largest Autonomous System in the world with over 37 millions managed equipment (see [33]). In order to maintain a fluent communication between all those devices, 2 protocols are in use: the Internet Protocol (IP) for the data plane and the Border Gateway Protocol (BGP) for the control plane.

Internet Protocol (IP)

The Internet Protocol [55] is based on an addressing system which allows identifying any existing router in its network. The address of a connected machine or router is simply called an IP address and is defined by 4 bytes [1], usually written in decimal notation for human readability's sake (e.g. 192.168.1.1). Given this 4 bytes format, note that the total number of possible IP address rises up to 2^{32} which is approximately 4 billions. Over the years, the number of connected devices encountered a significant increase with the modern accessibility of Internet for the common people and, more recently, with the birth

1. The IP norm described in this section is the 4^{th} iteration of the Internet Protocol IPV4. Noticeably, the transition the the 6^{th} version is slowly taking place. This particular version uses addresses encoded on 16 bytes.

of the Internet of Things. Consequently, the number of available address was dangerously approaching 0 and an update of the standard IP address was to be made.
An ingenious idea was applied to deal with this lack of address: The principle of Network Address Translation (NAT). The idea is to take advantage of the fact that a device on a LAN does not need to have a unique identifier worldwide, but only in its local environment. From this observation, dedicated ranges of addresses can be set to define private networks. For example, the classical private home network is usually ranged from 192.168.1.0 to 192.168.1.255. In that case, the first 24 bits serves as the identifier of the network, while the other 8 designate the devices connected on this network, leaving 256 potential devices connected at the same time on this network.

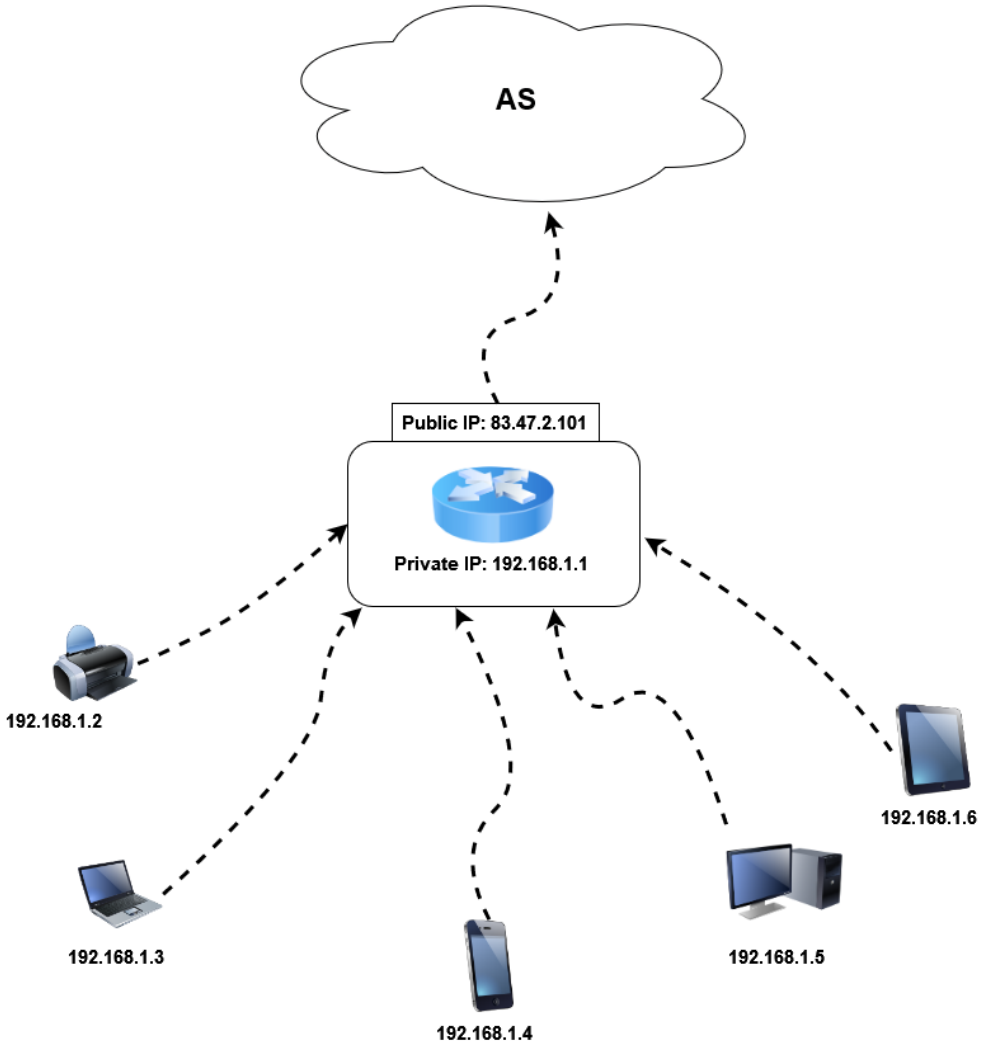


Figure 1.2 – A LAN with private and public IP addresses

This setup is suitable for personal environments with few users. For more populated

environments (e.g. industries), the network identifier can be limited to a smaller number of bits to allow more connected devices. To ease the notation and the separation of the network and devices identifiers, an address range uses an abbreviation by just indicating the first address in the range, followed by the number of bits defining the network identifier, e.g. the range from 192.168.1.0 to 192.168.1.255 is then denoted 192.168.1.0/24. This notation is called an IP prefix.
In that configuration, all the devices present in a private network are given an address, which is called a private IP. The related router also gets its own address for its private interface, but has a public address as well on its second interface. This public address is the unique identifier of the local network. When a message comes for a specific connected object from an external machine, it is sent to the public IP of the router, which redistributes it accordingly. Figure 1.2 pictures an illustration of the IP addressing system.
Along with the IP address standard, the Internet Protocol designates a process of encapsulation designed to provide routers important routing information. When a message m is sent from a node, it is concatenated on the left by this information. This concatenated part is called the IP header, and the all concatenation is called a packet. Figure 1.3 displays the content of an IP Header.

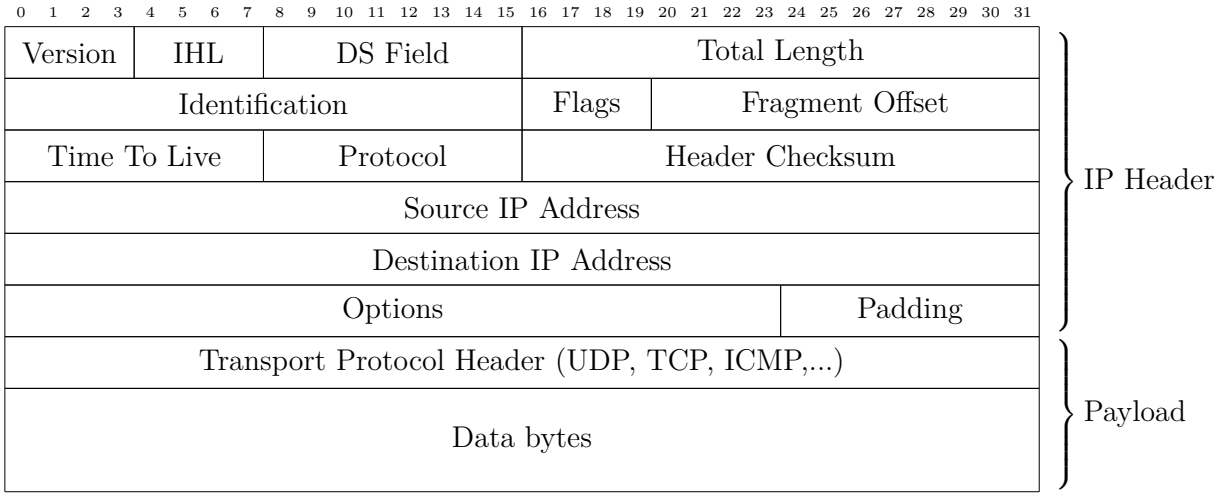


Figure 1.3 – IPv4 Header for 32 bits IP addresses.

The following describes the principal fields in the IP header:

— Version: designates the IP version used for this packet (typically IPv4 or IPv6).
— IHL: designates the header length. This length is variable due to optional fields and padding.
— Total Length: designates the size of the full packet. Note that the 16 bits space of the field implicitly defines a maximum packet size of 65536 bits.
— Time To Live: designates the total number of routers that can be visited before throwing the packet.
— Protocol: designates the underlying transport protocol (typically TCP for reliability, UDP for fastness, or ICMP for debugging).

— Source and Destination addresses: designates the public IP addresses of the source and destination nodes.

Border Gateway Protocol (BGP)

The Border Gateway Protocol [63] is a "Path-Vector" routing protocol. Path-Vector routing is a process where the path leading to a destination is constructed from the broadcast of a path-vector message. This path-vector is accumulated with the identifiers of the entities forwarding the message. In the case of BGP, this process is applied at the AS level, and the path vector is updated with the AS numbers. Along with the path-vector is sent a claim on an IP prefix so that any AS receiving a forwarded claim can learn the AS-level path reaching the prefix and choose whether to update its routing table accordingly and forward it, or to reject the path and drop the message.
Let's state here a few important notions:

1. A claim for an IP prefix with the Border Gateway Protocol is called a BGP announcement.
2. Neighboring ASes are linked with a direct connection between specific routers called BGP routers.
3. Connections between BGP routers is derivated in 2 categories:
   — e-BGP links: stands for external-BGP links. They support the exchange of BGP announcements between 2 BGP routers within different ASes. 2 routers connected through an e-BGP link are called e-BGP peers.
   — i-BGP links: stands for internal-BGP links. They support the exchange of BGP announcements between 2 BGP routers within the same AS. 2 routers connected through an i-BGP link are called i-BGP peers.
4. i-BGP links are specifically dedicated to the transmission of BGP announcements within an AS and are independent of the data links used for standard communications. Every BGP routers inside an AS are always one hop away (physically or virtually) from each other.
5. Every router on the Internet manages 2 different routing tables:
   — An internal routing table that allows to route traffic among routers inside a single AS. An entry in the internal table indicates a destination node's IP address within the AS, and the IP of the next hop to forward the packet to. These tables are constructed with an Internal Gateway Protocol (IGP) like OSPF, RIP, or EIGRP. The choice of the IGP is independently decided by the authority managing the AS. In a nutshell, IGPs allow each router of an AS to learn the next hop to send a packet to in order to reach any other router within the same AS.
   — An inter-domain routing table that allows to route traffic to an external AS. It indicates a destination IP prefix claimed by another AS, and the IP of the first BGP router to forward the packet to in the next AS.

Figure 1.4 illustrates a toy example topology for 6 Autonomous Systems labeled from $AS1$ to $AS6$. Each AS gets 2 BGP routers, with i-BGP links represented by dashed lines and e-BGP links represented by plain lines. The internal topology of $AS1$ is displayed in detail, with internal links in thin lines. Here, $AS2$ owns the IP-prefix 32.24.128.0/17 (which is the set of IP address going from 32.24.128.0 to 32.24.255.255).
In the following, for readability's sake, this IP prefix will be referred to as $pref$, and the IP addresses of every displayed node will be referred to as the corresponding router's name. For this example, assume that $AS2$ wishes to connect the IP addresses covered by the prefix so that any machine on this network can join them.

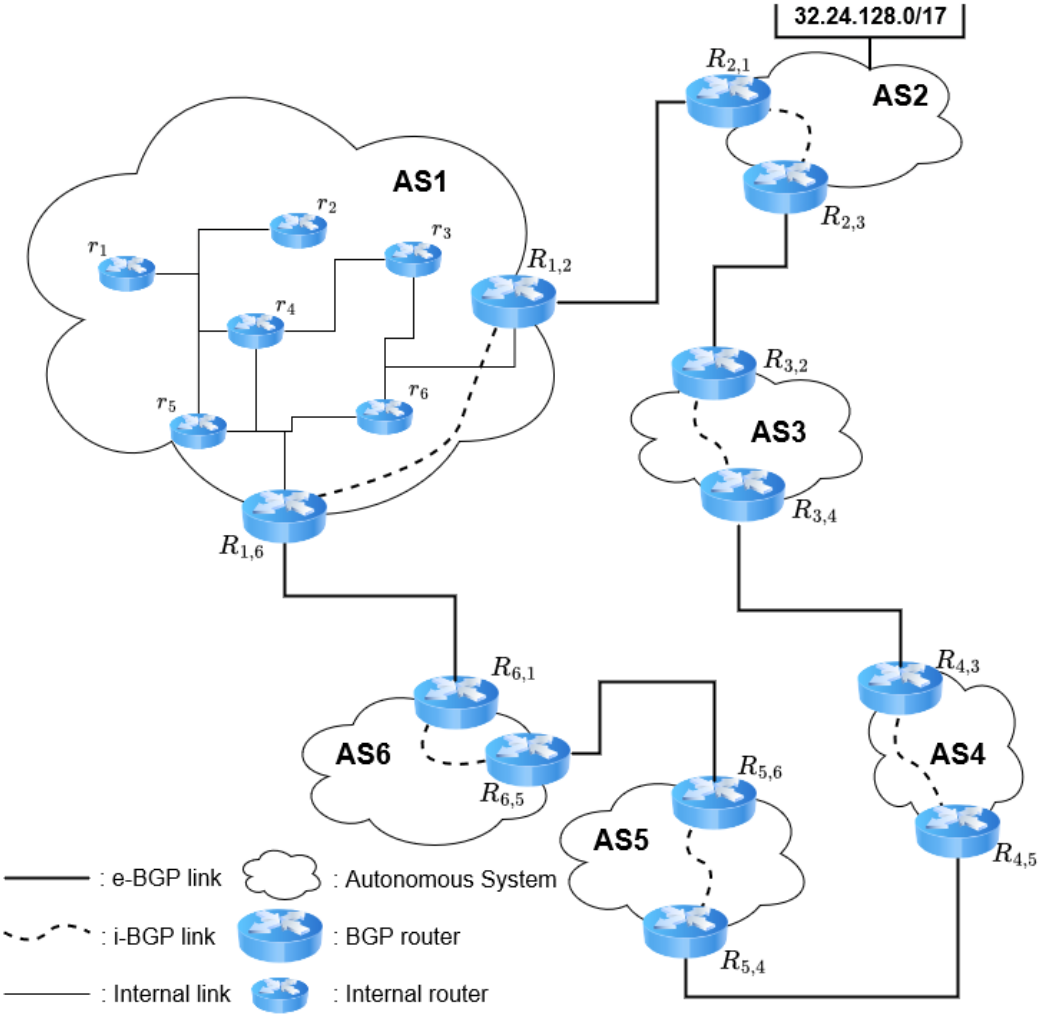


Figure 1.4 – Internal and External Border Gateway Protocol

BGP announcements carry all the information needed by the routers to complete their inter-domain routing table. Let's firstly take a look at the format of a BGP announcement on prefix $pref$ while $AS2$ uses BGP to broadcast it through the network.

A BGP announcement must contain at least 3 important fields:

1. The aforementioned IP prefix $pref$. This allows every node receiving the announcement to know which IP addresses will be reachable through the constructed route.

2. A path vector of ASN (denoted AS_PATH) that will contain the ASN of every visited AS so far. This allows ASes to choose whether to keep or drop the announcement, depending on their routing policies.

3. The IP address of the last visited BGP router before leaving the previous AS (denoted $next\_hop$). This allows the routers inside an AS to complete their inter-domain routing table.

Let's follow the full procedure step by step while the announcement travels through $AS2$ and $AS1$. $AS2$ creates its BGP announcement for the prefix $pref$, with the path vector $AS\_PATH = [2]$, leaves the field $next\_hop$ empty and broadcasts $(pref, [2], empty)$. When reaching $R_{2,1}$, the $next\_hop$ field is filled with its IP address as it is the last reached node before leaving $AS2$. The announcement $(pref, [2], R_{2,1})$ is then forwarded to $R_{1,2}$, the BGP router of $AS1$ neighboring $AS2$.

Upon receiving the claim, $AS1$ will perform the following actions:

— $R_{1,2}$ checks the internal routing policies of $AS1$. If the policies match the claim, it updates the AS_PATH field by adding its corresponding ASN. The new BGP announcement is now $(p, [2, 1], R_{2,1})$.
— $R_{1,2}$ then updates its inter-domain routing table according to the information within the announcement, namely, it sets the destination to the prefix $pref$ and the next AS-level hop to $R_{2,1}$. It then shares the inter-domain table update to every standard router within the AS.
— $R_{1,2}$ forwards the announcement to its other internal BGP router $R_{1,6}$ through the i-BGP link.
— As the announcement came from an Internal neighbor, $R_{1,6}$ forwards the claim to its neighboring BGP router: $R_{6,1}$.

Note that any router in $AS1$ can now learn how to forward packets to an IP address within prefix $pref$ by merging information from its internal and inter-domain routing table. Indeed, the inter-domain table informs that the next AS-level hop is $R_{2,1}$, and the next internal hop to reach $R_{2,1}$ appears in the internal table.

The procedure continues in the same fashion for each AS receiving the announcement. Table 1.1 summarizes the announcement received and forwarded by the AS $1, 3, 6$ and 4 along with the update entry in their inter-domain routing tables.

Table 1.1 – A BGP announcement spreading out in the network topology of Figure 1.4

ASN	Received	Inter-domain table update		Forwarded
		Destination	Next hop	
1	$(pref, [2], R_{2,1})$	$pref$	$R_{2,1}$	$(pref, [2,1], R_{1,6})$
3	$(pref, [2], R_{2,3})$	$pref$	$R_{2,3}$	$(pref, [2,3], R_{3,4})$
6	$(pref, [2,1], R_{1,6})$	$pref$	$R_{1,6}$	$(pref, [2,1,6], R_{6,5})$
4	$(pref, [2,3], R_{3,4})$	$pref$	$R_{3,4}$	$(pref, [2,3,4], R_{4,5})$

Observe that $AS5$ ends up receiving 2 different announcements, namely:
— $(pref, [2,3,4], R_{4,5})$ from $AS4$
— $(pref, [2,1,6], R_{6,5})$ from $AS6$
Once again, the AS will check on its internal policies to see if those claims match and decide to keep one, or possibly drop them both. Internal policies of an AS can (and most of the time do) prioritize their routing choices on commercial or political agreements. This means that the final choice of an AS will not always be the fastest or smallest route to join the destination.
In that specific case, $AS5$ receives 2 announcements with the exact same path length, its choice will then be uniquely defined by economical, political or technical preferences. Assume that $AS5$ has an ongoing partnership with $AS4$, it consequently chooses to keep and forward the announcement $(pref, [2,3,4,5], R_{5,6})$ to $AS6$. Having already updated its routing table on that prefix, $AS6$ will also check its routing policies and decide to drop the announcement.

1.2 Hijacking and Relay Attack

1.2.1 Hijacking incidents

A critical issue in this inter-domain routing procedure lies in the lack of authenticity requirements regarding the BGP announcements. Indeed, in its current version, BGP does not prevent an AS to broadcast false information on prefixes or AS paths. This kind of false claim produces unintended route alterations leading to a wide range of possible consequences, from putting down websites for a few hours to major leaks of confidential data. An attack aiming to cause such routing alterations is called an hijacking attack. This kind of misleading BGP announcements has become more and more frequent over the last decades. On February 2008, Youtube became unreachable for two hours after Pakistan Telecom falsely claimed being the better route for joining it [64]. A striking breakdown happened on April 2010, when China Telecom advertised wrong traffic routes: for approximately 20 minutes, no less than 15% of the Internet traffic adopted those routes, including some traffic of the US government, military sites and commercial sites like Yahoo! and IBM [9], this incident raised a clear geopolitical concern on what information could have effectively been collected afterwards. More recently, in June 2019, the same

kind of incident with China Telecom occurred for about 2 hours [8]. In 2017, it was showed in [5] that hijacking bitcoin messages in order to delay block propagation or to isolate mining pools from the network can entail critical financial losses. Aside from the direct consequences of an hijacking, it is also complicated to distinguish a real malicious attack from a genuine routing error if the incident is even detected. Consequently, the Autonomous Systems cannot fairly establish punitive responses towards an AS advertising false claims. Note that all those incidents induced a lot of chaos and were easily noticeable. For that reason, the route alteration did not last more than a few hours. However, this technique can also be used by a stealthy attacker to create a long term relay and attract traffic towards him, this could result in man-in-the-middle attacks, metadata gathering, and industrial or governmental espionage.

Let's take a closer look at an attacking strategy allowing an AS to perform a relay.

1.2.2 Relay with a Prefix Attack

A prefix attack on BGP is defined by an AS whether claiming a prefix already owned by another AS or broadcasting an announcement on a non-existing route. Because ASes have no means to distinguish genuine from false claims, a large portion of the network ends up choosing to update their routing tables according to the bad announcement. An attacker having insights on the topology and policies of certain AS can use a bad BGP announcement to set up a stealthy relay.

Consider the configuration displayed on Figure 1.5.

For this example, assume that every Autonomous Systems in this topology have only 3 policies for chosing or rejecting a received path:

1. If there is no entry for the prefix in the routing table, the AS keeps the first received path.

2. If there already is an entry for the prefix in the routing table, the AS keeps the path with the smallest length.

3. If the received path is longer than the one already stored, the AS does not forward the announcement.

Here, $AS1$ owns the prefix 81.76.0.0/16 and claim it through a BGP announcement. $AS2$, which is the unique direct neighbor of $AS1$ receive the claim with $AS\_path = [1]$. $AS2$ then broadcast the announcement to $AS3$ and $AS4$ with path $[1, 2]$ and so on. The paths received by the ASes are shown in the figure, and the ones chosen by the AS is marked with a "✓".

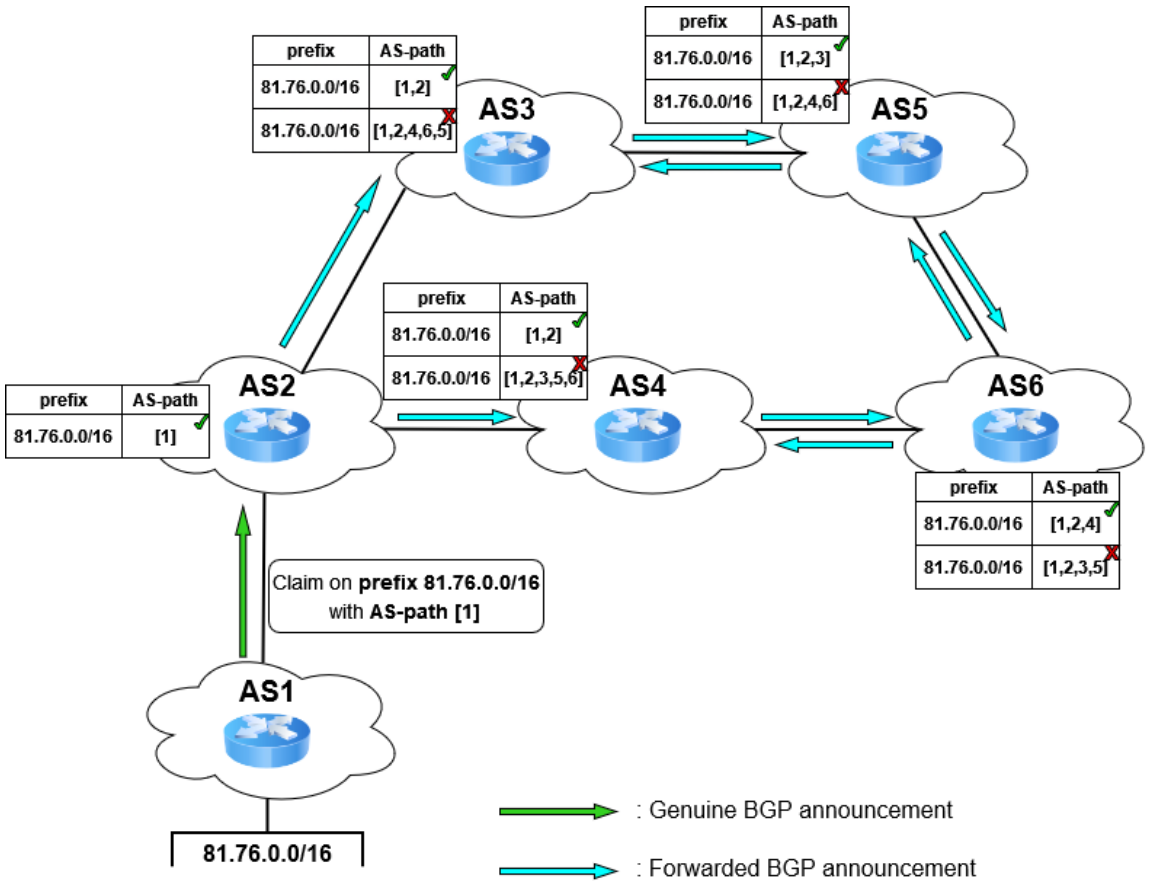


Figure 1.5 – $AS1$ claiming its own prefix and the travel of the corresponding BGP announcement

Now, assume that $AS6$ broadcasts a new BGP announcement on the same prefix, claiming to be a direct neighbor of $AS1$. To do so, it just sends the path $[1, 6]$. This path does not exist in reality, but there is no way for the other AS to deduce it, so the announcement travels normally inside the network and is treated like any other claim by the ASes. Figure 1.6 shows the evolution of the announcement in the network.

$AS4$ receives the path $[1, 6]$ but has already stored a path of length 2, therefore, it discards this announcement. $AS5$ however, had a path of length 3 and then accept the new path offered by $AS6$. It forwards the announcement with path $[1, 6, 5]$ to $AS3$ which also discards it, for it already has a path of length 2.

Consequently, as long as this operation stays unnoticed, all the traffic towards the targeted prefix passing through $AS5$ will ultimately be routed through $AS6$.

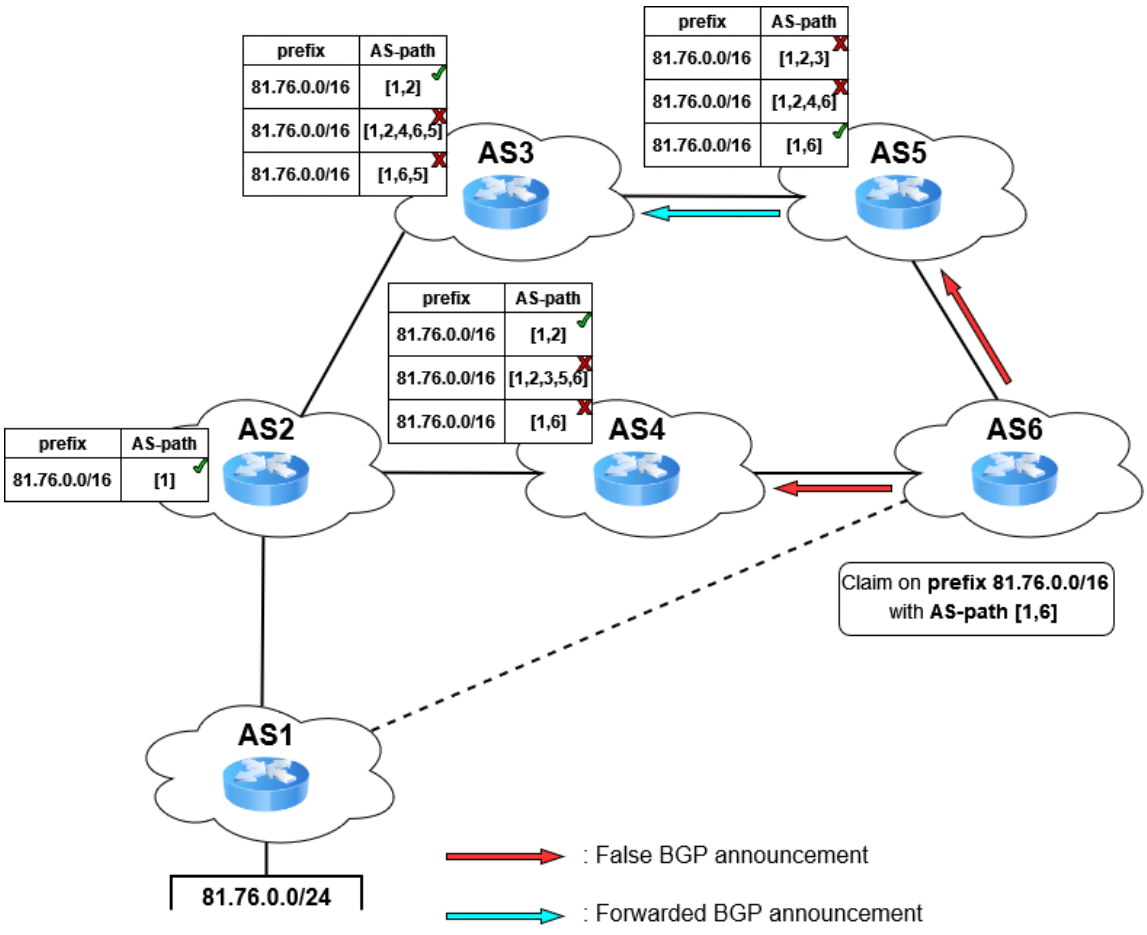


Figure 1.6 – A relay between $AS5$ and $AS1$ using a prefix attack

Please note that this example is a simplified view of an attack. In practice, an attacker must gather a precise knowledge of both the topology surrounding him and his victim, and the policies of the involved ASes. However, this scenario is still completely credible, given that ASes are owned by powerful institutions with large financial means. Also note that the attack presented here is not the only way to re-route traffic, another famous example called the subprefix attack takes advantage of the fact that a BGP announcement for a prefix covering a smaller range will always be chosen over one covering a larger range. For instance, in the topology presented with Figures 1.5 and 1.6, $AS6$ could have sent a claim on prefix 81.76.0.0/24 instead of 81.76.0.0/16 (the former only covering IP addresses from 81.76.0.0 to 81.76.0.255). This "smaller" prefix would have been chosen by every AS, regardless of the path length. However, this route alteration would probably be detected way quicker as it impacts a larger portion of the network.

1.3 Countermeasures

1.3.1 BGPSec

Description

Since July 1994, when BGP-4 was first described [63], many proposals using asymmetric cryptography tried to enhance the protocol [43], [74], [73], [42]. All these contributions aimed to strenghten BGP by working on the possibility to validate both origin and path sent with BGP announcement between ASes in order to prevent false or misconfigured claims. The *Internet Engineering Task Force* (IETF) initiated the BGPSec standardization project [46] based on Secure-BGP [43]. The key idea is to use a public-key infrastructure to certify the origin of an IP prefix and to allow the ASes to use digital signatures to authenticate their announcements and validate the AS-paths. In simple words, a digital signature is a value generated from a private key, and a given piece of data. This value can only be computed by the entity owning the secret key sk, but the validity of a signature can be verified by any other entity using the public key pk corresponding to sk. Hence, the digital signature provides both the authenticity and the integrity of a message.
A BGPSec announcement contains a so-called BGPSec-path instead of the AS-path of classical BGP. This BGPSec-path includes a list of signatures preventing any tampering on the AS-path. More precisely, an AS originating a claim on a prefix $pref$ with a BGPSec announcement adds a signature on $pref$, the current AS-path, and the ASN of the next AS receiving the claim, as shown in Figure 1.7. Because a digital signature can only be computed by the owner of the corresponding private key, every step of a received path can then be checked.

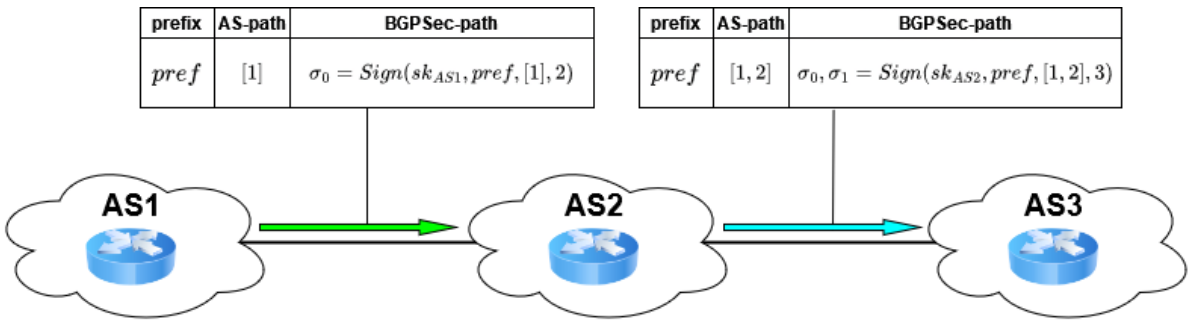


Figure 1.7 – BGPSec announcements

Drawbacks

The main downfall of BGPSec (and of every other attempt using attestation or signature that haven't been discussed here) is the demand of computational capabilities implied by the signature and verification part. Indeed, BGP routers are constantly receiving and treating new announcements, and are not designed to perform cryptographic encryption. This means that a strong hardware update is required. Current works on that matter are

aiming to lower this computational impact and data overhead using different signature schemes or by using paradigms like aggregation. Another issue caused by BGPSec is that claiming multiple IP prefixes in a single announcement is no longer possible, due to the fact that the prefixes are now signed. Multiple prefixes announcement is a frequent event with standard BGP, it decreases greatly the number of simultaneously traveling BGP packets. Adding the origin signature on multiple prefixes would force the ASes whether to accept every prefix or to refuse them all, as a signature on a subset of the prefixes can not be generated. Consequently, each claim on a single prefix is done independently, inducing an important workload for the network.

1.3.2 Path Aware Networking

The final attempt to overcome those current routing issues presented here revolves around a complete reworked of routing architecture. Path Aware Networking (PAN) is emerging as a novel way of thinking routing architecture, allowing more accurate knowledge on the path traveled by data. This clean slate redesign of the routing methods have given birth to a lot of proposals in the last 15 years [18, 76, 62, 27, 4, 54]. An important goal for those architectures is to achieve precise and, above all, trustworthy path tracking of the traversed routers during the sending of a packet.

Among those works, the SCION (Scalability, Control, and Isolation On Next-generation networks) architecture [54] might be the most promising one, with an active calendar since its first steps and recent publications [20] and rewards.

SCION Structure

The SCION architecture partitions the Autonomous Systems into smaller domains called Trust Domains (TD). TDs contains multiple entities willing to trust each other (e.g. 2 Internet Service Providers based in the same country or state, sharing common juridical requirements). A TD is hierarchically organized into several Autonomous Domains (AD), that is a domain managed by a single entity, typically an ISP. Each TD designate a sub set of ADs, called the TD Core, forming a gateway to other TDs. Finally, Trust Domains can be nested in larger TDs and a TD not belonging to any larger TD is called a top-level TD. Figure 1.8 shows an illustration of a Trust Domain structure.

According to the authors, this structural change of Internet topology provides a humanly manageable TD-level routing, as regrouping trusted entities together should limit the number of top-level TD to a few hundreds. Top-level TDs would then correspond to easily identifiable groups, such as well-defined geopolitical areas or international organizations. In that case, instead of using a path-vector routing protocol between the TD Cores, one can use a link-state protocol where the exact TD-level topology is learned by every involved party. Overall, this would offer a natural isolation of smaller domains and the capability of choosing specific path for specific traffic.

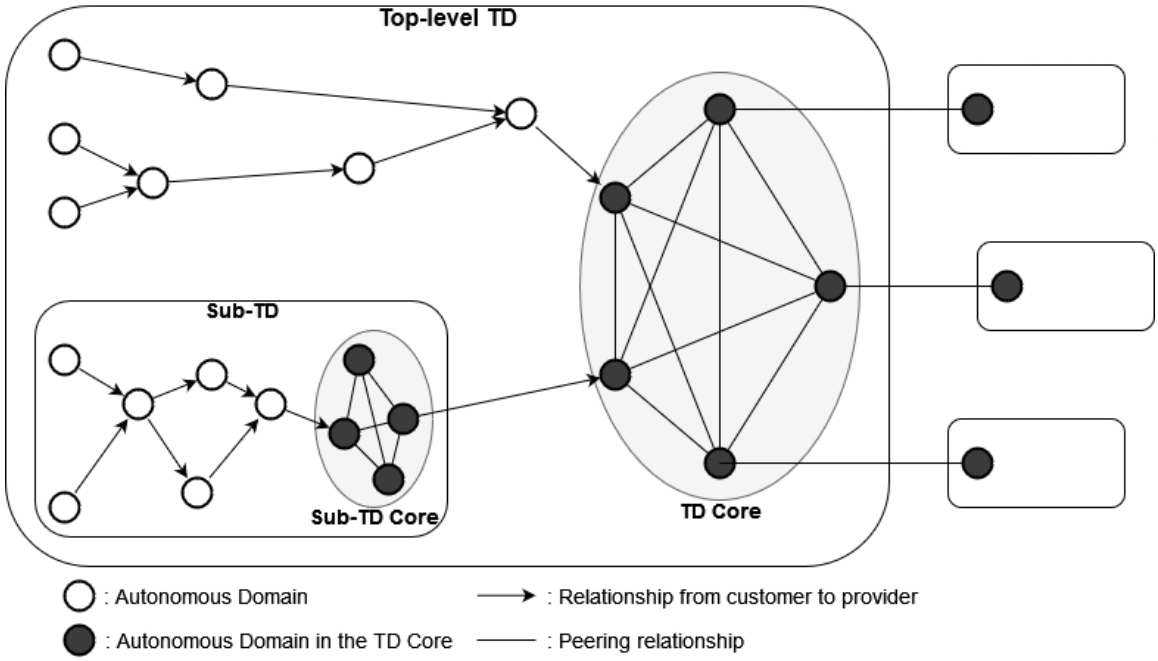


Figure 1.8 – SCION Trust Domains hierarchical structure

Drawbacks

While a complete redesign of the Internet routing architecture might just be the perfect long term solution, a complete and worldwide adoption of such a strong change needs an important involvement from the scientific community in order to gain trust. This implies that many years (not to say decades) of work remain necessary to obtain an acceptable confidence and maturity.

1.3.3 Other Attempts

Other proposals relying on different ideas also exist:

- Solutions based on symmetric cryptography [39, 40, 77, 17] use Message Authentication Codes (MACs) as a replacement for the signature of BGPSec to gain on the announcement's length but paradoxally pays this gain with a significant overhead induced by the need of exchanging additional state information.
- Solutions based on the use of Domain Name System (DNS) [26] for validating the origin of a BGP claim. This does not address the validity of a forwarded AS-path, and transfers the problem of implicit trust (that BGP suffers from) into the DNSSEC protocol.
- Solutions acting outside BGP with each AS getting a so-called Interdomain Route Validation (IRV) server handling AS policies and where each router can contact its corresponding IRV to check the validity of an origin ASN [29, 23]. While those

solutions provide satisfying results with few to none computational or storage overhead, they can hardly adapt to dynamic route changes. This could cause latency on the adoption of new valid routes. It also requires to update the hardware routing equipment.

— Solutions based on Multi-Party Computation [32, 10] separate the route computations between multiple servers inside each AS. These solutions allows fast convergence, but might be hardly scalable and induces a noticeable computational overhead.

The details of these works goes out of the scope of this manuscript, although, the curious reader will learn more referring surveys [51] and [2].

1.3.4 Detection techniques

Instead of trying to prevent erroneous routing behaviors, many works aimed at providing satisfying anomaly detection methods. The authors of [2] classified the main existing approaches in five different categories:

— Techniques using Time Series Analysis:
Time series analysis is the analysis of data points recorded at consistent intervals over a defined period of time. It allows witnessing the evolution of said data whether to forecast a future event or to detect unexpected events. A prior work by Labovitz *et al.* [44] applied Fast Fourier Transform to routing update rates. They monitored data over 9 months from 5 Internet Exchange Points, demonstrating a correlation between rapid routing updates changes and instability of the network. In 2008, Mai *et al.* [48] proposed a framework called BAlet using the Wavelet transform [1] and a clustering method to identify the location of the source of an anomaly. The same approach using the Wavelet transform was also taken by Prakash *et al.* [57] with a tool called BGP-lens. BGP-lens graduate the detected anomalies on a three-level scale depending on the gravity of the anomaly. Although some of these works are able to both detect some anomalies and locate their causes, they do not address real time detection.

— Techniques using Machine Learning:
Machine learning oriented approaches allow treating upcoming BGP updates information using a decent amount of trusted data. Li *et al.* [47] provided a framework based RouteViews [71] and RIPE NCC [65] data, and using a decision tree obtained from the C4.5 algorithm [75]. The authors of [72] used the same technique but with a larger range of data mining algorithms. Al-Rousan and Trajković [3] proposed a classifier for BGP updates based on raw data, also obtained from RouteViews and RIPE NCC, and demonstrated that the volume of an update message is a more critical feature than AS-PATH to detect abnormal behavior. The authors of [2] state that none of those works address the detection of malicious BGP hijacking.

— Techniques using Statistical Pattern Recognition:
These techniques consider raw BGP data as being observations from statistical

experiences to identify patterns and recognize abnormal behaviors. Huang *et al.* [41] proposed a technique to detect failures at the node, link, or peer level from the observation of BGP updates, operational mailing list and routing configuration. This approach succeeds in the recognition of the source of the failure, but need specific information about the routers' configuration. Deshpande *et al.* [21] used the Generalized Likelihood Ratio Test on individual routers and showed that observing both the presence of unusual AS in the AS-PATH and the volume of the message leads to a greater proportion of false positive than when observing the volume only. Ganiz *et al.* [25] managed to distinguish unwillingly caused route change from link failure from BGP updates obtained from RouteViews and using the Student's statistical test.

— Techniques using Historical BGP data: As the name suggests it, this technique uses formerly gathered BGP updates in order to validate newly gathered ones under the assumption that the network topology remains unchanged for rather long periods of time. Lad *et al.* [45] designed PHAS (Prefix Hijack Alert System), a protocol analyzing BGP updates on the fly to detect prefix hijackings and alert the targeted prefix of the malfunction. PHAS requires a registration from the ASes which induces an issue for proving the legitimacy of the statements of ownership for a given prefix. Heaberlen *et al.* presented NetReview, a protocol using messages obtained from neighboring ASes to detect link failure, misconfiguration, policy violation, and attacks. Unfortunately, NetReview demands each BGP router to store a history log file for a year of data that can be used by ASes to detect anomalies.

— Techniques using Reachability Checks: This last techniques uses data plane information gathered from typical reachability monitoring tools like ping, traceroute, nmap to deduce wrongful configurations. Zheng *et al.* [78] chose to use the change in route length as a triggering event for raising an alarm, indicating a possible hijack. Tahara *et al.* [70] used the ping tool originated from different locations to observe reachability inconsistencies, allowing to rapidly spot an anomaly but demanding a cohesion between multiple vantage points and not being able to locate the source of the anomaly.

Overall, these multiple approaches for detecting abnormal behaviors generally suffer from the need to gather large amounts of former data or constant probing.

1.4 Relay Attacks over Other Environments

This section steps outside the Internet paradigm to analyze how relay attacks can take place in a different environment: the contactless communications, and observe how such attacks are addressed to see if similarities between both worlds exist.

1.4.1 Radio Frequency IDentification

Contactless Communications use radio frequencies to send data from one device to another. A Radio Frequency IDentification (RFID) protocol usually designates a 2-party protocol where a first entity, called the prover (denoted $\mathcal{P}$) exchanges information with the second entity, called the verifier (denoted $\mathcal{V}$) through a contactless channel. By the end of the protocol, $\mathcal{V}$ should be convinced that a given claim attesting of $\mathcal{P}$'s identity is true.
An RFID protocol often revolves around the concept of challenge-response, where $\mathcal{V}$ sends a question that only the possessor of $\mathcal{P}$'s credentials could answer. A typical example would be for $\mathcal{V}$ to send a message (or challenge) m so that $\mathcal{P}$ can compute a digital signature with its secret key. However, such methods are not suited for every situation. Typically, in many practical scenarios, the proving device $\mathcal{P}$ is a smart card with very limited computational capabilities, making the computation of signatures a fastidious task.

Fiat-Shamir Zero-Knowledge Protocol

In 1985, Goldwasser, Micali, and Rackoff introduced the concept of Zero-Knowledge Proofs of Knowledge (ZKPoK) [28].

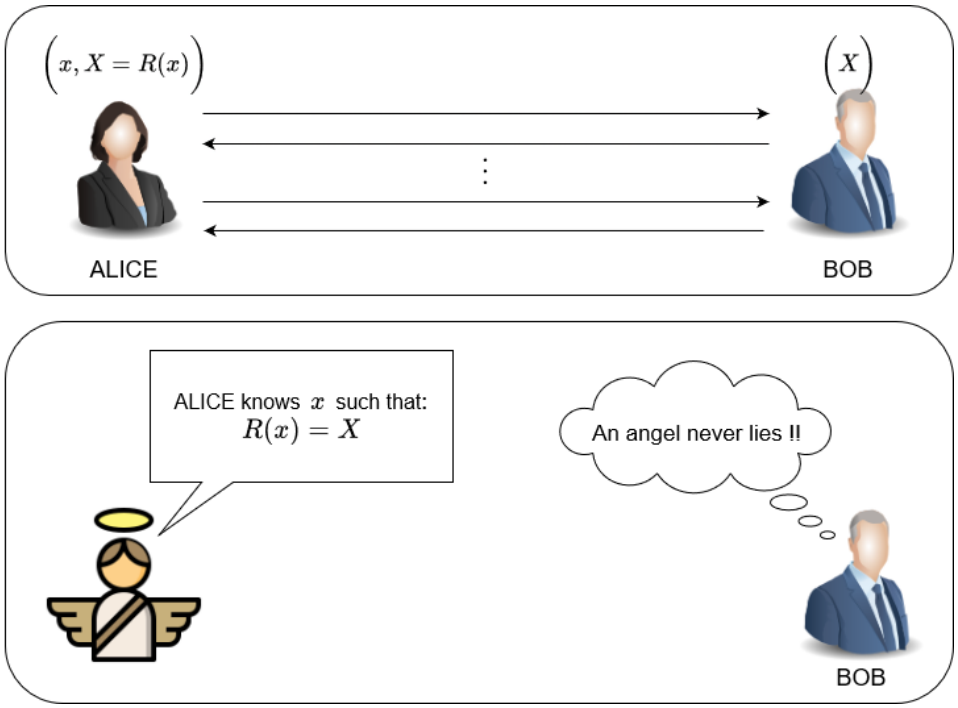


Figure 1.9 – Two scenario giving to Bob the exact same piece of information:
Top: ALICE and BOB running a ZKPoK.
Bottom: A trusted entity telling BOB a statement

A Proof of Knowledge is an interactive 2-party protocol in which the prover convinces

the verifier that he knows a value x satisfying a relation $X = R(x)$. Such a value is called a witness of X for the relation R. A Proof of Knowledge is qualified to be "Zero-Knowledge" if the interaction yields nothing but the validity of the statement S: "The prover knows a witness of X for the relation R". Therefore, a Zero-knowledge proof of knowledge interaction between a prover and a verifier should be equivalent to an absolutely trusted entity informing the verifier that the prover knows a valid witness of X for R, see Figure 1.9. A ZKPoK is caracterized by the following properties:

— Completeness: An honest verifier interacting with an honest prover will be convinced of the validity of the statement S with probability 1.

— Soundness: Roughly speaking, if a prover is able to make a verifier accepting the protocol, then the prover necessarily knows a valid witness. More formally, there exists an algorithm K called a knowledge extractor such that, for any dishonest prover $\widetilde{\mathcal{P}}$ (i.e. not knowing any witness) capable of convincing $\mathcal{V}$, K can interact with $\widetilde{\mathcal{P}}$ to output a witness x' for X of the relation R.

— Zero-Knowledge: Roughly speaking, no information can be learned from the observation of an exchange. More formally, there exists an algorithm S called a simulator, outputting a transcript t of an interaction between a prover and a verifier that would be accepted by the verifier. If S can generate a valid interaction with no access to a witness for the relation R, then necessarily, the observation of a valid interaction yields no more information about the witness.

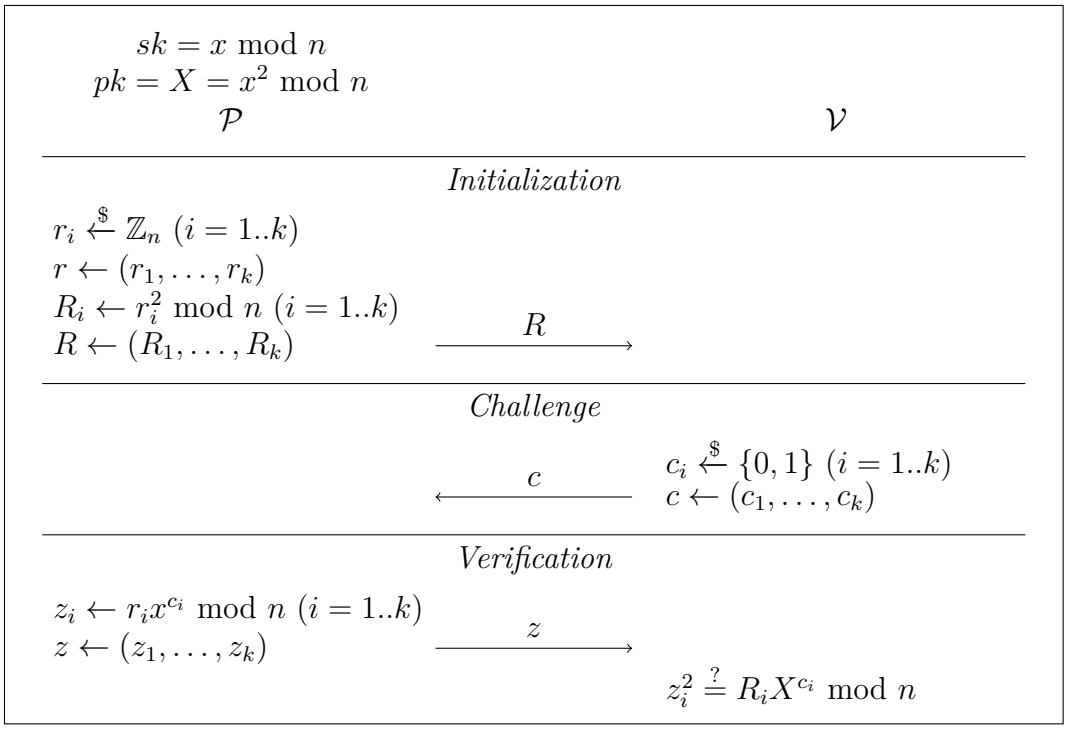


Figure 1.10 – Fiat-Shamir Zero-Knowledge Protocol

In 1986, Fiat and Shamir came up with the first functional protocol [24], known today as the Fiat-Shamir Protocol, displayed on Figure 1.10.
In this protocol, $\mathcal{P}$ proves its knowledge of the secret key x to $\mathcal{V}$, where x is a quadratic residue (a.k.a. square root) of the public key X in the set $\mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z}$ with n being the product of 2 large primes. The security of the Fiat-Shamir protocol is based on the quadratic residuosity assumption [14], stating that, given $X \in \mathbb{Z}_n$, there is no efficient algorithm capable of computing x such that $x^2 = X \mod n$.
The protocol starts with the prover generating k random nonces $r_i \in \mathbb{Z}_n$, computing their squares R_i modulo n, and sending the vector $(R_1, \ldots, R_k)$ to the verifier. This first phase is called the Initialization or Commitment phase because $\mathcal{P}$ has commited himself on the nonces r_i. $\mathcal{V}$ cannot retrieve any r_i according to the quadratic residuosity assumption and $\mathcal{P}$ cannot change any r_i to another value $\widetilde{r}_i$ because $\widetilde{r}_i^2$ would not match the corresponding R_i. Then, the verifier sends a challenge formed of k randomly chosen bits $c = (c_1, \ldots, c_k)$. The prover computes $z_i = r_i x^{c_i}$ for each i and sends the vector $(z_1, \ldots, z_k)$ to $\mathcal{V}$. If the equality $z_i^2 = R_i X^{c_i}$ holds for every single i, $\mathcal{V}$ is convinced that the statement "$\mathcal{P}$ knows a witness x of X for the relation $X = x^2 \mod n$".

1.4.2 Distance-Bounding Protocols

Mafia-Fraud Attack

In 1987, Desmedt, Goutier, and Bengio published an article highlighting several attacks on the Fiat-Shamir Protocol [22], one of which entitled "mafia-fraud".

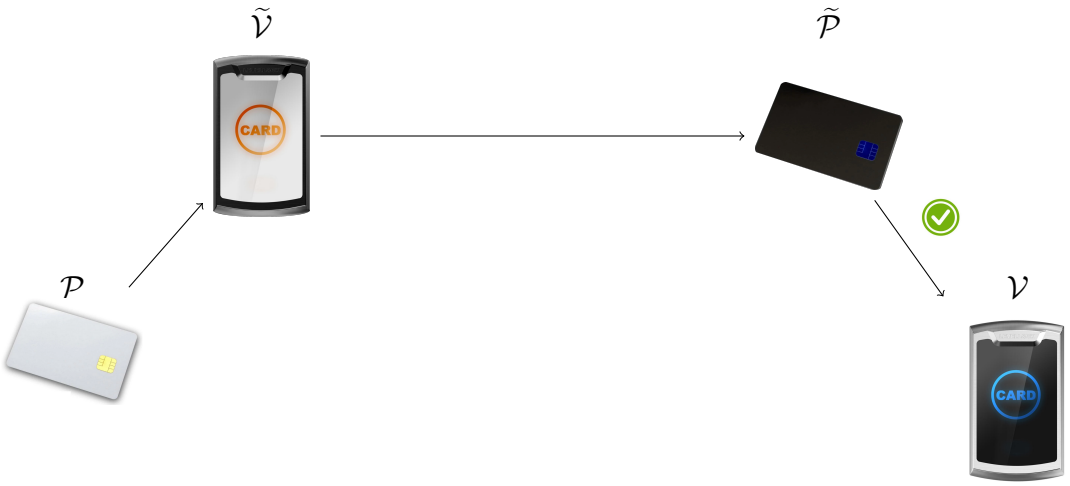


Figure 1.11 – Mafia-fraud illustration

The attack name comes from Shamir's claim that Fiat-Shamir protocol remains secure even in a scenario where the prover is a mafia-owned store, which is contradicted by [22]. The mafia-fraud actually allows the attacker to get authenticated by simply relaying the

exchange between the genuine prover and the verifying device. Such an attack especially makes sense in contactless authentication that needs the prover (card, transit pass, or else) to be in the proximity of the verifying device.
Assume a proving device $\mathcal{P}$ to be a smart card possessing secret credentials, and a verifying device $\mathcal{V}$ to be a reader placed on a secured door guarding a restricted area of an industry. Whenever $\mathcal{P}$ is placed near $\mathcal{V}$, the devices start the execution of the Fiat Shamir protocol for proving that $\mathcal{P}$ knows the secret credentials. If the interaction is accepted by $\mathcal{V}$, the door gets unlocked. Now assume that an unauthorized member of the industry owns a fake verifying device $\widetilde{\mathcal{V}}$ and his own smart card $\widetilde{\mathcal{P}}$, and that those can communicate through a long range private channel. Then, by placing $\widetilde{\mathcal{V}}$ near to $\mathcal{P}$ and $\widetilde{\mathcal{P}}$ near to $\mathcal{V}$, and by simply relaying every message, the unauthorized member walks through the door in perfect impunity. Figure 1.11 illustrates this toy example situation.

Brands and Chaum

The main countermeasure to the mafia-fraud relay attack is the family of so-called "distance-bounding" protocols, a.k.a. proximity checks. They have been massively studied [11] in the context of Radio Frequency IDentification (RFID), and are already implemented in some contactless smartcards, e.g., Mifare Plus [50] and Mifare DESfire [49].
The concept of distance-bounding is introduced by Brands and Chaum in 1994 [16]. Their idea is to enhance the Fiat-Shamir protocol by measuring communication time between the real prover and verifier, allowing to bound the distance from which is standing the genuine prover and to dismiss the authentication if it concludes that the prover is standing further than a given distance. To achieve this, the protocol of Brands and Chaum uses a series of rapid bit-exchanges to measure the round trip time between the prover and the verifier. Given that the signal propagation cannot be faster than the speed of light, a verifier considers that there is no relaying adversary if the round trip times between the verifier and the prover are below a given upper bound. Brands and Chaum protocol is depicted in Figure 1.12.
In this protocol, $\mathcal{P}$ proves to $\mathcal{V}$ that he knows x such that $x^2 = X \bmod n$ in the exact same fashion as in the classical Fiat-Shamir protocol, but here, the challenges c_i are collaboratively computed by the XOR of bits $c_i^{\mathcal{P}}$ drawn by $\mathcal{P}$ and $c_i^{\mathcal{V}}$ drawn by $\mathcal{V}$. The challenge phase is renamed "Fast bit-exchange" as its main purpose is now to compute accurate round trip times between the 2 parties:

- *Initialization*: $\mathcal{P}$ picks k nonces r_i, computes their squares $R_i = r_i^2 \mod n$, then picks k random bits $c_i^{\mathcal{P}}$. He then sends the R_i's and a commitment of the $c_i^{\mathcal{P}}$s. The commitment function can simply be seen as a hashing function in this specific case.
- *Fast bit-exchange*: The verifier $\mathcal{V}$ also computes k random bits $c_i^{\mathcal{V}}$. Then, for i from 1 to k, $\mathcal{V}$ creates a timestamp t_i, sends $c_i^{\mathcal{V}}$, receives the responses $c_i^{\mathcal{P}}$, immediatly creates another timestamp t_i', and stores $(t_i' - t_i)$.
- *Verification*: $\mathcal{P}$ computes $c_i = c_i^{\mathcal{P}} \oplus c_i^{\mathcal{V}}$, and $z_i = r_i x^{c_i} \bmod n$ for all i in $\{1, \ldots, k\}$, and sends $z = (z_1, \ldots, z_k)$ to $\mathcal{V}$. The latter checks (i) if the committed $c_i^{\mathcal{P}} s$ in the initialization phase are the same as those he received in the fast bit-exchange phase

(typically by recomputing the $\mathsf{commit}(c^{\mathcal{P}})$), (ii) computes the c_i by XORing just like $\mathcal{P}$ did, (iii) checks if z_i^2 is equal to $R_i X^{c_i}$ (this is the classical Fiat-Shamir check for the knowledge of x) and (iv) checks if $\max(\{t'_i - t_i\})$ is below a given upper bound.

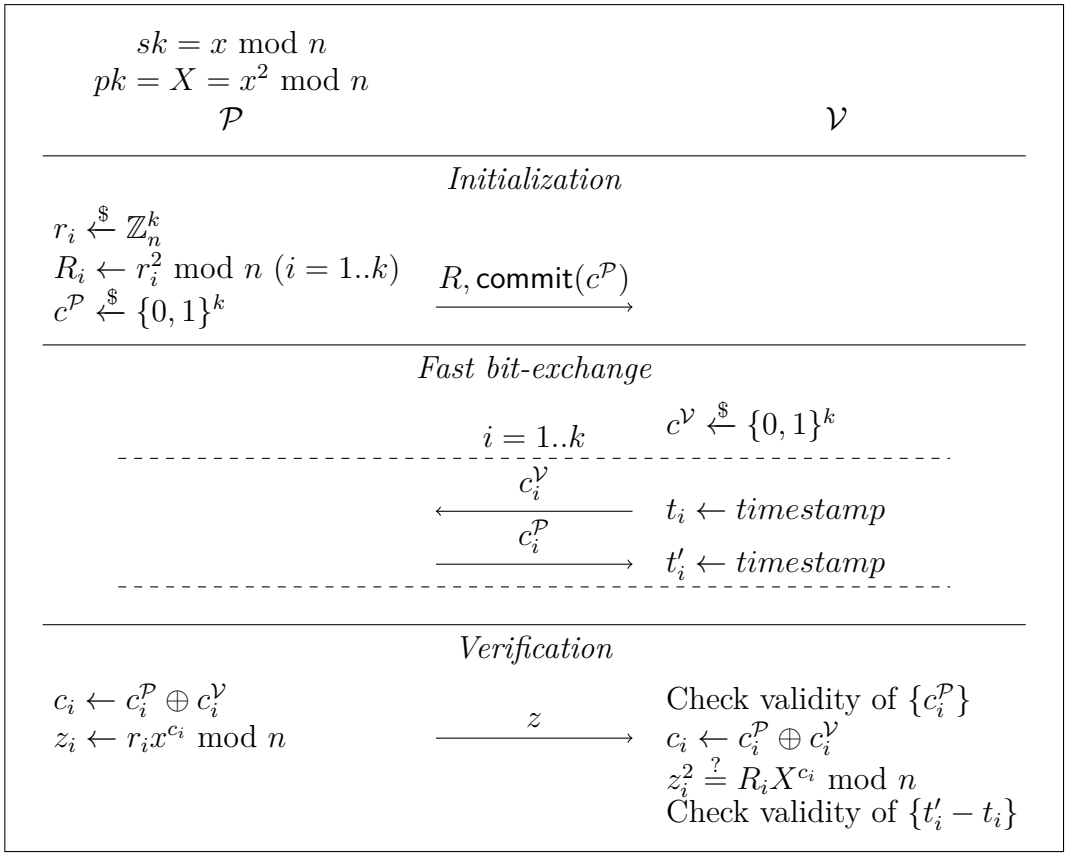


Figure 1.12 – Brands and Chaum's distance-bounding protocol based on the Fiat-Shamir zero-knowledge authentication

Attacking strategies

Despite no clear security proofs, Brands and Chaum argued on a possible attacking strategy. Assuming that the technology possessed by the attacker does not allow him to naturally trick the time check (which is a weak assumption considering that the relativity of the speed of light has been extensively used in countless studies), an attacker can still try to cheat by anticipating the challenges $c_i^{\mathcal{V}}$ sent by the verifier.

To this end, the attacker pre-selects k bits $\widetilde{c_i^{\mathcal{V}}}$ and plays the protocol with $\mathcal{P}$ from the attacking device $\widetilde{\mathcal{V}}$. From that execution, the attacker receives the vector R, the k bits $c_i^{\mathcal{P}}$ chosen by the prover, and a vector:

$$z = (r_1 \cdot x^{c_1^{\mathcal{P}} \oplus \widetilde{c_1^{\mathcal{V}}}}, \ldots, r_k \cdot x^{c_k^{\mathcal{P}} \oplus \widetilde{c_k^{\mathcal{V}}}})$$

Then, using this information, the attacker now plays the protocol with the genuine verifier $\mathcal{V}$ from the attacking device $\widetilde{\mathcal{P}}$ by sending him the bits $c_i^{\mathcal{P}}$ for his part of the Fast bit-exchange. By doing so, the time check is valid as the attacking device stands nearby. However, the attacker only succeeds if each one of his $\widetilde{c_i^{\mathcal{V}}}$ matches with the genuines $c_i^{\mathcal{V}}$. Otherwise, the vector z received from the previous execution does not pass the verification phase. Trivially, the attacker has a $\frac{1}{2^k}$ probability of guessing correctly every $c_i^{\mathcal{V}}$.

Other Distance Bounding Protocols

Brands and Chaum's seminal work paved the way to many other distance-bounding protocols addressing mafia-fraud attacks, as well as other variants that are out of the scope of this manuscript. One could for example cite Hancke and Kuhn's protocol [35] that uses only symmetric-key cryptography. Hancke and Kuhn's protocol allows slightly better chances on the anticipation attack presented above, with a probability of $\left(\frac{3}{4}\right)^k$ of tricking the verification phase. It trades this loss for a less computationally greedy protocol, which is better suited for limited devices. Hancke and Kuhn protocol is depicted in Figure 1.13.

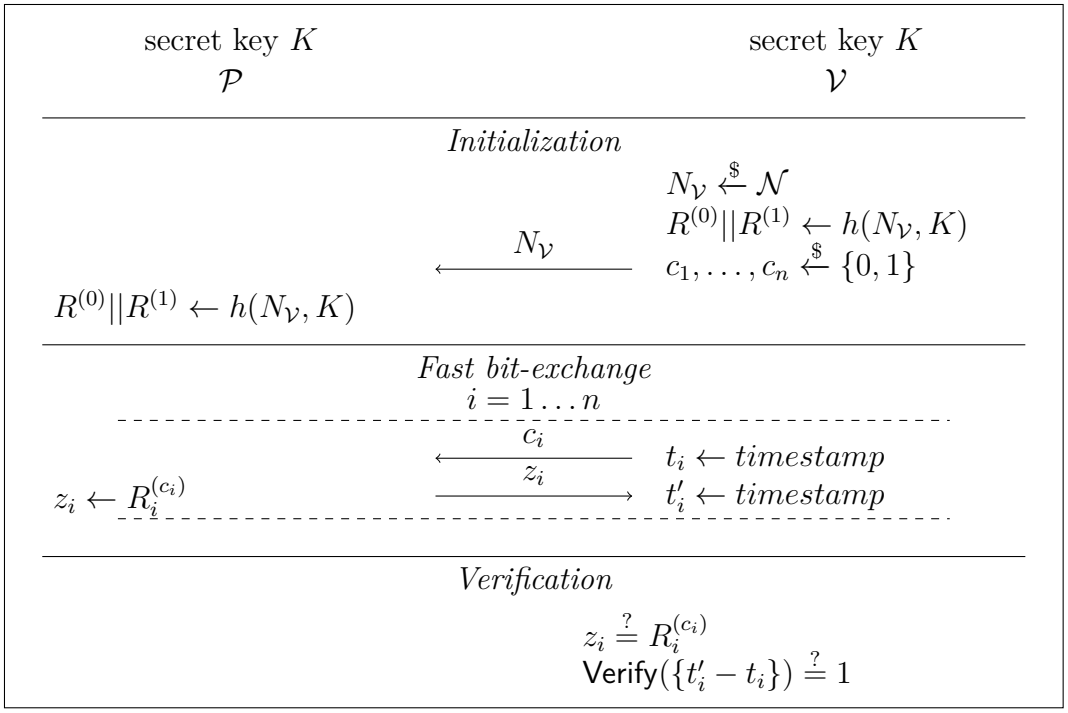


Figure 1.13 – Hancke and Kuhn's distance-bounding protocol based on symetric-key cryptography

Although describing the body of literature related to distance-bouding protocols is out of the scope of this manuscript, interested readers will find a complete analysis of distance-bounding protocols in [11]. It is worth noting that these protocols are well suited for

RFID authentication because communications are end-to-end (from the physical layer perspective) and the computations performed by the RFID tag are lightweight, which implies that the measured round trip times are very stable.

Remark

It is important to raise that distance-bounding protocols does actually not detect relays: they detect abnormally long communication times, and conclude that there is a risk of relay attack. To the contrary, an execution of a distance bounding protocol validating the authentication does not ensure the absence of a relay attack. Typically, if both the attacking devices involved in the mafia-fraud as well as the real prover, are standing near the verifying device, the communications time can probably remain low enough to be valid in the presence of the relay.

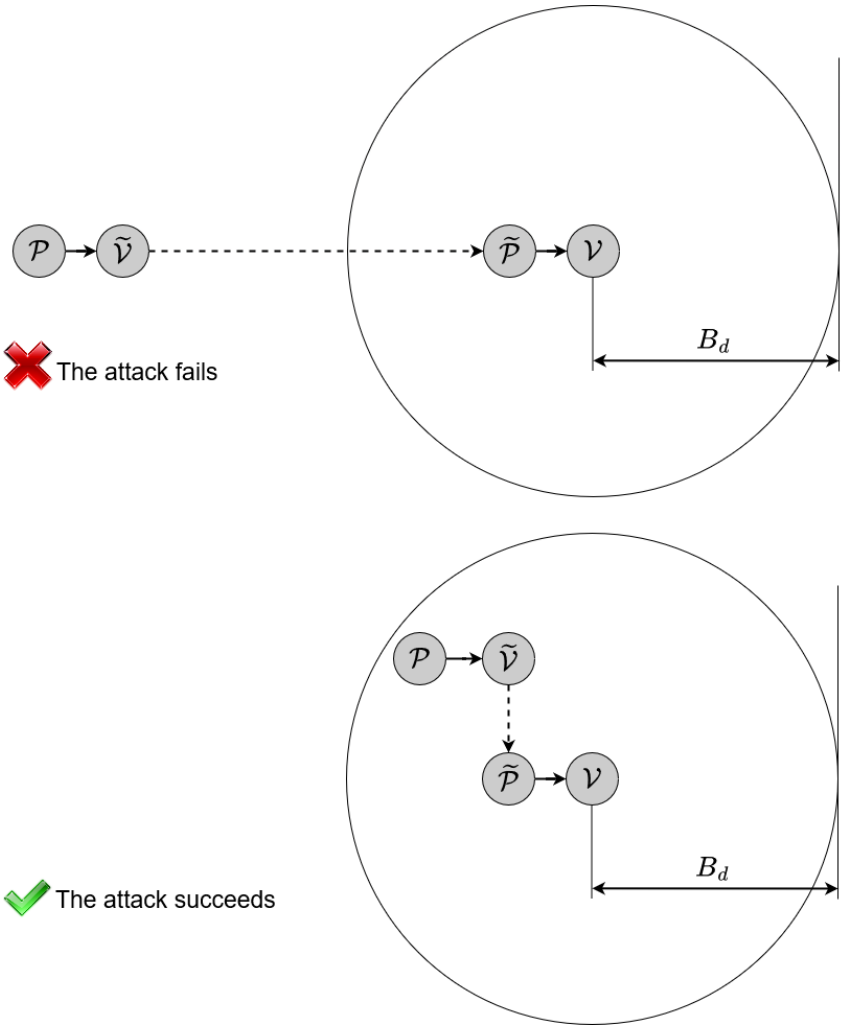


Figure 1.14 – A mafia-fraud attack failing or succeeding, depending on the position of the attacking and proving devices.

More precisely, when choosing an upper time bound B_t for the time check, the user is implicitly choosing a distance bound B_d which is the distance traveled by light during B_t units of time. This bound B_d defines a perimeter around the verifier $\mathcal{V}$. This delimited zone prevents an attacker to perform a successful relay if the prover $\mathcal{P}$ is standing outside of it, because then, the measured times would be too high. To the contrary, if $\mathcal{P}$ stands in the zone, and neglecting the processing time of the attacking devices, the relay does not create a sufficient impact on time to be detected, see Figure 1.14.
The risk of this scenario to happen can be controlled by lowering the bound B_d but can never be completely avoided.

Conclusion

This chapter highlighted the flaws of Internet routing procedures by providing a high-level overview of the current architecture. These flaws allow misleading route announcement that can ultimately create undesired and long-term relays. A successful relay attack on the Internet can turn out to be very damaging if the targeted traffic flows contains critical data, this has been the case during an incident in 2010, rerouting traffic from US governmental institutions and important industries.
To address these flaws, multiple promising countermeasures are working on mitigation techniques based on cryptography but remain hard to adopt due to the necessary hardware update that they imply, other proposals are aiming at the construction of a clean slate redesign of the routing architecture, then again, these solutions will take a long time to convince the scientific community, financial investors, and the general public.
Because sometimes, the grass is actually greener on the other side of the fence, this chapter also took interest in the mitigation of relay attack on a different kind of channel: the contactless communications. A very dense literature exists on this matter, since the distance-bounding protocol introduced by Brands and Chaum in 1994. These authentication protocols use the measurement of communication time separating a prover and a verifier to check the geographical distance separating them, and refuting the authentication whenever this distance is larger than a pre-selected bound. Distance-bounding protocols are already implemented in several devices and benefit from the fact that real life situations are easily reproducible in experimental environments.
This chapter concludes itself on the main question that was asked during these 3 years of research:

> Can the idea of Distance-Bounding protocols be transferred to the Internet for relay detection ?

Before diving into the core of this question, let's take a closer look at the differences between the 2 paradigms. In distance-bounding protocols, there is only 2 devices involved in the communication. This simple fact holds the first and maybe the more critical difference with the Internet, where messages travels from hop to hop until the destination is reached. Moreover, a route from source to destination is subject to change as the topology of the Internet is constantly evolving through the path-vector routing process. This implies that communication times over the Internet are trivially expected to be a lot more variable than the very accurate radio frequency measurements and, consequently, that a straight translation of distance-bounding method is doomed to fail. Computing a geographical distance separating two Internet nodes is not only compromise by the time scales involved, but it would also be devoid of sense. There is no reason to believe that a linear property like $distance = time \times speed$ would hold, even approximately. Finally, relay attacks over the Internet can target specifically the content of exchanges between entities, whereas distance-bounding protocols prevents a kind of impersonation of the real

prover. This means that a relay detection process must operate during a communication and not only a handshake.
Several questions arise from those observations:

- How stable can time measurement get over the Internet ?
- How can the computation of distance from time measurement be replaced by a method that efficiently detect a relay and is suited to this environment ?
- Can such a protocol take into account the messages of an exchange, regardless of the size of the transmitted data ?

Having access to a 2-party distance-bounding like protocol detecting relay attacks over the Internet provides 2 main advantages. Such a protocol would be (i) scalable, as it would not require any software or hardware update for intermediate nodes, and (ii) routing protocol independent, as it only involves the end-points, that protocol is not impacted by the method for constructing the routes, as long as the times suits the process deciding to declare a probable relay. The rest of this manuscript will answer all these questions to fully describe such a protocol, from the basics experiments on time over the Internet up to a cryptographic proof of security in the Random Oracle Model.

Chapter 2

Experiments on Time over the Internet

"Let me give you an advice.
Never forget what you are, for surely the world will not.
Make it your strength. Then it can never be your weakness.
Armor yourself in it, and it will never be used to hurt you."
- Tyrion Lannister

Introduction

In order to design a protocol taking advantage of time measurement, very strong insurances and knowledge about the behavior of the time between two nodes are needed. To gain such knowledge, we did set up multiple nodes in Western Europe and one node in the United States of America and made them communicate to observe the evolution of the corresponding time measurements over short, medium, and long periods.
This chapter elaborates on these observations made of about 5000 experiments spanned over the years 2020 and 2021. Section 2.1 present important notation and methodology choices for the experiments, and Section 2.2 presents the observation made from them. Section 2.2 is divided into 4 important parts. The Section 2.2.1 presents the spreadness for samples collected over a short period, while the Section 2.2.2 provides a comparison of samples gathered over longer periods. The Section 2.2.3 looks out for another factor presumably impactful: the packet length, and the Section 2.2.4 elaborates on the impact of a simulated relay.

2.1 Methodology

In the sequel of this manuscript, a set of collected measurements is referred to as a "sample" and is represented on a 2-dimensional graph with time on the y-axis and using one value on the x-axis per measurement. A sample between two nodes $\mathcal{S}$ and $\mathcal{R}$ will be denoted ($\mathcal{S}$,$\mathcal{R}$) when no further precision are needed.

Before going into the heart of our observations, the following highlights a few worthy methodology choices.

2.1.1 Measurement method

The transit time between two nodes can be expressed in two main ways:

— The *One Way Transit Time* ($OWTT$) represents the time measured between the sending of a packet by $\mathcal{S}$ and the reception of that packet by $\mathcal{R}$.

— The *Round Trip Time* (RTT) is the time measured between the sending of a packet by $\mathcal{S}$ and the reception of an acknowledgement sent by $\mathcal{R}$.

$OWTT$ attempts to capture the real time separating two endpoints but demands a precise clock synchronization of those nodes and to send the timestamp along with the packet.

To the contrary, RTT is a one-sided measurement, and so, offers the possibility to obtain precise measurements up to the nanosecond without having to handle any clock synchronization.

The approximation $OWTT = \frac{RTT}{2}$ is sometimes made, however there is no insurance that the transit times in both directions are comparable. Also, RTT might include some processing time. It is then preferable to consider RTT as a stand-alone metric rather than a way to measure $OWTT$. All the following measurements are computed using the RTT metric for a matter of accuracy and consistency with the methods of measurements involved in Distance-Bounding (which also are RTT). Figure 2.1 illustrates the computation of a single RTT between two nodes $\mathcal{S}$ and $\mathcal{R}$.

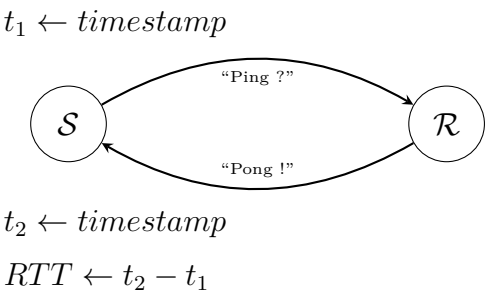


Figure 2.1 – Illustration of a RTT measurement between $\mathcal{S}$ and $\mathcal{R}$ performed by $\mathcal{S}$

2.1.2 Transport Protocol

All the experiments presented in this manuscript were performed using the *User Datagram Protocol* (UDP) packets (see RFC768 [56]). UDP acts on the transport layer of OSI model. UDP is an Internet standard protocol designed to communicate with minimalistic workload for the endpoints. No confirmation of delivery, integrity or authenticity checks is taken care of, the packet only contains the Source and Destination ports and is encapsulated in an IP packet containing the Destination IP address. Using UDP packets for our experiments avoids undesired actions jeopardizing the measurements, like the classical acknowledgements inherent to the *Transmission Control Protocol* (TCP). Figure 2.2 displays the content of a UDP header.

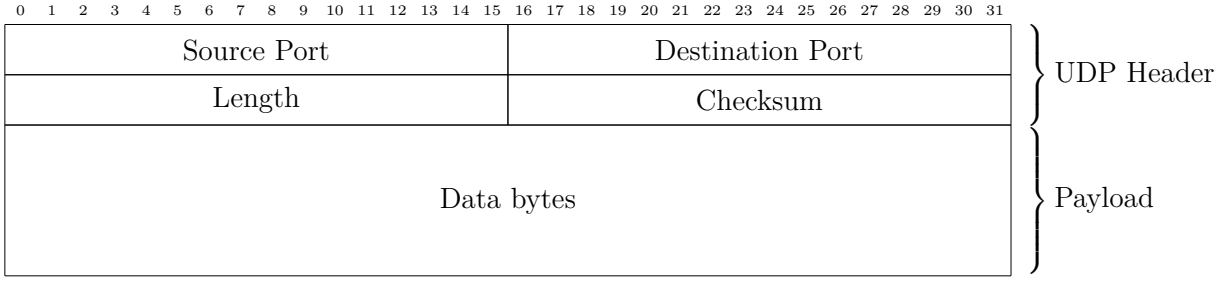


Figure 2.2 – UDP packet

2.1.3 Appreciation of RTT behavior

For a protocol based on time analysis for relay detection to be practical, 3 statements must be confirmed. The first one is obviously the impactfulness of a relay attack, which will be discussed in Section 2.2.2. The second and third ones are the stability and consistency of samples gathered between a given couple of nodes.
The notion of stability of a sample refers to how dense its values are. Intuitively, it represents the non-chaotic behavior of RTT s over a few measurements, the more stable a sample is, the easier it should be to detect an impactful event. Stability can be expressed, for instance, using variance. The variance of a sample expresses how far in average its values are from its mean:

Definition 2.1.1. ***Variance***
Let $S = (s_1, \ldots, s_n) \in \mathbb{R}_+^n$ be a sample and μ its mean, then:

$$var(S) = \frac{1}{n}\sum_{i=1}^{n}(s_i - \mu)^2$$

However, variance is strongly impacted by extreme values (or outliers), which are frequent when measuring RTTs over the Internet. A simple tool neglecting strong outliers can be defined by the following:

Definition 2.1.2. ***Smallest Representative Interval (SRI)***
Let $S = (s_1, \ldots, s_n) \in \mathbb{R}_+^n$ be a sample of size n such that:

$$s_1 \leq s_2 \leq \cdots \leq s_n$$

Let $p \in]0, 1]$.
The Smallest p-Respresentative Interval of S, denoted $I_p(S)$, is the smallest interval containing at least a proportion p of the values of S.

Definition 2.1.3. ***Representative Size (RS)***
Let $S = (s_1, \ldots, s_n) \in \mathbb{R}_+^n$ be a sample of size n such that:

$$s_1 \leq s_2 \leq \cdots \leq s_n$$

Let $I_p(S)$ be the SRI of S for some proportion p.
Let $s_i, s_j \in S$ be such that $I_p(S) = [s_i, s_j]$.
The size $(s_j - s_i)$ of the Smallest p-Representative Interval of S is called the p-Representative Size of S, and is denoted $|I_p(S)|$.

The smallest p-representative interval of a sample allows to express where the specified proportion of the values lives while the p-representative size measures the spreadness of the representative values. The representative size of a sample is not impacted by extreme outliers.
The remaining of this Chapter will systematically use the SRI and RS of samples with a proportion:

$$p = 0.9$$

The notion of consistency expresses the long term continuity of the stability, and is evaluated by observing the evolution of the mean, variance, and SRI over time.

2.1.4 Experimental Setup

To ensure a satisfying experimental diversity, the experiments were performed over a total of 7 nodes across 4 countries. This allows to observe the impact (if any) of the distance separating the nodes with 4 nodes located in France, 1 in Germany, 1 in Poland, to experiment on small to medium distances, and 1 in the USA (Oregon) for larger ones.
The notion of geographical distance is actually of second importance here, because the most probable impacting factors in the travel time of a packet are the number of visited routing equipment and their respective processing speed. Indeed, 2 geographically close nodes belonging to 2 different AS will most probably be further apart time-wise than 2

nodes in the same AS separated from a larger distance. In the following, if no further detail is given, the term "distance" will refer to the number of routing equipments separating the nodes.

2.1.5 Terminology

The samples discussed in this Chapter belong in one of the three following categories:

— Punctual: a punctual sample is formed of the result of a "burst" measurement. The next packet is sent right after the computation of the previous RTT. This kind of sample allows to observe time-behavior over very short periods of time.

— Continuous: a continuous sample is formed of measurements gathered at constant pace and over longer periods of time (typically, several months). This kind of sample allows to observe the general behavior of time without missing some potential intermediate events.

— Spreaded: a spreaded sample is a collection of punctual sample between the same 2 nodes separated from even longer periods of time. This kind of sample allows to observe possible long term changes.

2.2 Observations

2.2.1 Stability For Punctual Samples

A glance of the result of short period experiments is displayed in Figures 2.3, 2.4, and 2.5. Figure 2.3 shows 6 graphs, in which each "+" represents the value of one RTT in milliseconds (readable in the y-axis). Each graph is a plot of punctual samples formed of 7000 RTT between two end-points collected in a row. The dates and times of the start and end of the measurements are given on each individual graph.
Note that the scales have been uniformized on this representation to observe the variations depending on the source and destination nodes. As it should be expected, the times involved in the communications going from Europe to the USA (right-hand side of Figure 2.3) are higher than the ones involved in intra-Europe communications (on the left-hand side).

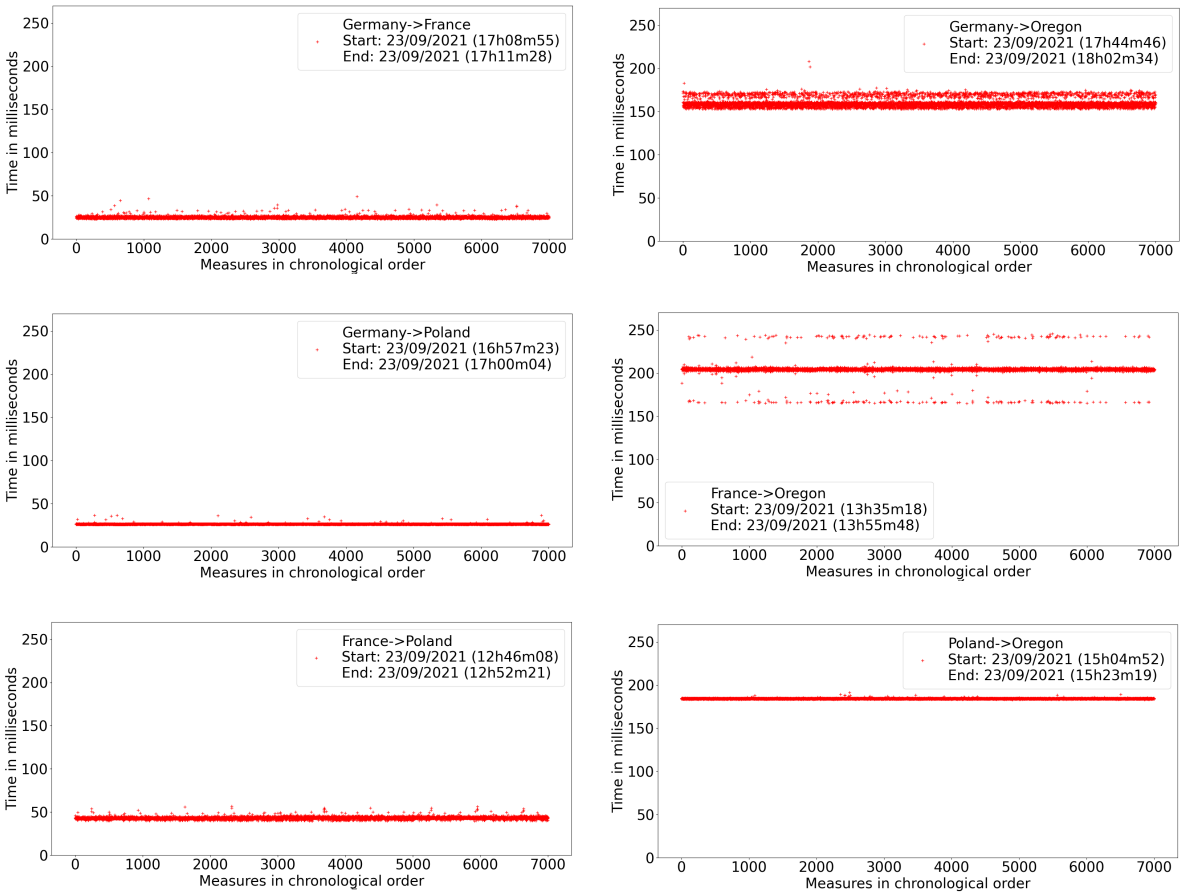


Figure 2.3 – RTTs for 7000 512-bytes packets between various locations

The samples (France,Oregon), (Germany,Oregon), and (Poland,Oregon) displayed in Figure 2.4 respectively live around 200, 159, and 185 milliseconds while (Germany,Poland), (Germany,France), and (France,Poland) all stand bellow 50ms (see Figure 2.5). Figures 2.4 and 2.5 display the same samples on a more zoomed in scale along with their statistic distribution to get a more precise look at the overall shapes of the samples.

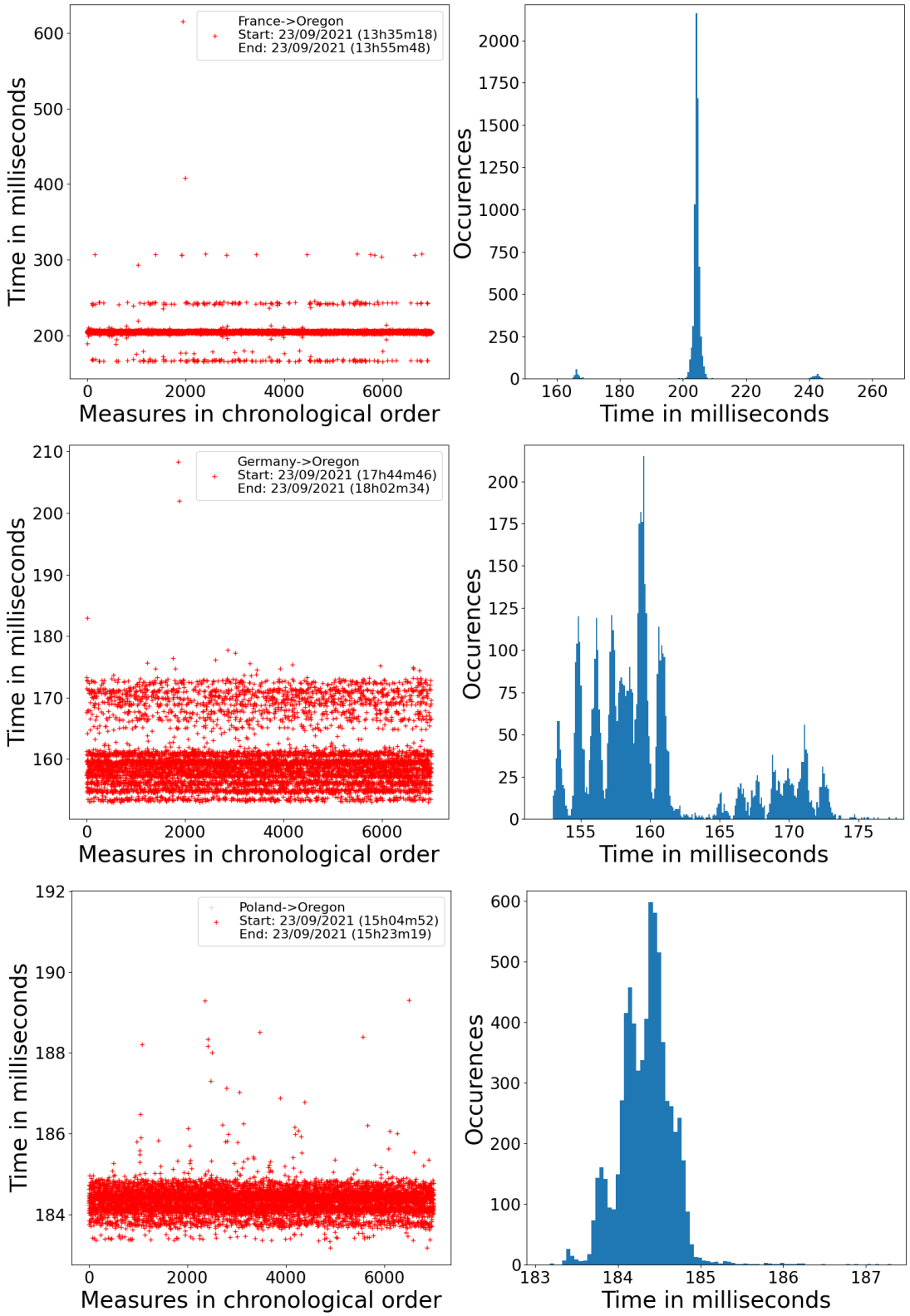


Figure 2.4 – Focus on RTTs from Europe to the USA with their distributions

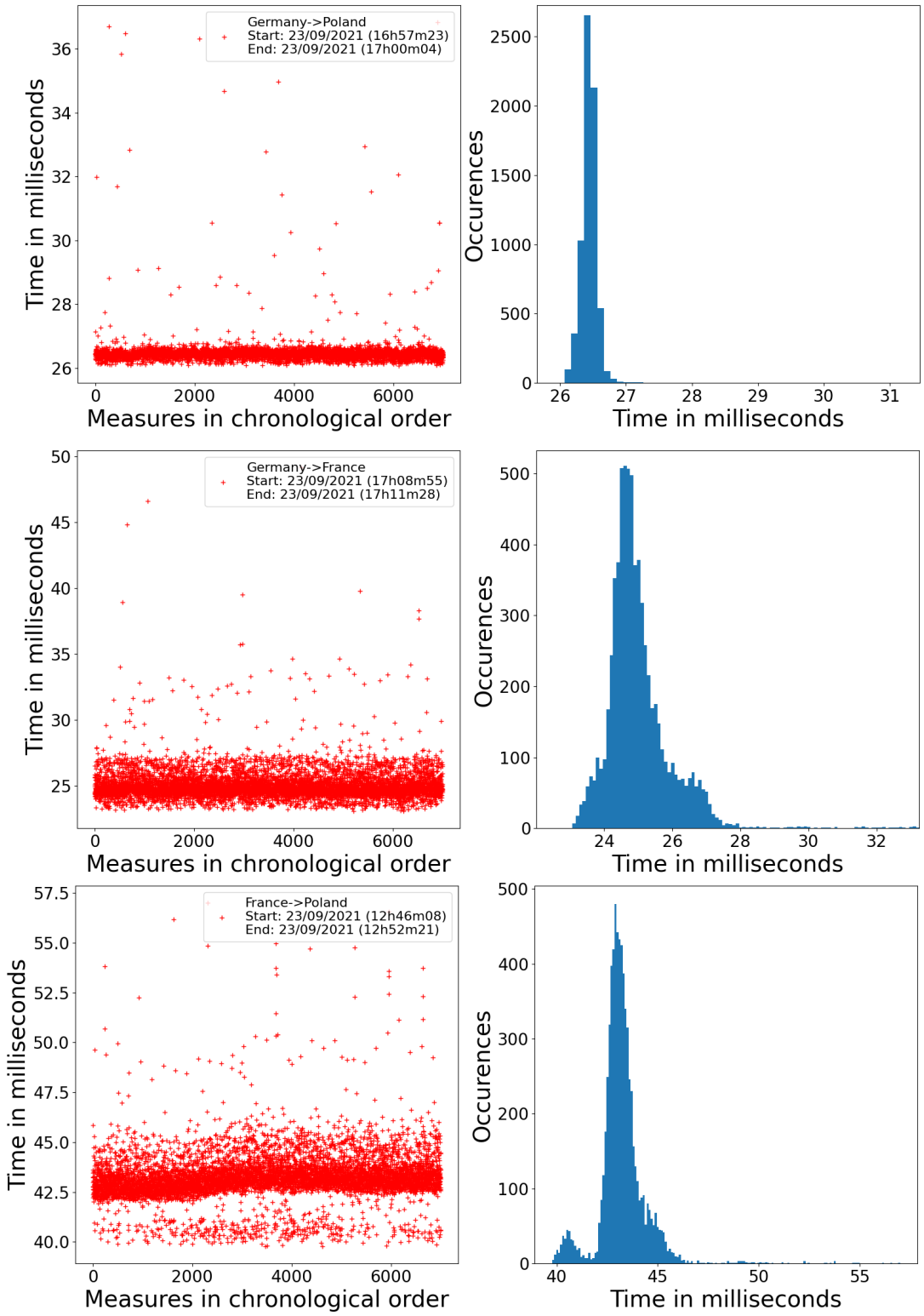


Figure 2.5 – Focus on RTTs from Europe to Europe with their distributions

Regarding stability, Table 2.1 explicits the principal statistical moments (mean, median, and variance), and the SRI of each sample.

Table 2.1 – Mean, Median, Variance, and SRI of each samples

sample S	Mean	Median	$var(S)$	$\|I_{0.9}(S)\|$	$I_{0.9}(S)$
(Germany,France)	25.04	24.81	1.44	2.92	$[23.70, 26.62]$
(Germany,Poland)	26.47	26.45	0.17	0.37	$[26.24, 26.61]$
(France,Poland)	43.24	43.13	1.51	3.17	$[42.12, 45.29]$
(Germany,Oregon)	160.20	159.13	26.86	16.62	$[154.58, 171, 20]$
(France,Oregon)	204.54	204.37	108.73	3.62	$[202.59, 206.21]$
(Poland,Oregon)	184.34	184.37	0.13	0.99	$[183.81, 184.80]$

Noticeably, the 4 samples (Germany,France), (Germany,Poland), (France,Poland), and (Poland,Oregon) show a satisfying stability with low variance and low representative size. The corresponding distribution graphs demonstrate similarities with a clearly defined area of expected times. However, (Germany,Oregon) stands out with a more spread out sample: a variance of 26.86 and a representative size of 16.62. This gap is graphically justified by the representation of two distinct areas of expected time, the first one between 152 and 162 milliseconds, and the other between 165 and 173 milliseconds. Another unexpected behavior is given by the sample (France,Oregon) which, despite getting a very low representative size, has the greatest variance of all the samples presented here. This is also graphically justified by three distinct areas: the first one between 202 and 206 milliseconds containing an overwhelming proportion of the sample (which explains the low representative size) and two others living around 165 milliseconds and 240 milliseconds.
The fact that those 2 previous samples show more than one area of density could be explained, for instance, by a load balancing process automatically handled by routing equipments. However, this question goes out of the scope of this manuscript, we believe that the knowledge of the representative size of the sample is enough for a relay detection mechanism to be efficient, whether or not the sample gets multiple areas of density.
Overall, this study of the stability of punctual samples has showcased 2 categories of possible behavior of RTT. The first one, samples with only one area of density, has proven to be extremely condensed in small intervals of time: below 5ms even for intercontinental communications like sample (Poland,Oregon). The second one, samples with more than one area of density, despite a more spread out result, are keeping an important proportion of their values in a well-defined interval, being at most 20 milliseconds long.

2.2.2 Stability For Continuous and Spreaded Samples

The last section highlighted the fact that samples gathered during a short period of time are stable in a range of a few milliseconds depending on the nodes involved in the communication. This section seeks to learn if this stability is consistent over longer periods. Figure 2.6 displays a continuous (Poland,Oregon) sample that has been measured

between the 21^{st} of December 2021 and the 17^{th} of January 2022. During this experiment, one RTT was computed each 3.6 seconds (that is 1000 RTTs per hour) resulting in almost 600000 measurements.

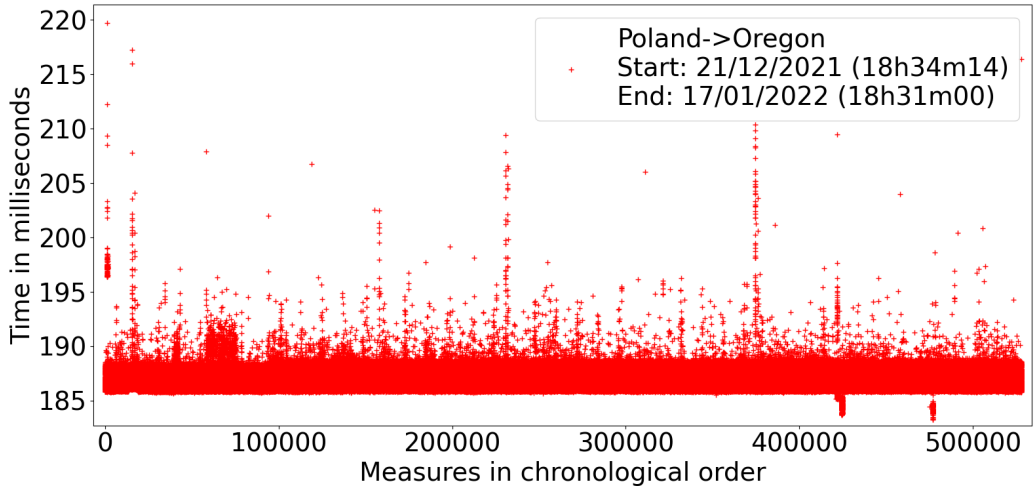


Figure 2.6 – Long term continuous experiments between Poland and Oregon

Figure 2.6 shows that the consistency is achieved over this period with a representative size of 2.08 corresponding to the interval $[185.89, 187.97]$ and a very low variance of 0.69. Although, this continuous sample differs from the punctual (Poland,Oregon) sample presented in the previous section (Figure 2.4) as its mean has increased of about 2 milliseconds. That slight RTT modification for a same couple of nodes is illustrated on Figure 2.7 for two samples gathered 2 months apart. The sample on top has been collected between the 14^{th} and the 16^{th} of October 2021 while the bottom one is a subset of the sample shown in Figure 2.6.

From these observations, it can be concluded that large periods of consistency can be achieved but slight change, probably due to occasional routing updates, might occur. Going further on this analysis, Figure 2.8 shows the means of eighteen samples collected between early September 2021 and mid-January 2022 and forming a large spreaded sample.

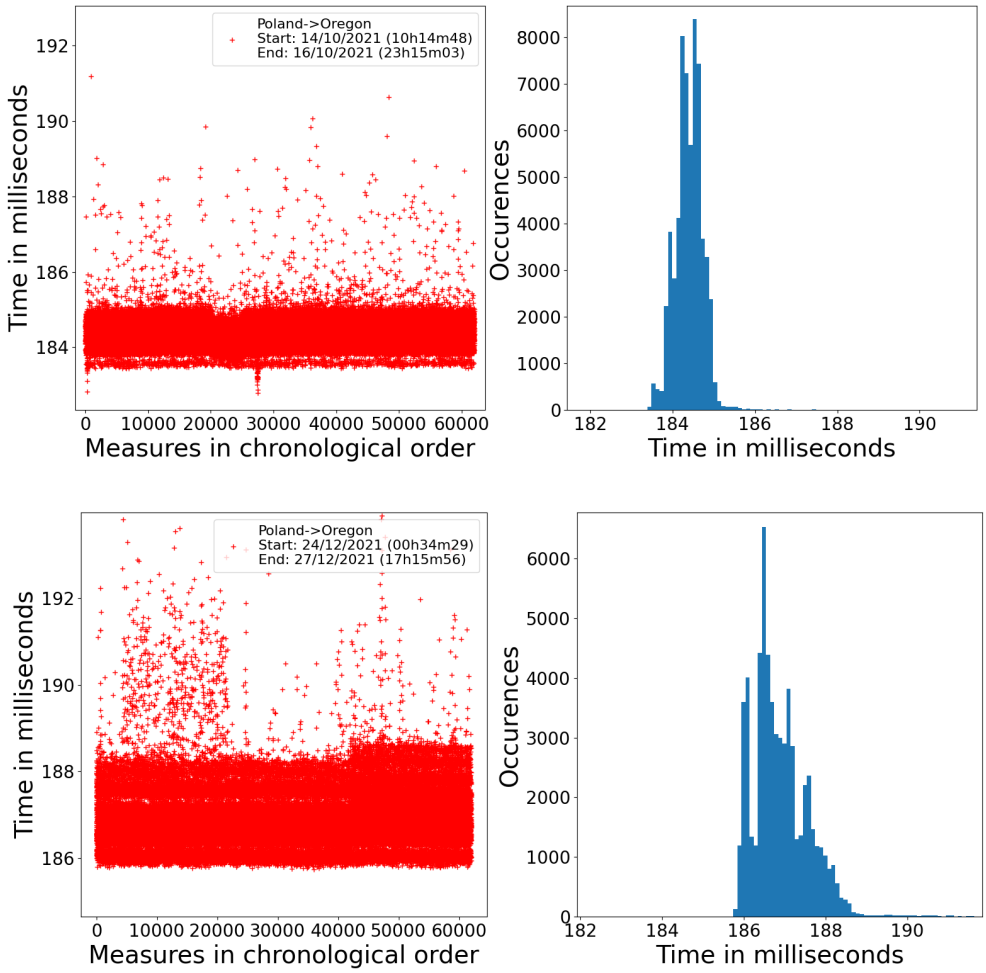


Figure 2.7 – Two samples and their distribution between Poland and Oregon 2 months apart

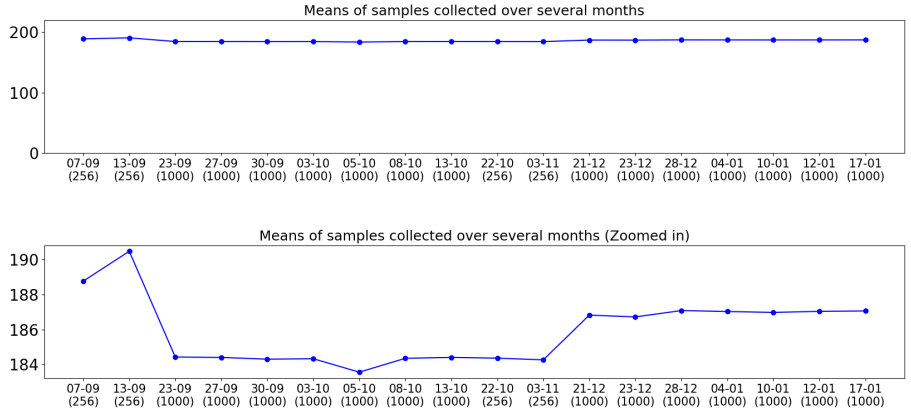


Figure 2.8 – Experiments between Poland and Oregon over 4 months

The graph on top of Figure 2.8 is showing the means in milliseconds of those samples, the days and months on which they were collected are readable in the x-axis with their respective sizes (between parenthesis). The bottom one shows the same sample on a more zoomed in scale. Figure 2.8 proves that the stability of the measurements is susceptible to evolve for the order of magnitude of the milliseconds. This same phenomena has also been noticed for samples between Germany and Oregon.

2.2.3 Impact of Packet Lenght

During all the previous experiments, UDP packets of identical sizes (512 bytes) were sent. However, a classical exchange between two nodes is frequently involving packets of various size. In order to finally conclude on the practicality of a relay detection protocol based on RTT measurement, the impact of said packet size must be conducted.
Figure 2.9 shows the RTTs in relation to the size of the sent packets, starting with a measurement using an empty packet and incrementing the byte size by one for each new packet. This shows a clear impact on the RTT, globally increasing the time by 4 milliseconds between an empty message and a 4096 bytes message. Note that such a size gap between packets is unrealistic in practice, because most services (for instance DNS) restrict the largest packet length in order to respect the Maximum Transmission Unit (MTU) on the Internet and avoid frequent packet losses.

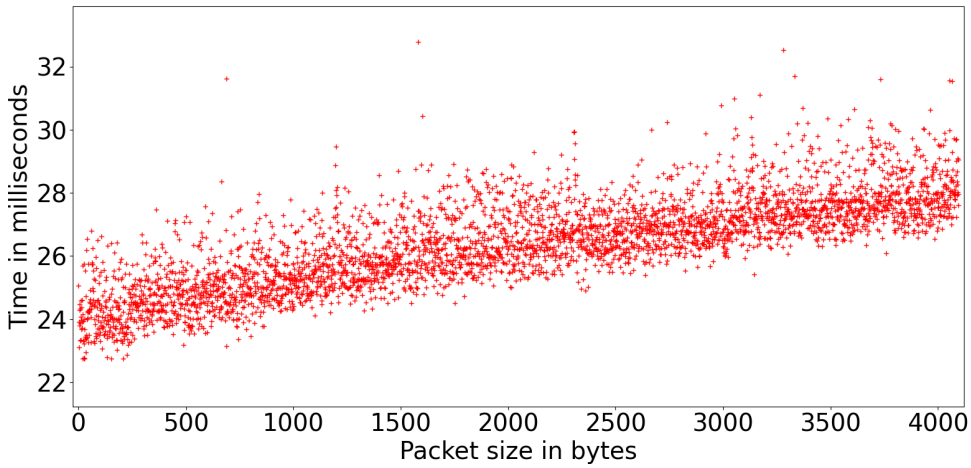


Figure 2.9 – Evolution of RTTs for packets of size 1 to 4096 bytes

Finally, Figures 2.10 and 2.11 compare two punctual samples, both gathered from an exchange between two nodes in France: Rouen and Caen, the 2^{nd} of March 2021. Both samples are represented alongside with their distributions, the top graph represents an exchange performed using packets of fixed size of 1024 bytes, while the bottom one represents an exchange of packet of random size, uniformly picked between 896 and 1152 bytes. Table 2.2 highlight their differences in terms of variance and representative size.

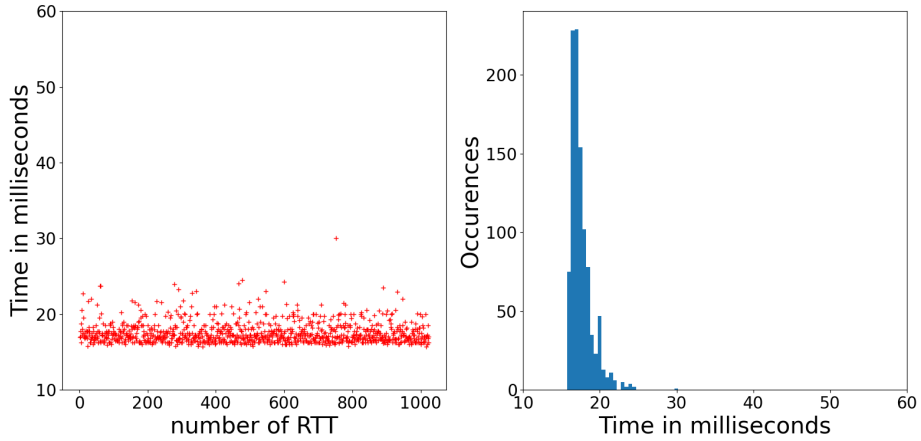


Figure 2.10 – A sample measured with constant size packets (1024 bytes)

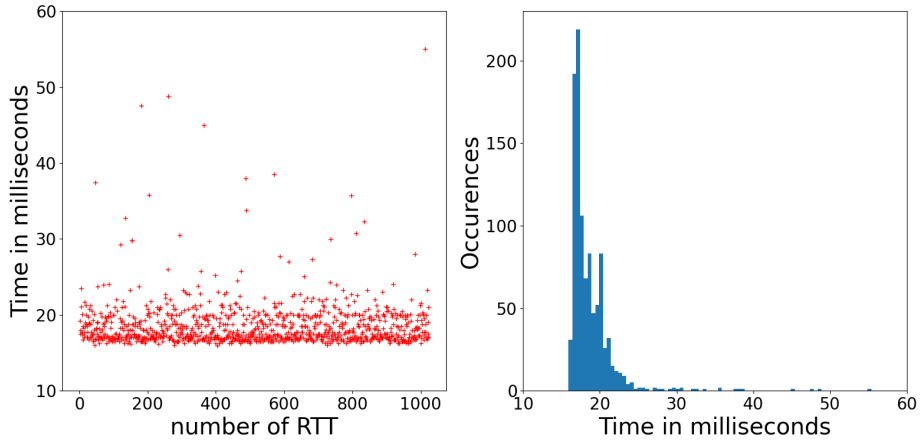


Figure 2.11 – A sample measured with random size packets (between 896 to 1152 bytes)

Table 2.2 – Comparison of Variance, and Smallest Representative Interval for sample from constant and random packet size

packet size in bytes of sample S	$var(S)$	$\lvert I_{0.9}(S)\rvert$	$I_{0.9}(S)$
1024	2.12	3.77	$[15.98, 19.75]$
random in $[896; 1152]$	10.25	4.85	$[16.20, 21.05]$

It appears that the randomness in the size of the packets induces a light increase of the sample spreadness. Indeed, the variance is almost multiplied by 5 in the random size sample, due to an increased number of outliers, while the representative size is only increased by 1 millisecond.

2.2.4 Impact Caused by the Presence of a Relay

The method used to perform a relay was to force the sender to communicate directly with the intermediary party instead of the receiver, then to make the intermediary transmit the packets to the receiver, and following the same method for the response from the receiver to the sender. Figure 2.12 illustrates the path of a packet p during a relay simulation by a node $\mathcal{I}$ between a node $\mathcal{S}$ and a node $\mathcal{R}$.

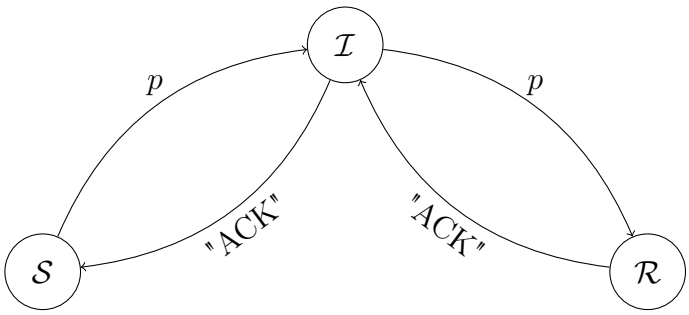


Figure 2.12 – Illustration of a relay between node $\mathcal{S}$ and node $\mathcal{R}$ through node $\mathcal{I}$

Figures 2.13 and 2.14 shows the impact of a relay over the RTT for exchanges between two couples of node: (Poland,Oregon), for an intercontinental test (see Figure 2.15), and (Toulouse,Rouen) for a national test (see Figure 2.16). For both cases, three samples are displayed, the first and third ones being genuine conversation and the second one being relayed through a node based in Caen, France.
It appears that the relay creates a drastic impact on the measured time. This impact being up to 150 milliseconds for the intercontinental exchange, and around 20 to 30 milliseconds for the national one. Given the previously observed stability and continuity of genuine samples, this kind of relay clearly stands out as an anomaly, even for the human eye.

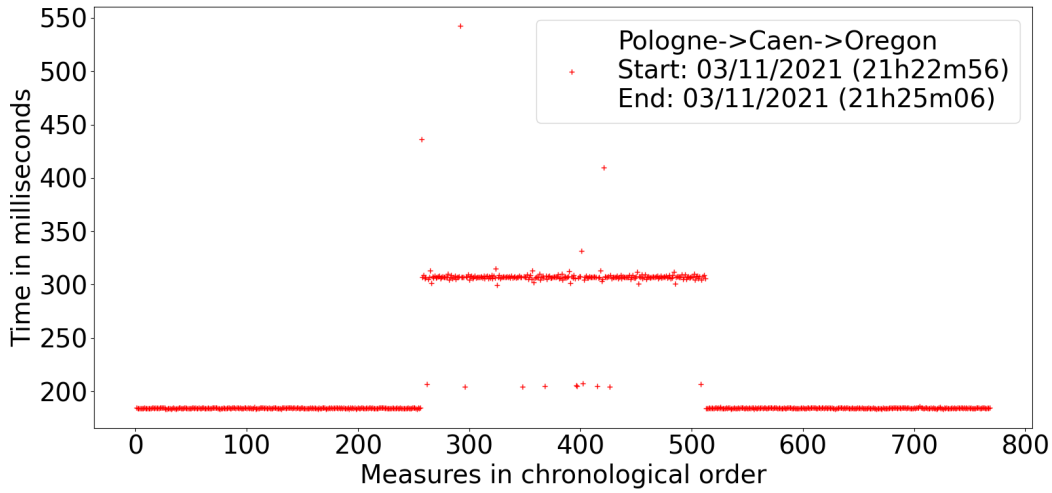


Figure 2.13 – 3 samples of size 256 captured in a row between Poland and Oregon, the second one being relayed through a node in Caen, France

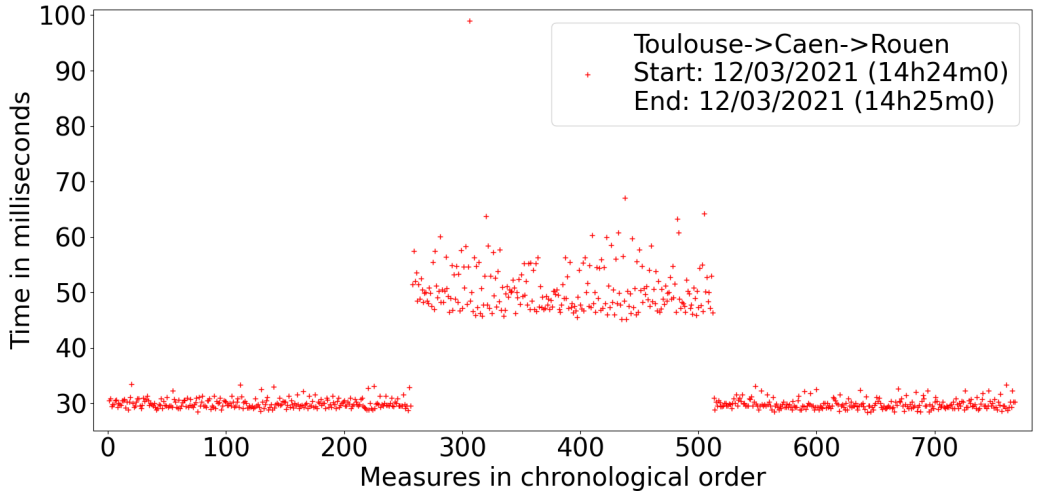


Figure 2.14 – 3 samples of size 256 captured in a row between Toulouse and Rouen, the second one being relayed through a node in Caen

Figure 2.15 – Position of nodes for an intercontinental exchange with relay: genuine nodes are represented with circles, attacker node is represented with triangle.

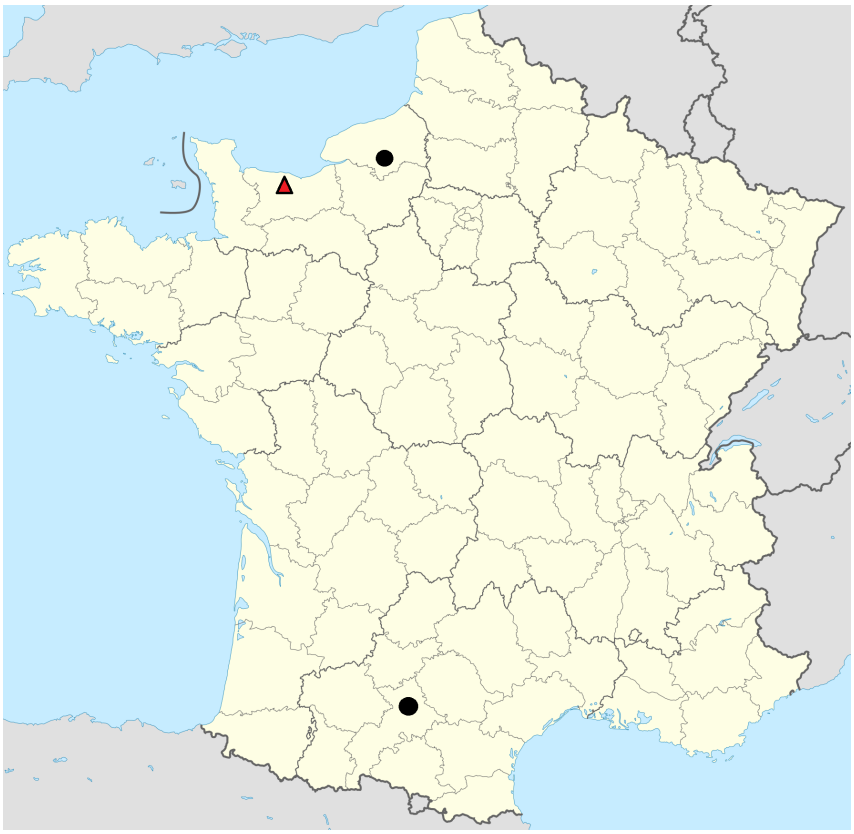


Figure 2.16 – Position of nodes for a national exchange with relay: genuine nodes are represented with circles (northern city is Rouen, southern city is Toulouse), attacker node is represented with a triangle (city is Caen).

2.3 Conclusion

Across all the observations that were made from those experiments, it seems that the measurement of time offers many usable properties for relay detection. First of all, for punctual samples, the collected measurements showed that RTT does get a satisfying stability. The main unit used to measure this stability was the 0.9-representative size, which is the length of the smallest interval of time containing at least 90% of the values of a sample, and has been of the order of the millisecond (16ms in the worst case) when using packets of constant size. For the more realistic observations with packets of random size, it appeared that the variance increased, due to a few more outliers in the sample, but the 0.9-RS remained satisfyingly l. Furthermore, this stability has demonstrated to hold over longer periods. For the continuous experiment running over one month, the 0.9-representative size remained at 2.08. It is however important to note that slight variations of the mean can be expected across multiple samples. It was the case for a collection of 18 samples collected over 4 months, during which the mean started around 189ms, then went down to around 184ms 20 days later, and finally went up to 187ms a month and a half later. Overall, those natural variations has proven to be weak in comparison with the observed impact of a relay.
From the two conducted experiments for national and international communications, the presence of a relay appeared highly noticeable with an impact of around 25 milliseconds for the national exchange from Toulouse to Rouen, relayed through Caen, and around 150ms for the intercontinental exchange from Poland to Oregon, relayed through Caen. For those specific routes, the impact on the time caused by the relay is more than enough to efficiently distinguish between a genuine route and a relayed one. Note that the method used to perform a relay simulates a prefix-attack on BGP but is not fully representative of an optimally capable attacker. Indeed, the impact level of such a relay may be caused by many factors, such as: the number of traversed routers, the location of the attacker, his proximity to a genuine route, his control over some network equipments and so forth. This means that there exists one or multiples optimum setups, lowering the impact of a relay to a minimum. Intuitively, the closer the relay node gets from a genuine route, the most time performant the attack gets. The same kind of issue exists for Distance Bounding protocols, where a mafia fraud attack will be detected only if the prover stands sufficiently far away from the verifier. In the specific case of Internet communications, a detection process will be effective up to a given limit of efficiency from the attacker, which will be discussed in Chapter 3.

Chapter 3

ICRP: Internet-friendly Cryptographic Relay-detection Protocol

"Night gathers, and now my watch begins.
I shall wear no crowns and win no glory.
I shall live and die at my post.
I am the sword in the darkness, the watcher on the walls.
I am the shield that guards the realms of men.
I pledge my life and honor to the Night's Watch.
For this night and all the nights to come."
- George R.R. Martin, A song of Ice and Fire

Introduction

The experiments presented in Chapter 2 provided good indicators on the possibility of taking advantage of time stability in a relay detection process. This Chapter offers a complete description of a protocol designed during this thesis. This protocol is named ICRP, standing for "Internet-friendly Cryptographic Relay-detection Protocol". In the following, ICRP is described according to 3 main points of interest.
Section 3.1 presents the global description of an ICRP run between 2 nodes, and the way this run can be practically adapted according to the nature of the interactions between the nodes. In Section 3.2, a so-called decision function is described. This function is the corner stone of ICRP as it is the one that analyzes the times gathered between the 2 nodes. The reliability of the ICRP protocol is highly related to the efficiency of this decision function. Consequently, the function is carefully evaluated. Finally, Section 3.3 describes a prototype implementation of ICRP. This implementation was done to support experiments aiming to measure the loss of performances inferred by the use of ICRP, in comparison with an identical communication without ICRP supervision. The prototype is carefully described along with the experiments on performances and their results.

3.1 Description

3.1.1 Cryptographic Background

Before jumping into the formal description, this section describes the cryptographic primitives used in ICRP. In this section the reader is assumed to be knowledgeable on the basics of complexity theory.

Cryptographic Hash Function

A hash function, or message digest function, is a primitive preserving the integrity of a message. The National Institute of Standards and Technology (NIST) provides the following definition [52].

Definition 3.1.1. ***Cryptographic hash function***
A hash function H maps a bit string m of arbitrary length to a fixed-length ℓ bit string.

$$\begin{aligned} H\colon \{0,1\}^* &\longmapsto \{0,1\}^\ell \\ m \quad &\longrightarrow H(m) \end{aligned}$$

The function is expected to have the following three properties:

- *Pre-image resistance: Given a randomly chosen target output, it is computationally infeasible to find any input that maps to that output. (This property is also called the one-way property).*
- *Second pre-image resistance: Given one input value, it is computationally infeasible to find a second (distinct) input value that maps to the same output as the first value.*
- *Collision resistance: It is computationally infeasible to find any two distinct inputs that map to the same output.*

The nature of the outputted element h depends on the needs the function is designed to answer, but one can always assume h to be an integer in the set $[0, 2^\ell - 1]$ (seen as a size ℓ binary vector) as it can be mapped to any size ℓ set.

Digital Signatures

A digital signature is a public-key cryptographic primitive serving the authenticity and integrity of a message. Contrarily to its manual counterpart, which is severely flawed on a security standpoint, the digital signature provides way better insurances. In a public-key cryptographic scheme, each user generates a key pair (pk, sk), with pk being publicly accessible and sk being secretly stored and only known by the user. A signature σ is

computed from an input message m and a user's private key sk. Knowing σ and the associated public key pk, anyone is able to verify with certainty that σ was indeed produced using the private key sk. Moreover, forging a signature σ' with no access to a targeted private key sk, will cause the verification to fail on input pk.
Formally:

Definition 3.1.2. ***Digital signature schemes***
a digital signature scheme is a set of 3 *algorithms* KeyGen, Sign, Verify_Sign, *described below:*

- KeyGen *takes as input a security parameter* λ *and outputs a key pair* (pk, sk).
- Sign *takes as input a message* m *and a secret key* sk *and outputs a signature* σ.
- Verify_Sign *takes as input a signature* σ, *a message* m *and public key* pk *and outputs a binary* $b \in \{0, 1\}$.

These algorithms are expected to have the following two properties:

- *correctness: the signature of a message* m *with key* sk *always verifies with public key* pk*:*

$$Pr\Big(\mathsf{Verify\_Sign}(\mathsf{Sign}(m, sk), m, pk) = 1\ \Big) = 1$$

- *unforgeability: for any polynomial time algorithm* $\mathcal{A}$ *with no access to* sk, *it is computationally infeasible to output* σ *such that:*

$$\mathsf{Verify\_Sign}(\sigma, m, pk) = 1$$

3.1.2 The protocol

The ICRP protocol runs between a sender $\mathcal{S}$ and a receiver $\mathcal{R}$.
A full run of a protocol consists of n rounds and allows one party to gather a sample of n RTTs concurrently to the sending of n packets $\{m_i\}_{i=1..n}$ containing data or messages (unlike Distance Bounding protocols where the exchange only serves an authentication purpose). If the total amount of data to be sent exceeds n packets, multiple executions of the protocol will be performed until the entire message has been transmitted. Each complete execution of the protocol is called a "session", and the total number of sessions needed to send a full message is denoted k. The use of ICRP to support the communication of $n \times k$ packets is called an ICRP supervision, and both parties must know the number of rounds n and the number of sessions k before launching such a supervision, which makes n and k public parameters. Ideally, the size of each packet over one session should remain constant [1], when this is the case, the packets size is denoted p. Table 3.1 summarizes this

1. This is not mandatory though, as the packet length creates few impact over measured time for

terminology.

Table 3.1 – Terminology for one or multiple ICRP execution

n	Number of rounds for one session
k	Number of sessions to send a full message
p	Size of packets (when constant) in bytes

ICRP execution overview: The scheme displayed on Figure 3.1 synthesizes a complete session of ICRP.
When a session is initiated, $\mathcal{S}$ marks n upcoming packets m_i by a random bit s_i, creates a timestamp t_i and sends $m_i||s_i$ to $\mathcal{R}$. In each marked round, $\mathcal{R}$ responds with a random bit r_i. Upon reception of r_i, $\mathcal{S}$ creates a new timestamp t'_i. The RTT of the current round is actually the time difference $t'_i - t_i$. This RTT is stored until the set $\{t'_i - t_i\}_{i=1..n}$ is complete.
Once the n rounds have been performed, $\mathcal{R}$ signs the hash of the m_i's along with the s_i's and the r_i's; this hash is denoted $H_\mathcal{R}$ and the signature $\sigma_\mathcal{R}$. Finally, $\mathcal{S}$ verifies that $\sigma_\mathcal{R}$ is a valid signature on $H_\mathcal{S}$ and let the collected sample of RTTs be analyzed by a *Decision Function* called Verify_Time. The signature authenticates $\mathcal{R}$ and confirms that the content of the exchange has not been altered. Signing the bits $\{s_i\}$ and $\{r_i\}$ also links the signature to this specific exchange which avoids possible replay attacks with the same packets $\{m_i\}$. The purpose of function Verify_Time is to accurately decide whether communications are genuine or relayed communications. The decision process in itself does not impact how the protocol runs, so it can be seen as an interchangeable blackbox primitive. Section 3.2 will present a particular design of this so-called decision function, which will be experimentally validated.

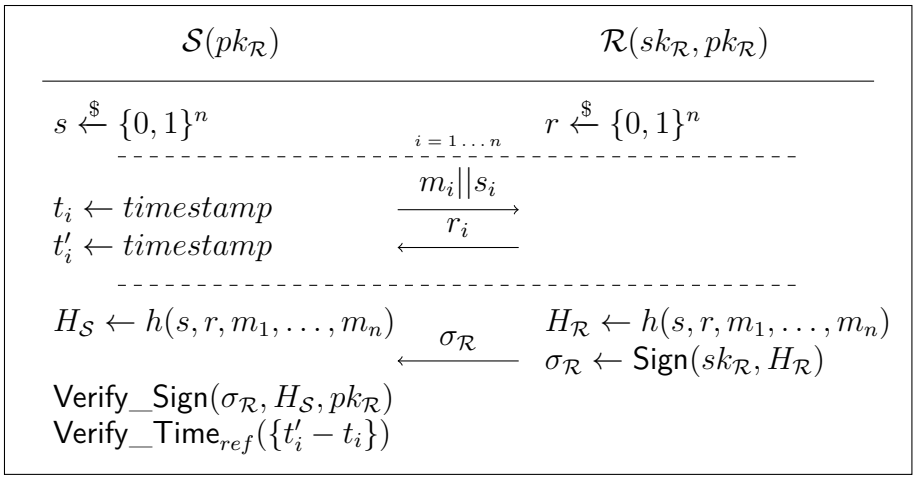


Figure 3.1 – an ICRP session for traffic hijacking detection with verification performed by $\mathcal{S}$

reasonable variations (see Chapter 2, Section 2.2.3)

In the previously described overview of an ICRP session, the party performing the verification is the party $\mathcal{S}$ willing to send the messages. In some cases though, $\mathcal{R}$ might be the party willing to perform the verification that no relay was on going. This is achievable by swapping around the time measurements and by making $\mathcal{S}$ sends to $\mathcal{R}$ some initialization message init message indicating to $\mathcal{R}$ that the protocol may start, this variant is described in Figure 3.2.

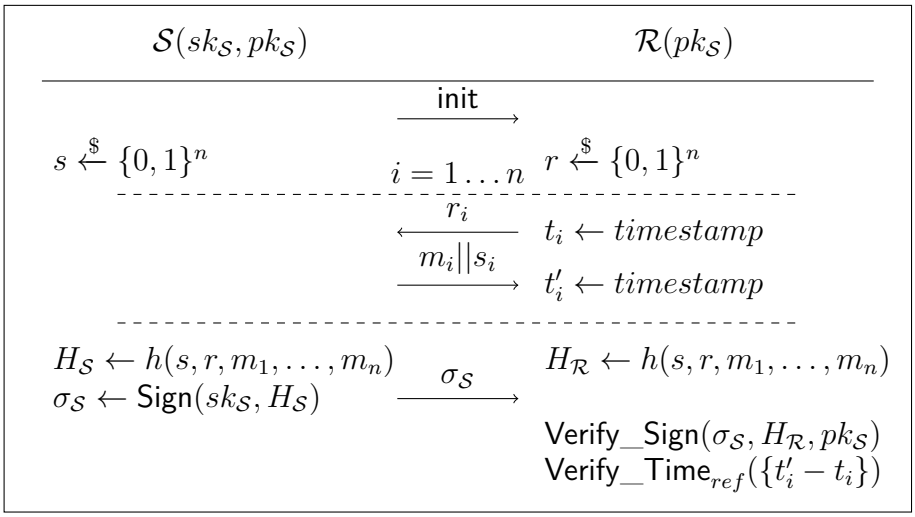


Figure 3.2 – an ICRP session for traffic hijacking detection with verification performed by $\mathcal{R}$

3.1.3 Active and Passive Modes

There are 2 main ways that ICRP can be used depending on the type and amount of data to be sent, and the nature of the exchanges of the 2 nodes $\mathcal{S}$ and $\mathcal{R}$:

1. Passive mode: passive mode is defined by a long-term background use of the protocol. This mode is relevant in cases where $\mathcal{S}$ and $\mathcal{R}$ frequently exchange small amounts of data. In this case, a complete session of the protocol might be achieved over multiple exchanges. The protocol then passively keeps track on the overall number of sent packets, and performs the verification when n packets have been sent. For instance, an industry daily updating some private information about customers to a distant database will most likely run ICRP in passive mode.
2. Active mode: active mode is defined by an execution of the protocol used specifically over the sending of one or several large files. This case is relevant when $\mathcal{S}$ and $\mathcal{R}$ exchange larger amounts of data on a more spread out frequency. In this case, multiple sessions of the protocol might be achieved over a single fast stream flow. For instance, a governmental institution needing to send a very large file containing classified videos, images, and texts, to a collaborating entity will most likely run ICRP in active mode.

Figure 3.3 illustrates how and when ICRP runs in both modes. It shows that, in active mode, the verification phase must be done concurrently to the measuring phase for performance's sake. This is not especially required for passive mode.

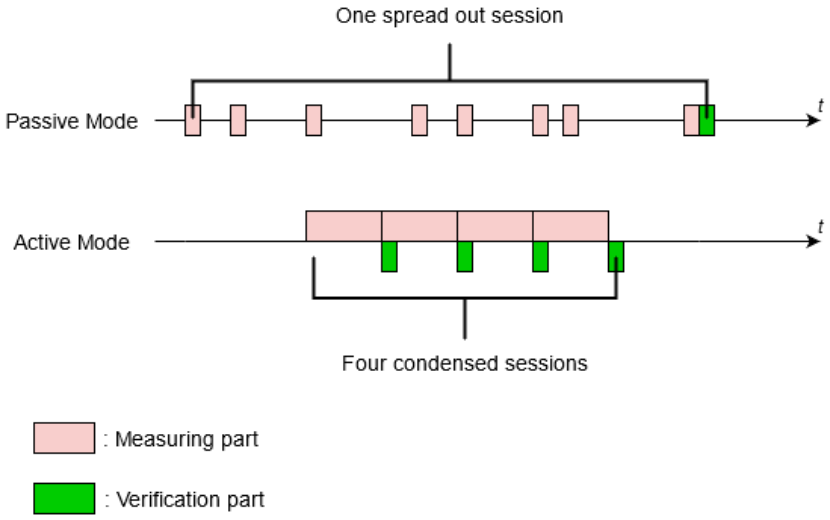


Figure 3.3 – Active and passive modes overview

3.2 Decision Function

3.2.1 Reference Sample

In Distance Bounding protocols, the samples of time are compared with the time needed by light to travel a pre-set distance bound. This process allows an absolute reference legitimated by a trusted law of physic, and common to any users, anywhere on Earth. In the context of Internet, there is no such convenient ways to analyze the collected times.
The reference sample ref consists of a large set of measures gathered in advance during a learning phase performed between $\mathcal{S}$ and $\mathcal{R}$. It represents the standard values that they can expect when measuring RTTs between them, and therefore, is unique for each possible couple ($\mathcal{S}$,$\mathcal{R}$). It is worth noting that the learning phase should take place when there is no ongoing attack, that is, when the route taken by the packets during the measurements has not been altered by a malicious party.
The reference sample should be updated when the genuine RTTs deviate from their reference due, for example, to modifications in the network topology. Experiments presented in Chapter 2, Figure 2.7 show that such a modification may occur, but does not cause a drastic change on the measures in comparison with the impact of a relay.
In environments where RTT is subject to more frequent continuity changes, one can consider performing dynamic updates of the reference sample to improve the reliability of the protocol. For example, any new valid execution of the protocol provides 256 fresh

RTTs that can be concatenated to ref while the 256 oldest ones can be removed from ref. Automatic updates should be monitored, though, as it may be exploited by an attacker to poison the reference sample.

3.2.2 Description

The decision function noted Verify_Time, defined in the following, takes as input a set of n real numbers formed by a sample of RTT and outputs a bit $b = 0$ when accepting the sample or $b = 1$ when rejecting it.

$$\mathsf{Verify\_Time}_{p,t}: \begin{array}{ccc} \mathbb{R}_+^n & \to & \{0,1\} \\ samp & \mapsto & b \end{array}$$

Having observed in Chapter 2 that the presence of a relay creates a clear increase in RTT, the following positional argument can be used as a decision process.
Verify_Time uses 2 parameters:

1. t is a threshold computed from the reference sample ref.
2. p is the maximum proportion of $samp$ expected to exceed the threshold t.

Using an appropriate threshold, the decision function only accepts samples having at most a given proportion above the threshold, and rejects it otherwise. The decision function is more accurately described in the pseudocode displayed on Algorithm 1.

Algorithm 1 Verify_Time pseudocode

```
    Input samp, p, t
    Output 0 or 1
  counter = 0
  for i in samp do
      if i ≥ t then
          counter = counter + 1
      end if
  end for
  prop = counter / len(samp)
  if prop ≥ p then
      return 1
  else
      return 0
  end if
```

3.2.3 Analysis of the Efficiency

Verify_Time outputs a binary response: 0 if the tested sample is considered genuine, 1 otherwise. Throughout this section, Verify_Time is challenged with both genuine and

relayed samples in order to analyze its efficiency using the following terminology:

— False positive : Verify_Time outputs 1 to a genuine sample

— False negative : Verify_Time outputs 0 to a relayed sample

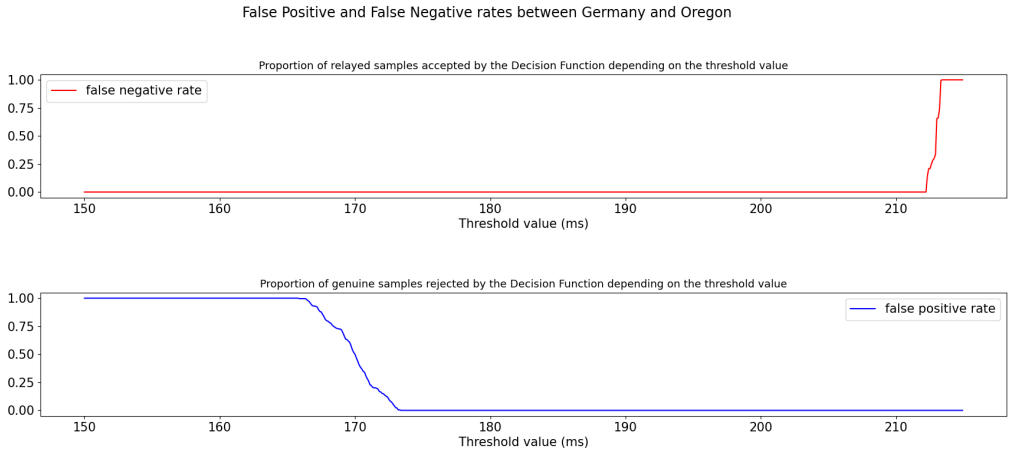


Figure 3.4 – Evolution of false positives and negatives for exchanges between Germany and Oregon

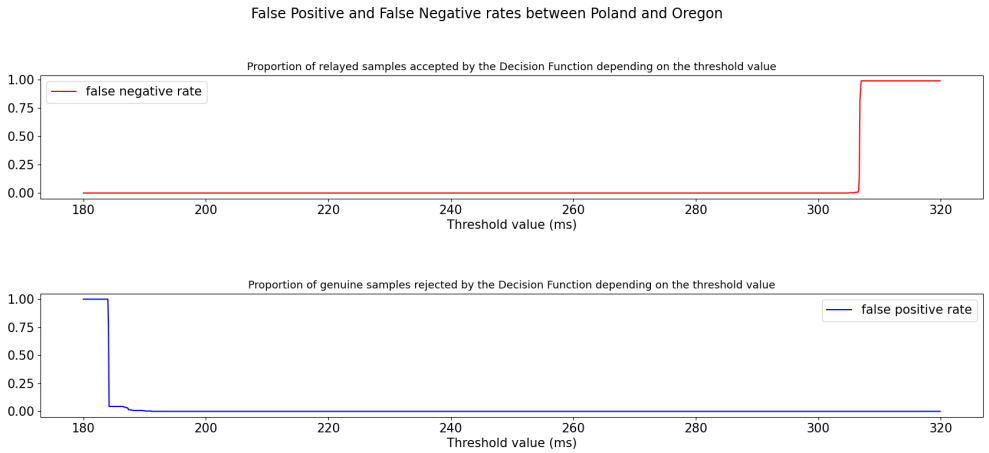


Figure 3.5 – Evolution of false positives and negatives for exchanges between Poland and Oregon

Tests on Verify_Time were performed on about 500 samples. Some of which are genuine communication between a sender $\mathcal{S}$ and a receiver $\mathcal{R}$, and the rest issued from a simulated relay where $\mathcal{S}$ sends its packets to an intermediary node $\mathcal{I}$ which relays them to $\mathcal{R}$. The tests consist in the successive analysis of a series of samples by Verify_Time with increasing

threshold value, which allows to keep track of the proportion of false negative and false positive.
More than 500 samples were gathered for couples (Germany,Oregon) and (Poland,Oregon). The accepted proportion rate above the threshold was arbitrarily fixed at 0.3 The results of those tests are summarized in Figures 3.4 and 3.5.
It clearly appears that Verify_Time achieved perfect detection on those tests for many threshold values, $[173, 212]$ for (Germany,Oregon) and $[190, 307]$ for (Poland,Oregon). Interestingly, the displayed curves can almost be seen as a binary signal, with very few threshold values producing a rate of false positive or false negative in the interval $]0, 1[$. This is easily explained by the very high stability of the samples. Indeed, as it was observed in Chapter 2, most of the samples' smallest representative intervals (see definition 2.1.2) were very small. Hence, a very small increase of the threshold value of the decision function can impact almost all the tested samples at once.
It is also important to keep in mind that the curves representing the false negative rates are not specific to the couples ($\mathcal{S}$,$\mathcal{R}$) involved, but are actually fully related to the attacker's capacity. Indeed, a more efficient attacker would be able to perform a relay causing less impact on the expected times, and therefore, to generate false negative with lower thresholds. Choosing a suited threshold then becomes a matter of appreciation of how efficient an attacker can get. To the contrary, the false positive rate is almost specific to the involved couples. It should not change a lot even if this curve was to be recomputed using another sample set because of the observation made on continuous samples in Chapter 2. Consequently, the false positive curve could even be computed in a practical scenario, if supplied enough trusted samples.

3.2.4 Choosing the Threshold

As it was stated in Section 2.2.4, the efficiency of an attack may depend on many factors: the attacker's connection speed, the current network topology, or the proximity from a genuine route between the source and destination are non-exhaustive examples. To detect an attacker disposing of an efficient relay setup, the decision function should be set to the highest sensitivity that can be supported. This means that the threshold must be as close as possible to the point where the false positive rate curve reaches 0. Although it is technically possible for users to gather enough trusted samples and to compute this optimal threshold value, this does require each couple of user to gather multiple samples over a decent period of time, which makes it unrealistic to ask in a practical context. The following procedure describes a way to choose p and t as suitable parameters for Verify_Time using only the reference sample ref.

1. Compute the smallest 0.9-representative interval $[a, b]$ of the trusted reference sample ref.
2. Let α be the proportion of elements of ref greater than b.

$$\alpha = \frac{|\{x \in \ ref; x > b\}|}{|ref|}$$

3. b is then the smallest possible value such that $\mathsf{Verify\_Time}_{ref,\alpha,b}(ref) = 1$.
4. Set a small margin m to avoid false positives caused by standard long term variation, for instance $m = 5$ (milliseconds).
5. Set $t = b + m$.

The choice of the margin m can be nothing but arbitrary, it depends only on how efficient an attacker can get. Allowing the samples to live about 5 to 10ms higher than normally expected trades off the insurance of very few false positives against the possibility of an attack, assuming that such an efficient relay is achievable between those nodes.

3.3 Implementation

In this section, ICRP's performances are evaluated. Firstly, an overview of a prototype implementation is supplied by explaining the problem with the sequential representation given in Figures 3.1 and 3.2. Then, the results based on this implementation are analyzed regarding 3 main points of attention: (1) the computational capabilities for the main operations performed by ICRP (hashes, signature, verification and decision function), (2) the throughput capabilities in comparison with an unsupervised communication, and (3). the overhead added compared to an unsupervised communication.

3.3.1 Prototype Description

Multi-Threading

The ICRP protocol as displayed in Figures 3.1 and 3.2 has a very clear downside, which is its sequentiality. Indeed, with this representation, each time $\mathcal{S}$ sends a packet, he has to wait for a response before sending the next one. This is especially problematic when ICRP runs in active mode and aims to send many consecutive packets, because it would highly slow the process. Hence, $\mathcal{S}$ needs to concurrently perform the sending and the reception of $\mathcal{R}$'s responses.
Similarly, if $\mathcal{S}$ and $\mathcal{R}$ need to run multiple consecutive sessions in active mode, the verification and authentication part of the protocol must not be realized sequentially as it would force $\mathcal{S}$ to wait to the end of the session to start a new one. Figure 3.6 schematically shows the differences in terms of efficiency between 3 simplified models of implementation for $\mathcal{S}$'s side: the top one is the sequential implementation, the middle one displays 2 concurrent threads, the first one for the sending, and the second one for the reception of acknowledgements and the verification part, the bottom one displays 3 concurrent threads for sending, receiving and verifying. The dotted lines represent a repeated operation, while the solid lines represent inactive periods of time for the current thread.

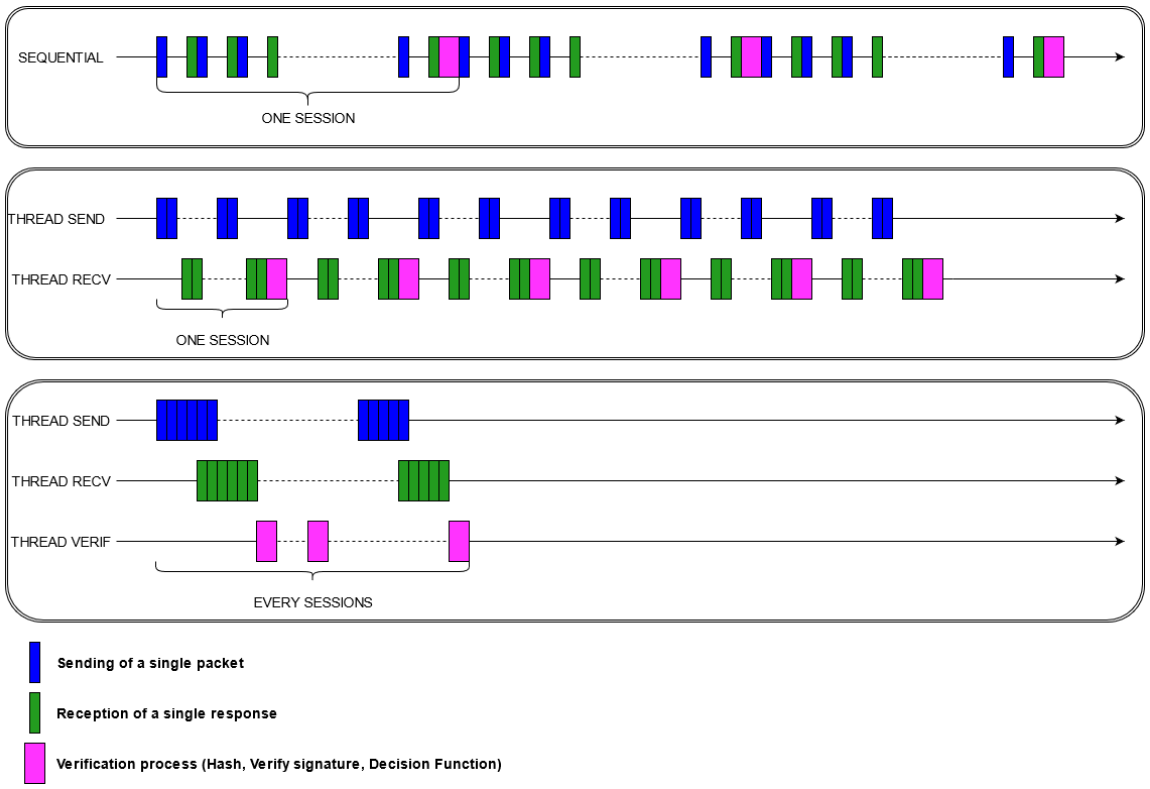


Figure 3.6 – Execution in active mode using 3 different implementations ($\mathcal{S}$'s side)

The ICRP prototype is implemented according to the bottom model of Figure 3.6. The third part handling verification is separated in 3 threads for synchronization purposes. The other party $\mathcal{R}$ is also implemented concurrently, with one thread handling the reception of packets and the sending of the response, and a second thread performing the cryptographic computations. We provide below a description of each thread actions.
On $\mathcal{S}$ side:

1. Thread send: this thread is in charge of sending all the packets to $\mathcal{R}$ and generating a timestamp when it does. It then stores the timestamp in a structure shared by all threads.
2. Thread recv: this thread is in charge of receiving every response from $\mathcal{R}$, generating a timestamp when it does, and computing the RTTs from the timestamps placed in the shared structure.
3. Thread pre_Hash: this thread is in charge of updating the Hash context with the values known beforehand by $\mathcal{S}$. That is the content of packets $\{p_i\}$ and the bits s_i.
4. Thread final_Hash: this thread is in charge of updating the Hash context with the values received from $\mathcal{R}$. That is the bits r_i.
5. Thread verif: this thread is in charge of receiving $\mathcal{R}$'s signature and waits for all the data it needs to be available from other threads. It then proceeds to check the signature and applies the decision function on the RTTs for the current session.

On $\mathcal{R}$ side:

1. Thread recv: this thread is in charge of receiving the packets and sending the responses to $\mathcal{S}$.
2. Thread auth: this thread is in charge of updating the Hash context with the values known beforehand by $\mathcal{R}$. That is the bits r_i. It then waits for all the data it needs to be available from other threads (the content of packets $\{p_i\}$ and the bits s_i), proceed to Hash&Sign and finally sends the signature to $\mathcal{S}$.
3. Thread auth: this thread waits for all the data it needs to be available from other threads (the content of packets $\{p_i\}$ and the bits s_i). It then updates the Hash context with these values and the ones known beforehand by $\mathcal{R}$ (i.e. the bits r_i). Finally, it proceeds to Hash&Sign and sends the signature to $\mathcal{S}$.

   Note that $\mathcal{R}$'s implementation does not have a pre-hash thread, it can not precompute anything in advance, because the hashes computed by $\mathcal{S}$ and $\mathcal{R}$ must match, and so, must be computed in the same order.

Using multiples threads to boost up the performances forces the parties to tag each packet with a sequence number in order to link each message and response to the correct round and session and don't get confused with the time measurements. Consequently, a 4 bytes header is added to the packets, indicating the sequence number. These 4 bytes are formed by 2 bytes indicating the current session number, followed by 2 bytes indicating the current round number. Writing the round and session indexes on 2 bytes each is good enough to fit the experimental needs of this section while being easily implemented for a prototype. In a real case scenario though, the number of rounds n for an ICRP session should not be greater than 512 because a relay detection can only occur once the n rounds are over. Hence, in a definitive protocol specification, the field for the round index in the 4 bytes header should be limited to only 9 bits, leaving 23 bits for the session index. The choice of a 4 bytes header is also influenced by the size used for the field SQN in TCP header. This field serves the same purpose of keeping the sessions synchronized between the nodes, and is reset once reached the maximum value of 2^{32}, which is high enough to ensure not having two packets with the same SQN transiting at the same time.

3.3.2 Performances

Recall that ICRP has 3 main parameters that might impact the performances:

— n: Number of rounds for one session.
— k: Number of sessions to send a full message.
— p: the size of each packet in bytes, p is also assumed to remain constant over rounds and sessions.

Note that n also defines the size of a collected sample and the number of packets sent during one session.

Definition 3.3.1. (n, p, k)***-sending***
Let $n, k \in \mathbb{N}$ be respectively the number of rounds and sessions needed to send a complete message with packets of constant size p. An ICRP supervision for a sending with this configuration is called a (n, p, k)-sending

Complexity of the Computations

The choice of the cryptographic primitives is left to the users to fit their needs, as they are interchangeable in this protocol. For the following experiments, were arbitrarily chosen $SHA256$ as the hash function, and $RSA2048$ as the Signature algorithm. Those choices are voluntarily poor performance-wise. However, they allow to get an upper complexity bound.
For each session, $\mathcal{R}$ (respectively $\mathcal{S}$) performs Hash&Sign (respectively Hash&Verify) over the packets $\{p_i\}_{0 \leq i < n}$ and the random bits $\{r_i\}_{0 \leq i < n}$ and $\{s_i\}_{0 \leq i < n}$. This is $n \cdot (8p + 2)$ bits of data to be hashed. $SHA256$ is based on the Merkel-Damgård construction, this means that the message to hash is separated into blocks of same size which are processed by a compression function. Hence, its complexity is linear in the number of blocks involved. $SHA256$ uses 512-bits blocks, so for each session, the Hash complexity will be $\mathcal{O}(np)$ in the number of compression function, given by the following computation: $\frac{n(8p+2)}{512} = \frac{n(4p+1)}{256}$
Table 3.2 shows the number of applied compression function and the corresponding hashing time depending on n with a fixed value of $p = 512$. Regarding the signature and

n	256	512	768	1024	1280	1536
#compressions	2049	4098	6147	8196	10245	12294
Hashing time(ms)	0.731	1.479	2.253	2.834	3.813	4.434

Table 3.2 – ICRP final Hash complexity (using $SHA256$) with $p = 512$ bytes

the verification, the input value is always a 256-bit string, and so the time taken for this operation remains constant for both operations.
Finally, $\mathcal{S}$ runs the decision function on the sample. This process is linear in the size n of the sample as it goes through the table of RTTs and increment a counter every time the treated value is higher than the chosen threshold. Note that n should not be too large because the verification is done each n packets sent. Hence, a high n leaves a wider amount of data to be relayed before the detection. We believe $n = 256$ or $n = 512$ to be the most suitable choices. These values being very small, we can consider the decision complexity to be negligible.
Note that, the slower is the overall verification process, the later will be detected a suspicious sample. However, as the authentication and verification are done concurrently with the other processes, those times do not impact the throughput performances.

Throughput

In this section, the impact of parameters n, p, and k over the sending time of numerous packets is analyzed. Those times are then compared with the throughput given by the communication of the exact same amount of data without using ICRP, this method is later referred to as "direct sending".

Impact of parameters n and k: Table 3.3 shows the times in seconds involved in a $(n, 500, k)$-sending between a node in Poland and a node in France (Caen) for a specific amount of data.

size (Mb)	n	k	n	k	n	k
	250	160	320	125	400	100
20	3.65		3.56		3.53	
size (Mb)	n	k	n	k	n	k
	250	800	320	625	400	500
100	16.71		16.81		16.62	

Table 3.3 – ICRP sending times in seconds for 20Mb and 100Mb depending on n (number of rounds) and k (number of sessions).

This allows to observe how the global communication time changes when n and k are attributed different values for the same amount of data:

— 20 Megabytes using 3 possible values of (n, k): $(250, 160)$, $(320, 125)$ and $(400, 100)$.

— 100 Megabytes using 3 possible values of (n, k): $(250, 800)$, $(320, 625)$ and $(400, 500)$.

The parameter p remains constant to 500 bytes for those tests. It clearly appears that the number of sessions k and rounds n creates no visible impact on the overall sending time.

Impact of packet size p: As it was stated in Chapter 2, the packet size has low impact over the sending time of a single packet. This means that the more data contained within every single packet, the faster the sending of the overall message (containing multiple packets) will be.

Figure 3.7 shows the evolution of the time to send 100 Megabytes of data depending on the size of the individual packets. Expectedly, the time decreases for increasing values of p. The maximal size of *UDP* packets is implicitly specified in the official IETF documentation RFC768 [56], as the UDP header contains a field called "Length" which is 16 bits in size and represents the length in bytes of the packet (header included). This means that the theoretical maximum size for a UDP packet would be of 65535 bytes. In practice however, most services (for instance DNS) restrict the maximum packet length to respect the Maximum Transmission Unit (MTU) on the Internet.

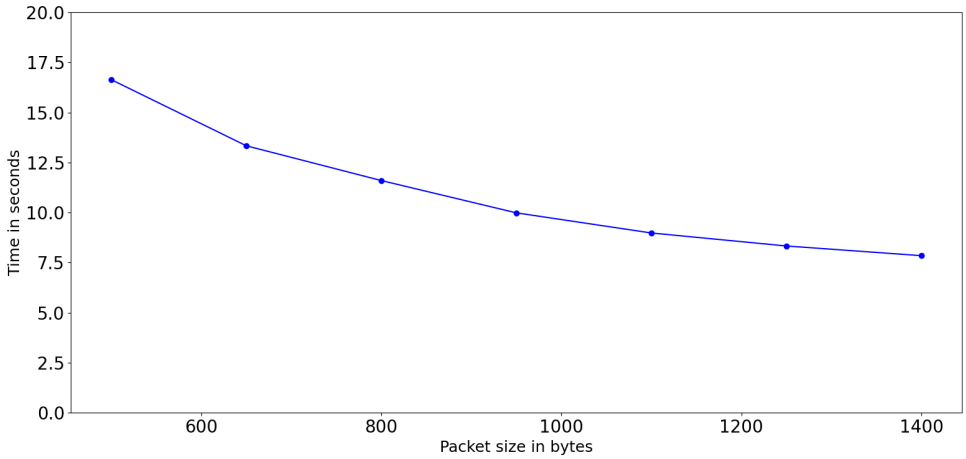


Figure 3.7 – Time to send 100Mb depending on the individual packet size

Comparison with a direct sending of UDP packets: To see how ICRP performs, it is necessary to compare the times involved in the sending of a given amount of data between two fixed nodes with and without an ICRP supervision. Using the prototype implementation and only raw UDP packets including no time measurements, authentication, or acknowledgements, allows to measure the time complexity added ICRP.
As Figure 3.7 has demonstrated, the size of individual packets has an impact on the global sending time of a file. Hence, those 2 methods should send packets of comparable sizes. The following Table 3.4 displays the measures of the overall sending of 10, 40, 100 and 200 Megabytes of data using the two methods with a constant packet size of 500 bytes and compares the obtained throughput. The average throughput is about 2.15Mb/s for the

Sending method / Size and parameters (n, k, p)	raw UDP	ICRP
10Mb $(250, 80, 500)$	4.668	6.138
40Mb $(250, 320, 500)$	20.010	21.936
100Mb $(250, 800, 500)$	44.384	54.323
200Mb $(250, 1600, 500)$	89.916	106.508

Table 3.4 – Overall sending times in seconds for large files

direct sending and about 1.79Mb/s using ICRP. The measures were performed between a personal computer based in Caen, France, and a Server supplied by AWS (Amazon Web Services). The slight loss in performance is due to the fact that ICRP has to handle multiple threads concurrently on both Sender's and Receiver's side, which is not the case for the method sending raw UDP. It induces that the processing capabilities of the

end points has an impact on throughput performances. This impact remains low though, as in this experiment, the sending machine was a personal laptop with few processing capabilities, and still limited the throughput loss to 16.7%.

Volume

Finally, this section quantifies the volume of overhead data added through a (n, p, k)-sending in active Mode in comparison with the amount of raw information transmitted ($n \cdot p \cdot k$ bytes).

1. Each message (resp. response) is marked with a bit s_i (resp. r_i). This gives an additional $2 \cdot n \cdot k$ bits of information traveling through the network.
2. Each message (resp. response) is complemented with a sequence number encoded onto a 4 bytes header. This adds another $64 \cdot n \cdot k$ bits of additional data traveling.
3. During the verification part of ICRP's prototype implementation, a $RSA2048$ signature is sent for each session with an additional 2 bytes tag indicating the current session number. This adds another $(2048 + 16) \cdot k$ additional bits.

Overall, the total overhead of our protocol is $(66n + 2064)k$ bits. The proportion of additional data traveling through the network is :

$$\frac{(66n + 2064)k}{8npk} = \frac{66n + 2064}{8np}$$

Assuming realistic values $p = 500$ bytes, and $n = 256$ for the sample size, the overhead proportion per session is 1.85% Table 3.5 displays this proportion value for a few practical values of n and p.

Table 3.5 – Proportion of additional data induced by ICRP depending on (n, p)

n \ p	500	1000	1500
256	1.85%	0.93%	0.62%
512	1.75%	0.88%	0.58%

3.4 Conclusion

The ICRP protocol is based on the same idea as distance bounding protocols, namely using RTT to detect the presence of a relay attack. Taking place in a completely different environnment, ICRP must apply a different strategy to analyze the collected time samples. This strategy takes the shape of decision function comparing a trusted reference sample with the sample to be analyzed. This method is empirically supported by the many observations that were made in Chapter 2 and that showed the suitability of RTT behaviour over the Internet for such a decision process.
It is simple in its design and is not sensitive to the cryptographic primitives it uses, neither it is to possible updates on routing protocols. Indeed, the only important property to achieve is time stability over the Network. Moreover, it does not rely on any intermediate actions from the network equipments involved in the communication between users.
The implementation of a prototype has allowed many performance tests:
Regarding the decision function Verify_Time, a solution taking into account the uncertainty of the attacker capabilities has been tested and offered convincing results in terms of false positive and false negative returns. Given the observations made in Chapter 2, Verify_Time is able to perform a 100% detection results with 0 false responses as long as the best possible relay attack is assumed to have an impact greater than a reasonable user-defined margin.
Regarding the capabilities of the protocol, using multi-threading to handle communication and verification has allowed to keep a satisfying throughput for sending large amounts of data, while keeping the volume of data overhead to about 1%.

Chapter 4

Security Analysis : Model and Proof

"When you play the game of thrones, you win or you die.
There is no middle ground."
- Cersei Lannister

Introduction

A very common way to organize security proofs for cryptographic schemes is the method based on sequences of games. This method is now quite popular as it offers an important procedural simplification in comparison with former techniques that could quickly get complex.
In this section, we will recall the basics in terms of cryptographic security and security proofs and present the threat model for a standard proof of Existential UnForgery on a signature scheme. This cryptographic primitive and security property will be assumed for proving the security of ICRP against relay attacks.

4.1 Background on Game-based Security

4.1.1 Proving Security

A security property is defined as the capacity for a cryptographic scheme to resist a given attack. An attack is usually pictured as a game opposing an attacker $\mathcal{A}$ and a challenger $\mathcal{C}$, which are probabilistic algorithms. This view allows to consider the game as 2 complementary probabilistic events G and $\neg G$ being:

$$\begin{array}{rcl} G & : & \text{“}\mathcal{A} \text{ wins the game”} \\ \neg G & : & \text{“}\mathcal{A} \text{ looses the game”} \end{array}$$

Then, proving the security of a scheme against an attack, is showing that $Pr(G)$ is negligibly close to a targeted probability p. For instance, the game IND-CPA (standing for INDistinguishability for Chosen Plaintext Attack) for an encryption scheme pictures an attacker trying to distinguish if a ciphertext c is computed from the encryption of a message m_0 or a message m_1, even when m_0 and m_1 are selected by the attacker himself. For an encryption scheme to be IND-CPA secure, the associated game should define an event G such that $Pr(G)$ is negligibly close to $\frac{1}{2}$ to express that $\mathcal{A}$ can only do negligibly better than random guessing. This difference between the effective probability $Pr(G)$ and the targeted probability is called the advantage of the attacker for the game. See the formal definition of 2 quantities negligibly close below:

Definition 4.1.1. ***Negligibly close to***
Let f and g be 2 applications.
f is said to be negligibly close to g if:

$$\forall P \in \mathbb{R}[X] \text{ such that } \forall x, P(x) > 0$$

$$\exists \Lambda \in \mathbb{R}^+, \ \forall \lambda \geq \Lambda, \ |f(\lambda) - g(\lambda)| < \frac{1}{P(\lambda)}$$

For a given variable λ, an application negligibly close to 0 is just said to be negligible. Without further precision, such an application can be noted $negl(\lambda)$

More specifically, cryptographic schemes are always defined in relation to a so-called security parameter λ. This security parameter is an integer increasing the computationnal power needed by an attacker for breaking the related scheme. Hence when defining the event G, the probability $Pr(G)$ is actually a function of λ which should be negligibly close to the targeted probability p (viewed as a constant application).
In many cases, the targeted probability of a security proof on a game $Game_0$ is itself the advantage of an attacker for another game $Game_1$. Then, showing that the probability $Pr(\text{“}\mathcal{A} \text{ wins } Game_0\text{”})$ is negligibly close to $Pr(\text{“}\mathcal{A} \text{ wins } Game_1\text{”})$ expresses the fact

that $Game_0$ remains secure as long as $Game_1$ is, and if the advantage of an attacker on $Game_1$ has already been proven to be negligible, the attacker's advantage on $Game_0$ is consequently also negligible. This process is called a reduction between $Game_0$ and $Game_1$.
To highlight a reduction between 2 security games $Game_0$ and $Game_1$, a common technique is to assume the existence of an attacker $\mathcal{A}$ winning $Game_0$ with non-negligible advantage and, using $\mathcal{A}$ as a black-box algorithm, to construct a second attacker $\mathcal{A}'$ winning $Game_1$ also with non-negligible advantage. If this approach on security proofs can be easily applicable for simple schemes, it can become far more difficult for more complex cryptographic primitives.

4.1.2 Transitions of Games

This section describes a systematic method for proving the security of a protocol or a cryptographic primitive and provide a toy-example on a classical digital signature scheme. This method introduces intermediary games to demonstrate a reduction between 2 main games, which is why the approach is called "game-sequence".

Concept of the proofs

The main idea of the game-sequence approach is to modify the original attack game $Game_0$ to construct a new game $Game_n$ through a sequence of small modifications defining the intermediate games $\{Game_1, \ldots, Game_{n-1}\}$. For all i between 0 and n, let G_i be the event:

$$G_i : \mathcal{A} \text{ wins } Game_i$$

In order to perform a proof with sequences of games, given the targeted probability p, and for $\{negl_i\}_{i=0..n}$ being n negligible positive functions, the construction of those games must be such that:

$$\begin{array}{rcl} \forall i \in \{0, \ldots, n-1\}, \quad \left|Pr(G_i) - Pr(G_{i+1})\right| & \leq & negl_i \\ \left|Pr(G_n) - p\right| & \leq & negl_n \end{array}$$

Once those statements proven to be true, it is then easy to obtain the desired result. According to the triangular inquality, it is true that:

$$\begin{array}{rcl} \left|Pr(G_0) - p\right| & \leq & \sum_{i=0}^{n-1} \left|Pr(G_i) - Pr(G_{i+1})\right| + \left|Pr(G_n) - p\right| \\ & \leq & \sum_{i=0}^{n} negl_i \end{array}$$

Moreover:

$$\left|Pr(G_0) - p\right| \leq \sum_{i=0}^{n} negl_i \Longrightarrow Pr(G_0) \leq p + \sum_{i=0}^{n} negl_i$$

Any finite sum of negligible quantities is still a negligible quantity. Hence $Pr(G_0)$ is negligibly close to p.

Transitions based on a failure event

Before going further on game-sequence proofs, let us introduce the two following useful Lemmas:

Lemma 4.1.1. ***Difference Lemma***
Let E_1, E_2, and F be events defined in some probability distribution. Suppose that $Pr(E_1 \wedge \neg F) = Pr(E_2 \wedge \neg F)$. Then :

$$\Big|Pr(E_1) - Pr(E_2)\Big| \leq Pr(F)$$

Proof. The proof comes immediately after partitionning the events E_1 and E_2 on the occurence of the event F:

$$\begin{array}{rcl} \Big|Pr(E_1) - Pr(E_2)\Big| & = & \Big|Pr(E_1 \wedge F) + Pr(E_1 \wedge \neg F) - (Pr(E_2 \wedge F) + Pr(E_2 \wedge \neg F))\Big| \\ & = & \Big|Pr(E_1 \wedge F) - Pr(E_2 \wedge F)\Big| \\ & \leq & Pr(F) \end{array}$$

The second equality comes from the assumption of the lemma. The final inequality comes from the fact that $Pr(E_1 \wedge F)$ and $Pr(E_2 \wedge F)$ are both lesser than $Pr(F)$ □

Lemma 4.1.2. ***Union Bound***
Let $E_1, \ldots, E_n$ be n events defined in some probability distribution, and let E be the event $\bigvee_{i=1}^{n} E_i$.
Then :

$$Pr(E) \leq \sum_{i=1}^{n} Pr(E_i)$$

Proof. This can be proven with a simple recurrence reasoning.
Let E_1 and E_2 be 2 events, and $E = E_1 \vee E_2$.
Then:

$$\begin{array}{rcl} Pr(E) & = & Pr(E_1) + Pr(E_2) - Pr(E_1 \wedge E_2) \\ & \leq & Pr(E_1) + Pr(E_2) \end{array} \quad (1)$$

Therefore, the lemma is true for $n = 2$.
Assume that, for $E_1, \ldots, E_k$, being k events and E being the event $\bigvee_{i=1}^{k} E_i$, it is true that $Pr(E) \leq \sum_{i=1}^{k} Pr(E_i)$.
Let E_{k+1} be an event. Then:

$$\begin{array}{rcll} Pr(E \vee E_{k+1}) & \leq & Pr(E) + Pr(E_{k+1}) & \text{from (1)} \\ & \leq & \sum_{i=1}^{k} Pr(E_i) + Pr(E_{k+1}) & \text{from the assumption} \\ \bigvee_{i=1}^{k+1} E_i & \leq & \sum_{i=1}^{k+1} Pr(E_i) & \end{array}$$

Therefor the lemma is true for all n. □

During a game-sequence approach, the intermediate games can be designed according to multiple transition types. These transitions are defined so that the intermediate properties (i.e. $\forall i \in \{0,\dots,n-1\}, \left|Pr(G_i)-Pr(G_{i+1})\right| \leq negl_i$) are easy to demonstrate. This section will only explore the type of transition used for the proof of ICRP security, which is the transition based on a failure event.
The idea is to transition from $Game_i$ to $Game_{i+1}$ by making the attacker loose $Game_{i+1}$ if a certain event F occurs. By doing that, as long as the event F does not happen, an instance of $Game_i$ is perfectly indistinguishable from an instance of $Game_{i+1}$. Hence the following equivalence :

$$G_i \wedge \neg F \iff G_{i+1} \wedge \neg F$$

According to the Difference Lemma 4.1.1, this directly implies that:

$$|Pr(G_i) - Pr(G_{i+1})| \leq Pr(F)$$

Assuming F to be an event occuring with negligible probability, this construction validates the i^{th} intermediate property.
Curious readers can go further on the matter of security proofs based on sequences of games by referring to [68].

Existential UnForgery for Chosen Message Attack

A classical example of a security property would be the euf-cma (Existential UnForgery for Chosen Message Attack) security for signature schemes. The euf-cma security captures the unforgeability property for signature schemes stated in Chapter 3, Section 3.1.1, it addresses the resistance of a given signature scheme to the capacity of any adversary $\mathcal{A}$ to forge a valid signature on a message m, chosen by $\mathcal{A}$, given only a public key pk, and an unlimited number of call to a signing oracle $\mathcal{O}\mathsf{Sign}$ on any message $m' \neq m$. This signing oracle allows to prove that having access to many valid couples (m, σ) can only give $\mathcal{A}$ a negligible advantage on his attack.

$\mathsf{Exp}^{\mathsf{euf\text{-}cma}}_{\mathcal{A},\Sigma}(\lambda)$

$b \leftarrow 0$
$(pk, sk) \leftarrow \mathsf{KeyGen}(\lambda)$
$(\sigma, m) \leftarrow \mathcal{A}^{\mathcal{O}\mathsf{Sign}(\cdot)}(pk)$
if $(\sigma, m, pk) \notin Req$ AND $\mathsf{Verify\_Sign}(\sigma, m, pk) = 1$:
  $b \leftarrow 1$
return b

Figure 4.1 – The euf-cma game for a digital signature scheme Σ.

Here is a formal description of oracle $\mathcal{O}\mathsf{Sign}$:
$\mathcal{O}\mathsf{Sign}(m, pk)$: This oracle handles an internal list Req. Whenever the oracle is called on input m, the oracle returns a couple (σ, m) such that $\mathsf{Verify\_Sign}(\sigma, m, pk) = 1$ and adds it to list Req. The formal description of the euf-cma game is given in Figure 4.1, and the formal definition of euf-cma security is given in Definition 4.1.2.

Definition 4.1.2. euf-cma
A signature scheme Σ is said to be euf-cma *secure if the advantage of any adversary $\mathcal{A}$ in the game $\mathsf{Exp}^{\mathsf{euf\text{-}cma}}_{\mathcal{A},\Sigma}(\lambda)$ is negligible.*
Here, the advantage of an adversary is defined as a function in a security parameter λ:

$$\mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathcal{A},\Sigma}(\lambda) = Pr(\mathsf{Exp}^{\mathsf{euf\text{-}cma}}_{\mathcal{A},\Sigma}(\lambda) = 1)$$

Please note that the protocol ICRP, such as described in Chapter 3 actually uses the Hash&Sign paradigm, where the signature of the triple (m, s, r) is obtained by applying the signing function on the output of a collision resistant hash function on (m, s, r). Bellare and Rogaway showed in [13] that an Hash&Sign signature scheme is euf-cma secure, assuming that the related signature algorithm is euf-cma secure and that the outputs of the hash function are perfectly random. This result is the cornerstone of the Random Oracle Model, which we will describe in Section 4.2.1

4.2 Security Model

4.2.1 Context

This section overviews the attack game called the "Relay Game" for proving the security of ICRP against a relay by presenting the attacker's capacities, the underlying assumptions, and describing the attack. In the following, ICRP uses a decision function called Verify_Time. Verify_Time is seen as a black box using a parameter T_r called the reference sample and taking as input a time sample T. The signature scheme is denoted Σ and is assumed euf-cma secure.

Attacker's capabilities

The attacker is an entity $\mathcal{A}$ controlling one or several nodes. $\mathcal{A}$ is assumed to have the following capabilities:

— *Interception*: At any given time, $\mathcal{A}$ can alter the route between 2 nodes $\mathcal{S}$ and $\mathcal{R}$ so that the n next packets sent by $\mathcal{S}$ towards $\mathcal{R}$ will instead be forwarded to a node $\mathcal{I}$ controlled by $\mathcal{A}$. Consequently, $\mathcal{A}$ can either impersonate $\mathcal{R}$ by answering directly to $\mathcal{S}$ from node $\mathcal{I}$ or relay the information to $\mathcal{R}$ to observe passively the content of the exchanges.

— *Timing Knowledge*: For any given 2 nodes $\mathcal{S}$ and $\mathcal{R}$, $\mathcal{A}$ is aware of the reference sample T used by the decision function to analyze the time sample collected during an execution of ICRP.
— *Timing Control*: For any given 2 nodes $\mathcal{S}$ and $\mathcal{R}$ using a reference sample T for the decision function, there is a node $\mathcal{I}$ controlled by $\mathcal{A}$ such that the time sample collected during an execution of ICRP between $\mathcal{S}$ and $\mathcal{I}$ will be valid using the reference sample T.

Assumptions

Assumption 1 For any given 2 nodes $\mathcal{S}$ and $\mathcal{R}$, any alteration of the genuine path between $\mathcal{S}$ and $\mathcal{R}$ will be detected by the decision function.

Consequently, an attacker relaying an ICRP execution through a node $\mathcal{I}$ it controls will systematically fail to validate the time-check at the end of the protocol.
Note that, in a practical context, a relay inducing very small changes on the route has high chances to remain undetected by the function. For instance, consider a relay through a node adjacent to the genuine route like shown in Figure 4.2 with the genuine route between $\mathcal{S}$ and $\mathcal{R}$ in straight lines, and the altered route for a relay through node $\mathcal{I}$ in dashed lines.
At these minor scales, a relay most probably creates very few impact on the measured times, and therefore, might not get detected. However, it is also important to notice that a relay attack at this scale is not so realistic, mainly because a relay attack only becomes really problematic when information leaks on long geographical distance, e.g. governmental data leaving the national territory and causing clear geopolitical issues. As demonstrated in Chapter 3, the decision function achieves 100% detection accuracy for this kind of large scale relay, which is why Assumption 1 seems reasonable.

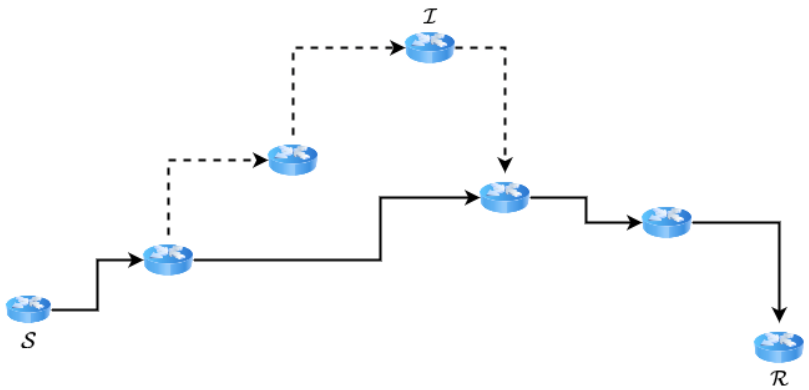


Figure 4.2 – A short relay in terms of additional intermediaries that might cause a sufficiently low impact on RTT for Verify_Time not to detect it.

Assumption 2 The oracles, games, and proof presented in the next section are placed in the Random Oracle Model (ROM).

This model assumes that hash functions over a set of given size ℓ are actually returning perfect random outputs. Precisely, in the Random Oracle Model, hash functions are replaced with oracles drawing their outputs according to the uniform distribution over $[0, 2^\ell - 1]$. Hence, any specific output of a random oracle is generated with the same probability $\frac{1}{2^\ell}$, while the same statement cannot be made for a classical hash function. The ROM is a frequently used model in cryptographic security as it usually brings convenient simplifications to the proofs.

Attack

The attack game called the "Relay Game" is an interaction between a challenger and an adversary or attacker. The game is set in a network, where entities (or nodes) are collaboratively participating to the routing of messages between 2 end-points (typically, the Internet). These entities can receive credentials allowing them to run ICRP as sending or receiving party. An entity having received such credentials is then called a participant.
Attacker's goal: The adversary must prove to the challenger that he has witnessed a complete execution of ICRP between 2 entities, such that ICRP has accepted the exchange. To do so, the adversary must outputs a tuple containing the identities of 2 entities $\mathcal{S}$ and $\mathcal{R}$, a message m, 2 binary vectors s and r, and a signature σ. The adversary wins the game if those values do match with an ICRP execution that has occurred, and been accepted. More precisely, if (1) the signature σ verifies for $\mathcal{H}(m, s, r)$ with the public key $pk_\mathcal{R}$, and if (2) the sample of RTT T, gathered during the same ICRP execution is accepted by Verify_Time.
Challenger's actions: The challenger deals with environment in which the adversary evolves. It can allow the usage by the adversary of specific oracles at specific moments of the game. These oracles are precisely described in Section 4.2.2. *Description of the game:*

- Learning phase: During the learning phase, the adversary is given the chance to manage his attack setup and learn as many information as he can get. Noticeably, the attacker can ask for the challenger to:
  - create new entities by distributing them a key pair so that they can execute an ICRP session.
  - corrupt existing entities by letting the adversary know their private keys, and controlling the nonces used during their ICRP sessions.
  - execute ICRP sessions between a sender $\mathcal{S}$ and a receiver $\mathcal{R}$ for a chosen message m. At the end of an interaction, the adversary obtain 2 tuples:
    $V_\mathcal{S} = (\mathcal{S}, \mathcal{R}, m, s_\mathcal{S}, r_\mathcal{S}, \sigma_\mathcal{S}, T)$ and $V_\mathcal{R} = (\mathcal{S}, \mathcal{R}, m, s_\mathcal{R}, r_\mathcal{R}, \sigma_\mathcal{R})$ where $V_\mathcal{S}$ contains the information sent, received, and computed by $\mathcal{S}$ and $V_\mathcal{R}$ contains the information sent, received, and computed by $V_\mathcal{R}$.

This phase allows the adversary to obtain as many couples $(\mathcal{H}(m,s,r),\sigma)$ as desired, with σ being a valid signature on $\mathcal{H}(m,s,r)$, and also to obtain precise knowledge about the reference sample used by each couple of entities.
Whenever the adversary decides, he designates the targeted victims of the attack, denoted $\mathcal{S}$ for the entity sending the message, and $\mathcal{R}$, for the entity receiving it. From now on, the adversary can no longer create or corrupt new entities, but still has the opportunity to play ICRP sessions at will. Finally, he outputs a message m that will be used during the exchange targeted by the attack.

— Attack phase: The adversary can ask for $\mathcal{S}$ and $\mathcal{R}$ to initiate an ICRP execution for sending the message m. The transcripts $V_{\mathcal{S}}$ and $V_{\mathcal{R}}$ of this last exchange are not communicated to the adversary but are stored by the challenger. Finally the adversary outputs a tuple $V_{\mathcal{A}} = (S,R,m,s,r,\sigma)$ that should match the transcript $V_{\mathcal{S}} = (\mathcal{S},\mathcal{R},m,s_{\mathcal{S}},r_{\mathcal{S}},\sigma_{\mathcal{S}},T)$.
The adversary wins the game if:
  — $(s,r,\sigma) = (s_{\mathcal{S}},r_{\mathcal{S}},\sigma_{\mathcal{S}})$
  — (m,s,r) has not already been used in a previous ICRP interaction with node $\mathcal{R}$.
  — $\mathsf{Verify\_Sign}(\sigma,\mathcal{H}(m,s,r),pk_{\mathcal{R}}) = 1$
  — $\mathsf{Verify\_Time}(T) = 0$

Attacker strategies

As mentioned in Paragraph "Attacker's capabilities", the adversary can intercept messages at will, and therefore, modify them to try and trick the supervision of ICRP. This means that, during an ICRP execution, the transcripts $V_{\mathcal{S}}$ and $V_{\mathcal{R}}$ might be completely different. For instance the attacker could use a controlled node $\mathcal{I}$ to pre-play a protocol between $\mathcal{S}$ and $\mathcal{I}$ (impersonating the node $\mathcal{R}$) before using data received from $\mathcal{S}$ to play the protocol with $\mathcal{R}$ and obtain a signature. Note that the final verification of the game only concerns the transcript $V_{\mathcal{S}}$ because $\mathcal{S}$ is the node performing the verification.

4.2.2 Oracles

Oracles have access to and can modify 6 lists, described below. Whenever a game starts, all those lists are empty.

— *Part* is the list containing *id*s of non-corrupted participants along with their public key pk_{id}. An element *id* is formed of all the information needed to send messages to the targeted entity (typically, IP address and port number).
— *Corr* is the list containing tuples of the form (id, pk_{id}, sk_{id}) of the corrupted participants.
— *Dig* is the list containing couples of the form (m,h), with h being the output of $\mathcal{H}$ on input m.
— *S-View* is a list containing tuples of the form $(\mathcal{S},\mathcal{R},m,s,r,\sigma,T)$, with $\mathcal{S}$ and $\mathcal{R}$ being the *id*s of two nodes, σ being a binary string in the format of a signature

from the signature scheme Σ. The vectors $s = (s_1, \ldots, s_n)$ and $r = (r_1, \ldots, r_n)$ are two binary vectors of size n, m is a message, and T is a n-tuple of $\mathbb{R}_+^n$.
Each of these tuples represent an ICRP interaction between $\mathcal{S}$ and $\mathcal{R}$ according to what $\mathcal{S}$ did send, receive, and compute.

— *R-View* is a list containing tuples of the form $(\mathcal{S}, \mathcal{R}, m, s, r, \sigma)$, with $\mathcal{S}$ and $\mathcal{R}$ being the *id*s of two nodes, σ being a binary string in the format of a signature from the signature scheme Σ. The vectors $s = (s_1, \ldots, s_n)$ and $r = (r_1, \ldots, r_n)$ are two binary vectors of size n, and m is a message.
Each of these tuples represent an ICRP interaction between $\mathcal{S}$ and $\mathcal{R}$ according to what $\mathcal{R}$ did send, receive, and compute.

— *Chal* is a list that will ultimately contain only 2 tuples.
The first tuple is in the same format as the elements of *S-View*, i.e. $(\mathcal{S}, \mathcal{R}, m, s, r, \sigma, T)$, and corresponds to the ICRP interaction between $\mathcal{S}$ and $\mathcal{R}$ targeted by the attack, according to what $\mathcal{S}$ did send, receive, and compute.
The second tuple is in the same format as the elements of *R-View*, i.e. $(\mathcal{S}, \mathcal{R}, m, s, r, \sigma)$, and corresponds to the ICRP interaction between $\mathcal{S}$ and $\mathcal{R}$ targeted by the attack, according to what $\mathcal{R}$ did send, receive, and compute .

The definition of the oracles and the description of the games will often require the tuples in the lists *Part*, *Corr*, *S-View*, *R-View* and *Chall* to be compared to other tuples containing some, but not all, of their elements. The notion of "partial belonging" (see Definition 4.2.1) is introduced to ease the readability of this section.

Definition 4.2.1. ***Partial belonging***
Let L be a list of size n of tuples of size k:

$$L = ((l_{1,1}, \ldots, l_{1,k}), (l_{2,1}, \ldots, l_{2,k}), \ldots, (l_{n,1}, \ldots, l_{n,k}))$$

Let $n_0 \in \{1, \ldots, n\}$. Then, any tuple $t = (l_{n_0,i_1}, \ldots, l_{n_0,i_j})$ with $\{i_1, \ldots, i_j\}$ being a subset of $\{1, \ldots, k\}$ is said to partially belong to L, and noted $t \,\widetilde{\in}\, L$.

We now present and describe each oracle that may be used by an attacker in the security game:

— $\mathsf{create}(id)$: this oracle checks if id corresponds to an existing entity. If it does, the oracle generates credentials (sk_{id}, pk_{id}) for the input id, adds the couple (id, pk_{id}) to list *Part* and outputs 1. If it does not, the oracle outputs $\perp$.

— $\mathsf{corrupt}(id)$: this oracle checks if $id \,\widetilde{\in}\, Part$. If it does, the oracle adds the tuple (id, pk_{id}, sk_{id}) to list *Corr*, removes the entry corresponding to id from list *Part*, and returns the secret key sk_{id}. If it does not, the oracle outputs $\perp$.

— $\mathcal{H}(m)$: this oracle checks if input $m \,\widetilde{\in}\, Dig$. If it does, the oracle outputs the value h_m forming the couple (m, h_m) in *Dig*. If it does not, the oracle uniformly draws an integer $h \in [0, 2^\ell - 1]$, adds the couple (m, h) to list *Dig*, and outputs h.

— $\mathsf{play}(\mathcal{S}, \mathcal{R}, m)$: this oracle checks if $\mathcal{S} \,\widetilde{\in}\, Part$ and $\mathcal{R} \,\widetilde{\in}\, Part$. If they do not, the oracle outputs $\perp$. If they do, the oracle runs the protocol between $\mathcal{S}$ as the sending party and $\mathcal{R}$ as the receiving party.
It then adds the tuple $V_{\mathcal{S}} = (\mathcal{S}, \mathcal{R}, m_{\mathcal{S}}, s_{\mathcal{S}}, r_{\mathcal{S}}, \sigma_{\mathcal{S}}, T)$ to list $S\text{-}View$, and the tuple $V_{\mathcal{R}} = (\mathcal{S}, \mathcal{R}, m_{\mathcal{R}}, s_{\mathcal{R}}, r_{\mathcal{R}}, \sigma_{\mathcal{R}})$ to list $R\text{-}View$. In these tuples, $m_{\mathcal{S}}$ is the message sent by $\mathcal{S}$, $s_{\mathcal{S}}$ (resp. $r_{\mathcal{S}}$) is the binary vector sent by $\mathcal{S}$ (resp. received by $\mathcal{S}$), $\sigma_{\mathcal{S}}$ is the signature received by $\mathcal{S}$, and T is the sample of RTT collected by $\mathcal{S}$ during the exchange. Similarly, $m_{\mathcal{R}}$ is the message received by $\mathcal{R}$, $s_{\mathcal{R}}$ (resp. $r_{\mathcal{R}}$) is the binary vector received by $\mathcal{R}$ (resp. sent by $\mathcal{R}$), and $\sigma_{\mathcal{R}}$ is the signature computed by $\mathcal{R}$. Finally, the oracle outputs the tuples $V_{\mathcal{S}}$ and $V_{\mathcal{R}}$.

— $\mathsf{play\_corr\_S}(\hat{\mathcal{S}}, \mathcal{R}, m)$: this oracle checks if $\hat{\mathcal{S}} \,\widetilde{\in}\, Corr$ and $\mathcal{R} \,\widetilde{\in}\, Part$. If they do not, the oracle outputs $\perp$. If they do, the oracle runs the protocol between $\hat{\mathcal{S}}$ as the sending party and $\mathcal{R}$ as the receiving party, with the bits s_i drawn from an unknown distribution Ψ.
It then adds the tuple $V_{\hat{\mathcal{S}}} = (\hat{\mathcal{S}}, \mathcal{R}, m_{\hat{\mathcal{S}}}, s_{\hat{\mathcal{S}}}, r_{\hat{\mathcal{S}}}, \sigma_{\hat{\mathcal{S}}}, T))$ to list $S\text{-}View$, and the tuple $V_{\mathcal{R}} = (\hat{\mathcal{S}}, \mathcal{R}, m_{\mathcal{R}}, s_{\mathcal{R}}, r_{\mathcal{R}}, \sigma_{\mathcal{R}})$ to list $R\text{-}View$. In these tuples, $m_{\hat{\mathcal{S}}}$ is the message sent by $\hat{\mathcal{S}}$, $s_{\hat{\mathcal{S}}}$ (resp. $r_{\hat{\mathcal{S}}}$) is the binary vector sent by $\hat{\mathcal{S}}$ (resp. received by $\hat{\mathcal{S}}$), $\sigma_{\hat{\mathcal{S}}}$ is the signature received by $\hat{\mathcal{S}}$, and T is the sample of RTT collected by $\hat{\mathcal{S}}$ during the exchange. Similarly, $m_{\mathcal{R}}$ is the message received by $\mathcal{R}$, $s_{\mathcal{R}}$ (resp. $r_{\mathcal{R}}$) is the binary vector received by $\mathcal{R}$ (resp. sent by $\mathcal{R}$), and $\sigma_{\mathcal{R}}$ is the signature computed by $\mathcal{R}$. Finally, the oracle outputs the tuples $V_{\hat{\mathcal{S}}}$ and $V_{\mathcal{R}}$.

— $\mathsf{play\_corr\_R}(\mathcal{S}, \hat{\mathcal{R}}, m)$: this oracle checks if $\mathcal{S} \,\widetilde{\in}\, Part$ and $\hat{\mathcal{R}} \,\widetilde{\in}\, Corr$. If they do not, the oracle outputs $\perp$. If they do, the oracle runs the protocol between $\mathcal{S}$ as the sending party and $\hat{\mathcal{R}}$ as the receiving party, with the bits r_i drawn from an unknown distribution Ψ.
It then adds the tuple $V_{\mathcal{S}} = (\mathcal{S}, \hat{\mathcal{R}}, m_{\mathcal{S}}, s_{\mathcal{S}}, r_{\mathcal{S}}, \sigma_{\mathcal{S}}, T))$ to list $S\text{-}View$, and the tuple $V_{\hat{\mathcal{R}}} = (\mathcal{S}, \hat{\mathcal{R}}, m_{\hat{\mathcal{R}}}, s_{\hat{\mathcal{R}}}, r_{\hat{\mathcal{R}}}, \sigma_{\hat{\mathcal{R}}})$ to list $R\text{-}View$. In these tuples, $m_{\mathcal{S}}$ is the message sent by $\mathcal{S}$, $s_{\mathcal{S}}$ (resp. $r_{\mathcal{S}}$) is the binary vector sent by $\mathcal{S}$ (resp. received by $\mathcal{S}$), $\sigma_{\mathcal{S}}$ is the signature received by $\mathcal{S}$, and T is the sample of RTT collected by $\mathcal{S}$ during the exchange. Similarly, $m_{\hat{\mathcal{R}}}$ is the message received by $\hat{\mathcal{R}}$, $s_{\hat{\mathcal{R}}}$ (resp. $r_{\hat{\mathcal{R}}}$) is the binary vector received by $\hat{\mathcal{R}}$ (resp. sent by $\hat{\mathcal{R}}$), and $\sigma_{\hat{\mathcal{R}}}$ is the signature computed by $\hat{\mathcal{R}}$. Finally, the oracle outputs the tuples $V_{\mathcal{S}}$ and $V_{\hat{\mathcal{R}}}$.

— $\mathsf{play\_chall}(\mathcal{S}, \mathcal{R}, m)$: This oracle can only be called once. it checks if $\mathcal{S} \,\widetilde{\in}\, Part$ and $\mathcal{R} \,\widetilde{\in}\, Part$. If they do not, the oracle outputs $\perp$. If they do, the oracle runs the protocol between $\mathcal{S}$ as the sending party and $\mathcal{R}$ as the receiving party.
It then adds the tuples $V_{\mathcal{S}} = (\mathcal{S}, \mathcal{R}, m_{\mathcal{S}}, s_{\mathcal{S}}, r_{\mathcal{S}}, \sigma_{\mathcal{S}}, T)$ and $V_{\mathcal{R}} = (\mathcal{S}, \mathcal{R}, m_{\mathcal{R}}, s_{\mathcal{R}}, r_{\mathcal{R}}, \sigma_{\mathcal{R}})$ to list $Chal$. In these tuples, $m_{\mathcal{S}}$ is the message sent by $\mathcal{S}$, $s_{\mathcal{S}}$ (resp. $r_{\mathcal{S}}$) is the binary vector sent by $\mathcal{S}$ (resp. received by $\mathcal{S}$), $\sigma_{\mathcal{S}}$ is the signature received by $\mathcal{S}$, and T is the sample of RTT collected by $\mathcal{S}$ during the exchange. Similarly, $m_{\mathcal{R}}$ is the message received by $\mathcal{R}$, $s_{\mathcal{R}}$ (resp. $r_{\mathcal{R}}$) is the binary vector received by $\mathcal{R}$ (resp. sent by $\mathcal{R}$), and $\sigma_{\mathcal{R}}$ is the signature computed by $\mathcal{R}$. Then, the oracle outputs 1.

— $\mathsf{intercept}(\mathcal{S}, \mathcal{R}, \mathcal{I}, k)$: this oracle changes the route between $\mathcal{S}$ and $\mathcal{R}$ so that $\mathcal{I}$

becomes an intermediate node for the k next messages sent from $\mathcal{S}$ to $\mathcal{R}$. The messages stop upon reaching node $\mathcal{I}$, which can whether forward them to $\mathcal{R}$, drop them, or store them the later uses.

These oracles are gathered in sets to ease readability. Table 4.1 describes the 3 defined sets $\mathcal{O}$.set, $\mathcal{O}$.act, and $\mathcal{O}$.chall.

Table 4.1 – Three sets of Oracles

	$\mathcal{O}$.set	$\mathcal{O}$.act	$\mathcal{O}$.chall
create	$\surd$	$\times$	$\times$
corrupt	$\surd$	$\times$	$\times$
$\mathcal{H}$	$\surd$	$\surd$	$\surd$
play	$\surd$	$\surd$	$\times$
play_corr_S	$\surd$	$\surd$	$\times$
play_corr_R	$\surd$	$\surd$	$\times$
play_chall	$\times$	$\times$	$\surd$
intercept	$\surd$	$\surd$	$\surd$

4.2.3 The Relay Game

The Relay Game represents the attack in normal conditions. The adversary $\mathcal{A}$ attempts to perform a relay on a full session without being detected. Hence, the attacker goal is to return the content $(\mathcal{S}, \mathcal{R}, m, s, r, \sigma)$ of a complete exchange accepted by the protocol.
During the attack, $\mathcal{A}$ is provided 3 sets of oracles, $\mathcal{O}$.set, $\mathcal{O}$.act, and $\mathcal{O}$.chall allowing to perform specific actions. The oracles and the sets $\mathcal{O}$.set, $\mathcal{O}$.act, and $\mathcal{O}$.chall are described in Section 4.2.2.
The game let $\mathcal{A}$ perform a learning phase in which he has access to the set of oracles $\mathcal{O}$.set and has to designate two victim nodes $\mathcal{S}$ and $\mathcal{R}$. The victim nodes should not be corrupted, otherwise the game aborts. The adversary then keeps experimenting at will, using the set $\mathcal{O}$.act. When satisfied with this experimenting phase, $\mathcal{A}$ proceeds with the attack phase and chooses a message m.
During the attack phase, the attacker has access to the set $\mathcal{O}$.chall and must output a tuple $(\mathcal{S}, \mathcal{R}, m, s, r, \sigma)$. This tuple must satisfy 4 conditions:

1. The triple (m, s, r) is not associated with the designated victim receiver $\mathcal{R}$ in any entry of list $View_{\mathcal{S}}$.
2. There exists $T \in \mathbb{R}_+^n$ such that:
$$(\mathcal{S}, \mathcal{R}, m, s, r, \sigma, T) \in Chal$$
3. Verify_Sign$(\sigma, \mathcal{H}(m, s, r), pk_{\mathcal{R}})$ outputs 1 (i.e. the signature is valid).
4. Verify_Time(T) outputs 0 (i.e. the sample of RTT is accepted).

Condition (1) states that the message and bits exchanged during the attack phase have not been already seen in a previous exchange involving $\mathcal{R}$, condition (2) states that the exchange returned by the attacker has occurred during the attack phase, and condition (3) and (4) states that the exchange returned by the attacker has been accepted by the protocol.
If those 4 conditions are satisfied, $\mathcal{A}$ wins the game, and the experience outputs a bit $b = 1$. In any other case, $\mathcal{A}$ looses the game and the experience outputs 0. In the following, the Relay Game is indexed as Game 0.
Game 0 is formally described in figure 4.3.

Relay Game: $\mathsf{Exp}^{relay}_{\mathcal{H},\Sigma,\mathcal{A}}$

Learning Phase
$(\mathcal{S}, \mathcal{R}) \leftarrow \mathcal{A}^{\mathcal{O}.\mathsf{set}}()$
if $\mathcal{S} \in Part$ AND $\mathcal{R} \in Part$:
  $m \leftarrow \mathcal{A}^{\mathcal{O}.\mathsf{act}}(\mathcal{S}, \mathcal{R})$
else :
  return $\perp$
Attack phase
$(\mathcal{S}, \mathcal{R}, m, s, r, \sigma) \leftarrow \mathcal{A}^{\mathcal{O}.\mathsf{chall}}(\mathcal{S}, \mathcal{R}, m)$
if $(\mathcal{R}, m, s, r) \,\widetilde{\in}\, R\text{-}View$:
  return $\perp$
if $\exists T \in \mathbb{R}^n_+$, $(\mathcal{S}, \mathcal{R}, m, s, r, \sigma, T) \in Chal$:
  $b_t \leftarrow \mathsf{Verify\_Time}(T)$
  $b_s \leftarrow \mathsf{Verify\_Sign}(\sigma, \mathcal{H}(m, s, r), pk_{\mathcal{R}})$
  $b \leftarrow \overline{b_t} \cdot b_s$
  return b
return $\perp$

Figure 4.3 – Game for relay attack on protocol ICRP with hash function $\mathcal{H}$ and signature scheme Σ

4.3 Proof

This section will prove the following theorem:

Theorem 4.3.1. ***Security of ICRP against adversaries in the ROM***
Let n be the number of rounds for an execution of ICRP, ℓ be the length of the output returned by the random oracle $\mathcal{H}$, and Σ be a euf-cma *secure signature scheme in a parameter λ.*
For any adversary $\mathcal{A}$ that makes at most q_{play} calls to oracles play *and* play_corr_S*, in a setup of at most q_{crea} participant, the advantage for $\mathcal{A}$ to win the game $\mathsf{Exp}^{relay}_{\mathcal{H},\Sigma,\mathcal{A}}$ (i.e. Game 0) is such that:*

$$Pr\left(\mathsf{Exp}^{relay}_{\mathcal{H},\Sigma,\mathcal{A}} = 1\right) \leq q_{crea}\left(\frac{1}{2^n} + \frac{q_{play}+1}{2^\ell} + \mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathcal{A},\Sigma}(\lambda)\right)$$

4.3.1 Games and Transitions

In the sequel, we describe 4 security games indexed from 1 to 4. The relay game described in Section 4.2.3 is designated as Game 0. For all $i \in [0,4]$, let G_i be the event "Game i outputs 1" and let p_i be the probability $Pr(G_i)$.

Game 1

Description.
Let q_{crea} be the expected maximum number of query to oracle create made by the attacker. Game 1 is identical to Game 0 except that, at the beginning of the learning phase, an integer i is drawn uniformly in $\{1,\ldots,q_{crea}\}$. Let $\mathcal{R}_0$ be the entity created by the i^{th} query to oracle create. Once $\mathcal{A}$ has returned his targeted couple $(\mathcal{S},\mathcal{R})$, the game aborts if $\mathcal{R} \neq \mathcal{R}_0$.
Transition from Game 0 to 1.
Let $F_{0,1}$ be the event:

$$F_{0,1} : \ \mathcal{R}_0 = \mathcal{R}$$

The event $F_{0,1}$ occurs whenever $\mathcal{A}$ designates $\mathcal{R}_0$ as the victim receiver node.
Regardless, of the attacker strategy for the choice of the victim, the index of $\mathcal{R}$ in the list of all created entities will be lesser than q_{crea}. Also, the choice of $\mathcal{R}_0$ is done uniformly in a list of size q_{crea}. Therefore:

$$Pr(F_{0,1}) = \frac{1}{q_{crea}}$$

Moreover, observe that the event G_1: "$\mathcal{A}$ wins Game 1" is exactly equivalent to the event

$G_0 \wedge F_{0,1}$: "$\mathcal{A}$ wins Game 0 AND $\mathcal{R}_0 = \mathcal{R}$".

$$\begin{array}{ccc} G_0 \wedge F_{0,1} & \iff & G_1 \\ Pr(G_0) \cdot Pr(F_{0,1}) & = & Pr(G_1) \\ Pr(G_0) & = & \frac{Pr(G_1)}{Pr(F_{0,1})} \end{array}$$

Consequently:

$$p_0 = q_{crea} \cdot p_1 \tag{1}$$

Game 2

Description.
Game 2 is identical to Game 1, except that, whenever the tuple $(\mathcal{S}, \mathcal{R}, m, s, r, \sigma)$, returned by the attacker during the attack phase, is such that the triple (m, s, r) collides on $\mathcal{H}$ with another triple (m', s', r') partially belonging to $R\text{-}View$ and obtained from a previous exchange with $\mathcal{R}$, the game aborts. That is to say:

$$\exists (\mathcal{R}, m', s', r') \widetilde{\in} \ R\text{-}View, \text{ such that } (m', s', r') \neq (m, s, r) \text{ AND } \mathcal{H}(m', s', r') = \mathcal{H}(m, s, r)$$

Transition from Game 1 *to* 2.
From each query to oracles play and play_corr_S with entity $\mathcal{R}$ as the receiving party, the attacker obtains a tuple $(\mathcal{R}, m', s', r')$ from which he can query $\mathcal{H}(m', s', r')$ to the random oracle $\mathcal{H}$. Let H be the set of all the $\mathcal{H}(m', s', r')$, and q_{play} be the number of calls made to the random oracle $\mathcal{H}$ throughout the game.

Let $(\mathcal{S}, \mathcal{R}, m, s, r, \sigma)$ be the tuple returned by the attacker after the call of oracle play_chall.

Let then $F_{1,2}$ be the event:

$$F_{1,2} : \ \exists h' = \mathcal{H}(m', s', r') \in H, (m', s', r') \neq (m, s, r) \text{ AND } \mathcal{H}(m', s', r') = \mathcal{H}(m, s, r)$$

Let $(m', s', r')^{(i)}$ be the triple obtained from the i^{th} oracle call and let $F_{1,2}^{(i)}$ be the event:

$$F_{1,2}^{(i)} : \ (m', s', r')^{(i)} \neq (m, s, r) \text{ AND } \mathcal{H}\left((m', s', r')^{(i)}\right) = \mathcal{H}(m, s, r)$$

Then:

$$F_{1,2} \iff \bigvee_{i=1}^{q_{play}} F_{1,2}^{(i)} \tag{2.1}$$

Let us define for all i in $\{1, \ldots, q_{play}\}$ the events:

$$\begin{array}{ll} A^{(i)} : & \mathcal{H}\left((m', s', r')^{(i)}\right) = \mathcal{H}(m, s, r) \\ B^{(i)} : & (m', s', r')^{(i)} \neq (m, s, r) \end{array}$$

Hence, for all i in $\{1, \ldots, q_{play}\}$, $F_{1,2}^{(i)} = A^{(i)} \wedge B^{(i)}$, and:

$$\begin{aligned} Pr(F_{1,2,}^{(i)}) &= Pr(A^{(i)} \wedge B^{(i)}) \\ &= Pr(B^{(i)}) \cdot Pr(A^{(i)}|B^{(i)}) \\ &\leq Pr(A^{(i)}|B^{(i)}) \end{aligned}$$

The random oracle $\mathcal{H}$ outputs an integer in $[0, 2^\ell - 1]$ from the uniform distribution. Hence:

$$Pr(A^{(i)}|B^{(i)}) = \frac{1}{2^\ell}$$

Consequently:

$$\forall i \in [1, \ldots, q_{play}],\ Pr(F_{1,2}^{(i)}) \leq \frac{1}{2^\ell} \tag{2.2}$$

From statements 2.1, 2.2 and the Union bound Lemma (lemma 4.1.2), it is the true that:

$$\begin{aligned} Pr(F_{1,2}) &\leq \textstyle\sum_{i=1}^{q_{play}} Pr(F_{1,2}^{(i)}) \\ &\leq \frac{q_{play}}{2^\ell} \end{aligned} \tag{2.3}$$

Now, observe that Game 1 and Game 2 are strictly indistinguishable if event $F_{1,2}$ does not occur. Hence:

$$Pr(G_1 \wedge \neg F_{1,2}) = Pr(G_2 \wedge \neg F_{1,2})$$

Consequently, according to statement 2.3 and the Difference Lemma (lemma 4.1.1), the next inequality follows:

$$|p_2 - p_1| \leq \frac{q_{play}}{2^\ell} \tag{2}$$

Game 3

Description.
Let $(\mathcal{S}, \mathcal{R}, m, s, r, \sigma)$ be the tuple return by the attacker during the attack.
Let $V_{\mathcal{S}} = (\mathcal{S}, \mathcal{R}, m_{\mathcal{S}}, s_{\mathcal{S}}, r_{\mathcal{S}}, \sigma_{\mathcal{S}}, T)$ and $V_{\mathcal{R}} = (\mathcal{S}, \mathcal{R}, m_{\mathcal{R}}, s_{\mathcal{R}}, r_{\mathcal{R}}, \sigma_{\mathcal{R}})$ be the tuples stored in list *Chal* after the query on oracle play_chal.
Game 3 is identical to Game 2 except that, if $r = r_{\mathcal{R}}$ and Verify_Time$(T) = 0$, the game aborts.
Transition from Game 2 *to* 3.
Let $F_{2,3}$ be the event:

$$F_{2,3} :\ r = r_{\mathcal{R}} \text{ AND Verify\_Time}(T) = 0$$

Given that $\mathcal{R}$ is necessarily a non-corrupted participant, it is true that $r_{\mathcal{R}}$ has been drawn from the uniform distribution. Moreover, because Verify_Time$(T) = 0$, and from Assumption 1, the adversary could not have observed the vector $r_{\mathcal{R}}$ by relaying messages between $\mathcal{S}$ and $\mathcal{R}$. Therefore, $\mathcal{A}$ must guess in advance each bit of the binary vector r in advance.

This guess succeeds with a probability of $\frac{1}{2}$ for each bit. It is then trivial that:

$$Pr(F_{2,3}) = \frac{1}{2^n} \tag{3.1}$$

Furthermore, because Game 2 and 3 are indistinguishable if event $F_{2,3}$ does not occur, it is true that:

$$Pr(G_2 \wedge \neg F_{2,3}) = Pr(G_3 \wedge \neg F_{2,3})$$

Consequently, according to statement 3.1 and the Difference Lemma 4.1.1, the next inequality follows:

$$|p_3 - p_2| \leq \frac{1}{2^n} \tag{3}$$

Game 4

Description.
Let $(\mathcal{S}, \mathcal{R}, m, s, r, \sigma)$ be the tuple return by the attacker during the attack.

Let $V_{\mathcal{S}} = (\mathcal{S}, \mathcal{R}, m_{\mathcal{S}}, s_{\mathcal{S}}, r_{\mathcal{S}}, \sigma_{\mathcal{S}}, T)$ and $V_{\mathcal{R}} = (\mathcal{S}, \mathcal{R}, m_{\mathcal{R}}, s_{\mathcal{R}}, r_{\mathcal{R}}, \sigma_{\mathcal{R}})$ be the tuples stored in list *Chal* after the query on oracle play_chal.

Game 4 is identical to Game 3 except that, if $r \neq r_{\mathcal{R}}$ and $\mathcal{H}(m_{\mathcal{S}}, s_{\mathcal{S}}, r_{\mathcal{R}}) = \mathcal{H}(m_{\mathcal{S}}, s_{\mathcal{S}}, r)$, the game aborts.

Transition from Game 3 *to* 4.
Let $F_{3,4}$ be the event:

$$F_{3,4} : \ r \neq r_{\mathcal{R}} \text{ AND } \mathcal{H}(m_{\mathcal{S}}, s_{\mathcal{S}}, r_{\mathcal{R}}) = \mathcal{H}(m_{\mathcal{S}}, s_{\mathcal{S}}, r)$$

Given that $r \neq r_{\mathcal{R}}$, and because the random oracle $\mathcal{H}$ outputs an integer in $[0, 2^{\ell} - 1]$ from the uniform distribution. It is trivial that:

$$Pr(F_{3,4}) = \frac{1}{2^{\ell}} \tag{4.1}$$

Furthermore, because Game 3 and 4 are indistinguishable if event $F_{3,4}$ does not occur, it is true that:

$$Pr(G_3 \wedge \neg F_{3,4}) = Pr(G_4 \wedge \neg F_{3,4})$$

Consequently, according to statement 3.1 and the Difference Lemma 4.1.1, the next inequality follows:

$$|p_4 - p_3| \leq \frac{1}{2^{\ell}} \tag{4}$$

4.3.2 Final part of the proof

It can now be observed that:

$$\begin{array}{lcll} |p_4 - p_1| & \leq & |p_4 - p_3| + |p_3 - p_2| + |p_2 - p_1| & \text{(from the triangular inequality)} \\ \left|p_4 - \frac{p_0}{q_{crea}}\right| & \leq & \frac{1}{2^n} + \frac{q_{play}+1}{2^\ell} & \text{(from results 1, 2, 3, and 4)} \end{array}$$

This implies that:

$$\begin{array}{lcl} \frac{p_0}{q_{crea}} - p_4 & \leq & \frac{1}{2^n} + \frac{q_{play}+1}{2^\ell} \\ p_0 & \leq & q_{crea}\left(\frac{1}{2^n} + \frac{q_{play}+1}{2^\ell} + p_4\right) \end{array} \tag{5}$$

The proof of theorem 4.3.1 finds its conclusion in the demonstration of the following lemma:

Lemma 4.3.1. *Let Σ be the signature scheme used for every ICRP execution in Game 4. Assume that Σ is* euf-cma *secure.*
Then:

$$Pr(G_4) \leq \mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathcal{A},\Sigma}(\lambda)$$

Proof. Let $\mathcal{A}$ be an attacker for Game 4. Let $\mathcal{B}$ be an attacker for the euf-cma game (see Section 4.1.2) on the signature scheme Σ. $\mathcal{B}$ can act as the challenger interacting with $\mathcal{A}$ on Game 4, trying to obtain a forgery from $\mathcal{A}$ and win the euf-cma game by following the procedure described below:
$\mathcal{B}$ initiate the euf-cma game, is given a public key pk_0, has access to an oracle $\mathcal{O}$Sign which, on an input message M, returns a valid signature σ on M for the public key pk_0 and stores the couple (M, σ) in a list *Req*. $\mathcal{B}$'s goal is to provide a couple (M_0, σ_0) such that Verify_Sign$(\sigma_0, M_0, pk_0) = 1$ and M_0 has not already been queried to $\mathcal{O}$Sign.
To do so, $\mathcal{B}$ ask for $\mathcal{A}$ to initiate Game 4. During $\mathcal{A}$'s learning phase, the selected entity $\mathcal{R}_0$ (see the description of Game 1) is distributed the public key pk_0 instead of receiving a classical key pair. For the rest of the learning phase, every query to oracles play and play_corr_S with input $\mathcal{R}_0$ as the receiving party is executed as usual, except that the signature $\sigma_{\mathcal{R}}$, stored in *R-View*, is obtained from a query to $\mathcal{O}$Sign on input message $\mathcal{H}(m_{\mathcal{R}}, s_{\mathcal{R}}, r_{\mathcal{R}})$.
Assume that:

$$\mathcal{A} \text{ wins Game 4 with the tuple } (\mathcal{S}, \mathcal{R}_0, m, s, r, \sigma_0) \qquad \text{(assumption a)}$$

Therefore, $\mathcal{B}$ obtains $M_0 = \mathcal{H}(m, s, r)$ and σ_0 such that Verify_Sign$(\sigma_0, M_0, pk_0) = 1$.
We now describe the following notation for clarity's sake:

- During the i^{th} call to one of the oracles play or play_corr_S, two tuples $V_{\mathcal{S},i}$ and $V_{\mathcal{R},i}$ are respectively stored in list *S-View* and *R-View*.
  - The elements in the tuples $V_{\mathcal{S},i}$ are denoted $m_{\mathcal{S},i}, s_{\mathcal{S},i}, r_{\mathcal{S},i}$ and $\sigma_{\mathcal{S},i}$ and represent the elements sent and received by $\mathcal{S}$ during the corresponding execution of

ICRP. T_i is the list of RTT computed by $\mathcal{S}$.
  - The elements in the tuples $V_{\mathcal{R},i}$ are denoted $m_{\mathcal{R},i}, s_{\mathcal{R},i}, r_{\mathcal{R},i}$ and $\sigma_{\mathcal{R},i}$ and represent the elements sent and received by $\mathcal{R}$ during the corresponding execution of ICRP.
- During the unique call to oracle play_chall, two tuples $V_{\mathcal{S}}$ and $V_{\mathcal{R}}$ are stored in list $Chal$.
  - The elements in the tuple $V_{\mathcal{S}}$ are denoted $m_{\mathcal{S}}, s_{\mathcal{S}}, r_{\mathcal{S}}$ and $\sigma_{\mathcal{S}}$ and represent the elements sent and received by $\mathcal{S}$ during the corresponding execution of ICRP. T is the list of RTT computed by $\mathcal{S}$.
  - The elements in the tuple $V_{\mathcal{R}}$ are denoted $m_{\mathcal{R}}, s_{\mathcal{R}}, r_{\mathcal{R}}$ and $\sigma_{\mathcal{R}}$ and represent the elements sent and received by $\mathcal{R}$ during the corresponding execution of ICRP.

Now, $\mathcal{B}$ wins the euf-cma game if and only if $(M_0, \sigma_0) \notin Req$. That is to say, if $\mathcal{O}$Sign has never been queried on $M_0 = \mathcal{H}(m, s, r)$.
All along the interaction between $\mathcal{A}$ and $\mathcal{B}$, each call on $\mathcal{O}$Sign is only made when one of the oracles play, play_corr_S, or play_chall is queried.
Assume that:

$$(M_0, \sigma_0) \in Req \qquad \text{(assumption b)}$$

This means that the oracle $\mathcal{O}$Sign has been call on M_0 in a previous exchange whether from play or play_corr_S in the learning phase, or from play_chall in the attack phase, for an interaction with party $\mathcal{R}_0$ as the receiver.
Consequently, at least one of these queries provided either the exact same tuple (m, s, r) or another tuple colliding on $\mathcal{H}$ with (m, s, r).

Case 1. The query that has provided the right tuple is the i^{th} call to one of the oracles play or play_corr_S.
In that case, the oracle $\mathcal{O}$Sign has been called on $(m_{\mathcal{R},i}, s_{\mathcal{R},i}, r_{\mathcal{R},i})$. Hence, one of the two following statements is true:

1. $(m_{\mathcal{R},i}, s_{\mathcal{R},i}, r_{\mathcal{R},i}) = (m, s, r)$
2. $(m_{\mathcal{R},i}, s_{\mathcal{R},i}, r_{\mathcal{R},i}) \neq (m, s, r)$ AND $\mathcal{H}(m_{\mathcal{R},i}, s_{\mathcal{R},i}, r_{\mathcal{R},i}) = \mathcal{H}(m, s, r)$

If statement 1 is true, then $(\mathcal{R}_0, m_{\mathcal{R},i}, s_{\mathcal{R},i}, r_{\mathcal{R},i}) \,\widetilde{\in}\, R\text{-}View$ which aborts the game (see the description of Game 0). Therefore $\mathcal{A}$ does not win Game 4, and this contradicts the assumption a.
Moreover, if statement 2 is true, then the game aborts because this is the exact failure condition described in Game 2. Therefore $\mathcal{A}$ does not win Game 4, and this contradicts the assumption a.

Case 2. The query that has provided the right tuple is the call on play_chall.
In that case, the oracle $\mathcal{O}$Sign has been called on $(m_{\mathcal{R}}, s_{\mathcal{R}}, r_{\mathcal{R}})$. Hence, one of the two following statements is true:

1. $(m_{\mathcal{R}}, s_{\mathcal{R}}, r_{\mathcal{R}}) = (m, s, r)$

2. $(m_{\mathcal{R}}, s_{\mathcal{R}}, r_{\mathcal{R}}) \neq (m, s, r)$ AND $\mathcal{H}(m_{\mathcal{R}}, s_{\mathcal{R}}, r_{\mathcal{R}}) = \mathcal{H}(m, s, r)$

If statement 1 is true, then we have $r = r_{\mathcal{R}}$ and $\mathsf{Verify\_Time}(T) = 0$ (from the assumption a, $\mathcal{A}$ wins the game, so the times must be valid) and the game aborts because this is the exact failure condition described in Game 3. Therefore $\mathcal{A}$ does not win Game 4, and this contradicts the assumption a.
Moreover, if statement 2 is true, then we have $r \neq r_{\mathcal{R}}$ and $\mathcal{H}(m_{\mathcal{R}}, s_{\mathcal{R}}, r_{\mathcal{R}}) = \mathcal{H}(m, s, r)$ and the game aborts because this is the exact failure condition described in Game 4. Therefore $\mathcal{A}$ does not win Game 4, and this contradicts the assumption a.

To conclude. If $\mathcal{A}$ wins Game 4 (i.e. assumption a), then, the couple (M_0, σ_0) returned by $\mathcal{A}$ has not been obtained through the call of the oracle $\mathcal{O}\mathsf{Sign}$. This contradicts assumption b.
Consequently, in order to win Game 4, $\mathcal{A}$ necessarily needs to forge a valid signature on $\mathcal{H}(m, s, r)$.
Hence:

$$Pr(G_4) \leq \mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathcal{A},\Sigma}(\lambda)$$

□

From lemma 4.3.1 and statement 5, the claim of theorem 4.3.1 immediately follows:

$$Pr\left(\mathsf{Exp}^{relay}_{\mathcal{H},\Sigma,\mathcal{A}} = 1\right) \leq q_{crea}\left(\frac{1}{2^n} + \frac{q_{play}+1}{2^\ell} + \mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{\mathcal{A},\Sigma}(\lambda)\right)$$

4.4 Conclusion

In this chapter, we have provided a proof for ICRP security against relay attacks. This proof relies on 2 main assumptions. The first one is implicit from the use of the Random Oracle Model (ROM). In that model, the cryptographic hash function are assumed to return perfectly random outputs, meaning that the security is proven as long as the practical hash function used in the protocol is deemed robust.
The second assumption is that the decision function is a black-box that perfectly distinguish between a genuine time sample and a time sample issued from a relay. If this assumption might seems quite strong, it is actually acceptable in attacks where the impact of the relay is important. As Chapters 2 and 3 showed it, a decision function achieving a perfect detection accuracy exists for the distances involved in our experiments. In real case scenarios, a relay attack occurs mainly on international scale, implying high RTT impact.
To conclude, the proof of Theorem 4.3.1 actually demonstrates that the practicality of the protocol is ensured as long as the protocol uses efficient and secure primitives. Future works should mainly focus on the improvement of the decision function capacity, i.e. being able to detect smaller differences from the expected times, highlight statistical properties of an RTT sample, or finding a secure way to dynamically update the reference samples without human intervention.

Conclusion

"What is this brief, mortal life...
If not the pursuit of legacy."
- Corlys Velaryion

This thesis aimed to efficiently detect relay attacks over the Internet. The ICRP protocol, built from scratch during these 3 years achieves this goal while offering some very desirable properties:

1. efficiency: it requires only two cryptographic operations per execution inducing negligible workload for users and very few loss of throughput.
2. network-friendliness: the operations are only performed by the end-points, adding zero computational charge for the routers, and the overhead volume of transiting data remains only about 1.5%.
3. usability: no software or harware update is required for any intermediary nodes.
4. independence: no software or hardware updates on the routing process or equipment would impact the protocol, as long as the RTT remain stable.
5. security: the protocol is proven secure under reasonable assumptions.

We strongly believe our relay detection method to be a satisfying substitute, until an efficient countermeasures gets widely adopted. However, we also raise many interesting follow up questions for future work contents, as detailed below.

TCP-based ICRP. The measurements performed throughout the tests conducted during this thesis used UDP packets. UDP is very convenient for computing RTTs because neither the sender of the packet nor the corresponding receiver performs any additional action regarding integrity or delivery confirmation. This means that the measurement performed with UDP strictly encapsulates the time for the packet to reach the receiver added to the time the response goes back to the sender.
However, in practical cases, and especially for sensitive data, TCP is a much more usual protocol. TCP is already built so that, once a packet is received, an immediate acknowledgement is sent back to the sender for delivery confirmation. An interesting follow up research to conduct would be to see if this acknowledgement can be used to compute RTTs as stable as the ones computed from UDP packets. Moreover, the TCP header has a few optional empty field reserved for future use. One of these fields could be filled with the random bits sent during the fast-bit exchange phase.

Optimal setups for relay attacks. The experimental relays presented in Chapter 3 have been performed on an international scale, for either short, medium and long range distances separating the nodes. These experiments did show a critical impact on RTTs for every test. However, the experimental setup was limited to a few nodes.
We believe that even wider experiments using more nodes should be made to learn more about the real efficiency a relay attack can get. In particular, two interesting unanswered questions are:

— What is the impact on RTT if the relaying node is placed as close as possible to one of the party executing ICRP (i.e. in the same AS) ?
— How does this impact evolves in relation to the number of nodes separating the relay to the end-points ?
— Are there other impacting factors ?

Resilience to optimal setups. The decision function we define in this manuscript has shown to be efficient on the set of measures at our disposal. We believe our experiments to be representative of a real relay attack scenario. However, we can never completely reject the possibility of an optimal attacker, i.e. capable of relaying with very little time impact. Our decision function uses a positional argument to decide if a sample is suspicious or not. Consequently, a hypothetically very efficient relay could completely trick the decision. Future works should then look for other decision arguments, maybe using the Grubb statistical law as authors did in [36], or machine learning techniques.

Complete implementation. The prototype implementation mentionned in Chapter 3 is still very limited and needs to be improved. So far the prototype only sends random strings of ASCII characters encapsulated in UDP packets. The user can control the packet size, the number of packets per ICRP sessions, the total number of sessions to execute, and the time the sender should wait before sending the next packet. A more advanced version of this prototype should send complete files over multiples sessions of ICRP in active mode, and run passively in the background for the supervision of isolated packet.

Dynamic reference sample's update. As shown in Chapter 2, some slight changes can occur on the measured sample. This has been the case for 2 samples between the same nodes, 3 months apart. These rare events must be taken into account by updating the reference sample when they occur. As it was briefly mentioned in Section 3.2.1, this updating process could be automated by regularly refreshing the reference sample with the last few accepted samples. But, without any human supervision on this dynamic update, an attacker could attempt to slowly poison the reference sample by progressively adding small delays until being able to perform a relay without being detected.
For instance, by relaying only a small proportion of the packets involved in an ICRP session, our current decision function will not reject the sample and will consider the RTT related to the relayed packets to be natural outliers. By repeating this process over a sufficiently long period, the dynamic update will step by step update the reference

sample with RTTs computed from a relayed trajectory, until those times become the RTTs actually expected by the decision function.
We believe that there is 2 main research axes for mitigating this kind of attacks while using an automated update of the reference sample:

1. To include in the updating process a comparison with every previously used reference samples, allowing to raise an alarm when an update looks suspicious.
2. To use a different decision function, that would be able to detect a slow poisoning attempt and to reject with overwhelming probability every poisoned sample.

Bibliography

[1] Patrice Abry and Darryl Veitch, "Wavelet Analysis of Long-Range-Dependent Traffic", *in*: *IEEE Trans. Inf. Theory* 44.*1* (1998), pp. 2–15, DOI: 10.1109/18.650984, URL: https://doi.org/10.1109/18.650984.

[2] Bahaa Al-Musawi, Philip Branch, and Grenville Armitage, "BGP Anomaly Detection Techniques: A Survey", *in*: *IEEE Commun. Surv. Tutorials* 19.*1* (2017), pp. 377–396, DOI: 10.1109/COMST.2016.2622240, URL: https://doi.org/10.1109/COMST.2016.2622240.

[3] Nabil M. Al-Rousan and Ljiljana Trajkovic, "Machine learning models for classification of BGP anomalies", *in*: *13th IEEE International Conference on High Performance Switching and Routing, HPSR 2012, Belgrade, Serbia, June 24-27, 2012*, ed. by Aleksandra Smiljanic, Mounir Hamdi, H. Jonathan Chao, Eiji Oki, and Cyriel Minkenberg, IEEE, 2012, pp. 103–108, DOI: 10.1109/HPSR.2012.6260835, URL: https://doi.org/10.1109/HPSR.2012.6260835.

[4] Tom Anderson, Ken Birman, Robert Broberg, Matthew Caesar, Douglas Comer, Chase Cotton, Michael J Freedman, Andreas Haeberlen, Zachary G Ives, Arvind Krishnamurthy, et al., "The nebula future internet architecture", *in*: *The Future Internet Assembly*, Springer, 2013, pp. 16–26.

[5] Maria Apostolaki, Aviv Zohar, and Laurent Vanbever, "Hijacking Bitcoin: Routing Attacks on Cryptocurrencies", *in*: *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, IEEE Computer Society, 2017, pp. 375–392, DOI: 10.1109/SP.2017.29, URL: https://doi.org/10.1109/SP.2017.29.

[6] Ghada Arfaoui, Gildas Avoine, Olivier Gimenez, and Jacques Traoré, "How Distance-Bounding Can Detect Internet Traffic Hijacking", *in*: *Cryptology and Network Security - 20th International Conference, CANS 2021, Vienna, Austria, December 13-15, 2021, Proceedings*, ed. by Mauro Conti, Marc Stevens, and Stephan Krenn, vol. 13099, Lecture Notes in Computer Science, Springer, 2021, pp. 355–371, DOI: 10.1007/978-3-030-92548-2_19, URL: https://doi.org/10.1007/978-3-030-92548-2%5C_19.

[7] Ghada Arfaoui, Gildas Avoine, Olivier Gimenez, and Jacques Traoré, "ICRP: Internet-Friendly Cryptographic Relay-Detection Protocol", *in*: *Cryptography* 6.*4* (2022), p. 52, DOI: 10.3390/cryptography6040052, URL: https://doi.org/10.3390/cryptography6040052.

[8] *Arstechnica - BGP event sends European mobile traffic through China Telecom for 2 hours*, https://arstechnica.com/information-technology/2019/06/bgp-mishap-sends-european-mobile-traffic-through-china-telecom-for-2-hours/.

[9] *Arstechnica - How China swallowed 15% of net traffic for 18 minutes*, https://arstechnica.com/information-technology/2010/11/how-china-swallowed-15-of-net-traffic-for-18-minutes/.

[10] Gilad Asharov, Daniel Demmler, Michael Schapira, Thomas Schneider, Gil Segev, Scott Shenker, and Michael Zohner, "Privacy-Preserving Interdomain Routing at Internet Scale", *in*: *Proc. Priv. Enhancing Technol.* 2017.*3* (2017), p. 147, DOI: 10.1515/popets-2017-0033, URL: https://doi.org/10.1515/popets-2017-0033.

[11] Gildas Avoine, Muhammed Ali Bingöl, Ioana Boureanu, Srdjan Capkun, Gerhard P. Hancke, Süleyman Kardas, Chong Hee Kim, Cédric Lauradoux, Benjamin Martin, Jorge Munilla, Alberto Peinado, Kasper Bonne Rasmussen, Dave Singelée, Aslan Tchamkerten, Rolando Trujillo-Rasua, and Serge Vaudenay, "Security of Distance-Bounding: A Survey", *in*: *ACM Comput. Surv.* 51.*5* (2019), 94:1–94:33, DOI: 10.1145/3264628, URL: https://doi.org/10.1145/3264628.

[12] Hitesh Ballani, Paul Francis, and Xinyang Zhang, "A study of prefix hijacking and interception in the internet", *in*: *Proceedings of the ACM SIGCOMM 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Kyoto, Japan, August 27-31, 2007*, ed. by Jun Murai and Kenjiro Cho, ACM, 2007, pp. 265–276, DOI: 10.1145/1282380.1282411, URL: https://doi.org/10.1145/1282380.1282411.

[13] Mihir Bellare and Phillip Rogaway, "Random Oracles are Practical: A Paradigm for Designing Efficient Protocols", *in*: *CCS '93, Proceedings of the 1st ACM Conference on Computer and Communications Security, Fairfax, Virginia, USA, November 3-5, 1993*, ed. by Dorothy E. Denning, Raymond Pyle, Ravi Ganesan, Ravi S. Sandhu, and Victoria Ashby, ACM, 1993, pp. 62–73, DOI: 10.1145/168588.168596, URL: https://doi.org/10.1145/168588.168596.

[14] Lenore Blum, Manuel Blum, and Mike Shub, "Comparison of Two Pseudo-Random Number Generators", *in*: *Advances in Cryptology: Proceedings of CRYPTO '82, Santa Barbara, California, USA, August 23-25, 1982*, ed. by David Chaum, Ronald L. Rivest, and Alan T. Sherman, Plenum Press, New York, 1982, pp. 61–78, DOI: 10.1007/978-1-4757-0602-4_6, URL: https://doi.org/10.1007/978-1-4757-0602-4%5C_6.

[15] Ioana Boureanu, Constantin Catalin Dragan, François Dupressoir, David Gérault, and Pascal Lafourcade, "Mechanised Models and Proofs for Distance-Bounding", *in*: *34th IEEE Computer Security Foundations Symposium, CSF 2021, Dubrovnik,*

Croatia, June 21-25, 2021, IEEE, 2021, pp. 1–16, DOI: 10.1109/CSF51468.2021.00049, URL: https://doi.org/10.1109/CSF51468.2021.00049.

[16] Stefan Brands and David Chaum, "Distance-Bounding Protocols (Extended Abstract)", *in*: *Advances in Cryptology - EUROCRYPT '93, Workshop on the Theory and Application of of Cryptographic Techniques, Lofthus, Norway, May 23-27, 1993, Proceedings*, ed. by Tor Helleseth, vol. 765, Lecture Notes in Computer Science, Springer, 1993, pp. 344–359, DOI: 10.1007/3-540-48285-7_30, URL: https://doi.org/10.1007/3-540-48285-7%5C_30.

[17] Bezawada Bruhadeshwar, Sandeep S. Kulkarni, and Alex X. Liu, "Symmetric Key Approaches to Securing BGP - A Little Bit Trust Is Enough", *in*: *IEEE Trans. Parallel Distributed Syst.* 22.*9* (2011), pp. 1536–1549, DOI: 10.1109/TPDS.2011.19, URL: https://doi.org/10.1109/TPDS.2011.19.

[18] Kenneth L. Calvert, Jim Griffioen, and Leonid B. Poutievski, "Separating routing and forwarding: A clean-slate network layer design", *in*: *Fourth International Conference on Broadband Communications, Networks and Systems, (BROADNETS 2007), 10-14 September 2007, Raleigh, North-Carolina, USA*, IEEE, 2007, pp. 261–270, DOI: 10.1109/BROADNETS.2007.4550434, URL: https://doi.org/10.1109/BROADNETS.2007.4550434.

[19] Ang Chen and Andreas Haeberlen, "PRISM: Private Retrieval of the Internet's Sensitive Metadata", *in*: *8th Workshop on Cyber Security Experimentation and Test (CSET 15)*, Washington, D.C.: USENIX Association, Aug. 2015, URL: https://www.usenix.org/conference/cset15/workshop-program/presentation/chen.

[20] Laurent Chuat, Markus Legner, David A. Basin, David Hausheer, Samuel Hitz, Peter Müller, and Adrian Perrig, *The Complete Guide to SCION - From Design Principles to Formal Verification*, Information Security and Cryptography, Springer, 2022, ISBN: 978-3-031-05287-3, DOI: 10.1007/978-3-031-05288-0, URL: https://doi.org/10.1007/978-3-031-05288-0.

[21] Shivani Deshpande, Marina Thottan, Tin Kam Ho, and Biplab Sikdar, "An Online Mechanism for BGP Instability Detection and Analysis", *in*: *IEEE Trans. Computers* 58.*11* (2009), pp. 1470–1484, DOI: 10.1109/TC.2009.91, URL: https://doi.org/10.1109/TC.2009.91.

[22] Yvo Desmedt, Claude Goutier, and Samy Bengio, "Special Uses and Abuses of the Fiat-Shamir Passport Protocol", *in*: *Advances in Cryptology - CRYPTO '87, A Conference on the Theory and Applications of Cryptographic Techniques, Santa Barbara, California, USA, August 16-20, 1987, Proceedings*, ed. by Carl Pomerance, vol. 293, Lecture Notes in Computer Science, Springer, 1987, pp. 21–39, DOI: 10.1007/3-540-48184-2_3, URL: https://doi.org/10.1007/3-540-48184-2%5C_3.

[23] Nick Feamster, Hari Balakrishnan, Jennifer Rexford, Aman Shaikh, and Jacobus E. van der Merwe, "The case for separating routing from routers", *in*: *Proceedings of the ACM SIGCOMM workshop on Future directions in network architecture, FDNA '04, Portland, Oregon, USA, August 30, 2004*, ed. by Kevin Fall and Srinivasan Keshav, ACM, 2004, pp. 5–12, DOI: 10.1145/1016707.1016709, URL: https://doi.org/10.1145/1016707.1016709.

[24] Amos Fiat and Adi Shamir, "How to Prove Yourself: Practical Solutions to Identification and Signature Problems", *in*: *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, ed. by Andrew M. Odlyzko, vol. 263, Lecture Notes in Computer Science, Springer, 1986, pp. 186–194, DOI: 10.1007/3-540-47721-7_12, URL: https://doi.org/10.1007/3-540-47721-7%5C_12.

[25] Murat Can Ganiz, Sudhan Kanitkar, Mooi Choo Chuah, and William M. Pottenger, "Detection of Interdomain Routing Anomalies Based on Higher-Order Path Analysis", *in*: *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006), 18-22 December 2006, Hong Kong, China*, IEEE Computer Society, 2006, pp. 874–879, DOI: 10.1109/ICDM.2006.52, URL: https://doi.org/10.1109/ICDM.2006.52.

[26] Joseph Gersch and Daniel Massey, "ROVER: Route Origin Verification Using DNS", *in*: *22nd International Conference on Computer Communication and Networks, ICCCN 2013, Nassau, Bahamas, July 30 - Aug. 2, 2013*, IEEE, 2013, pp. 1–9, DOI: 10.1109/ICCCN.2013.6614187, URL: https://doi.org/10.1109/ICCCN.2013.6614187.

[27] Brighten Godfrey, Igor Ganichev, Scott Shenker, and Ion Stoica, "Pathlet routing", *in*: *Proceedings of the ACM SIGCOMM 2009 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Barcelona, Spain, August 16-21, 2009*, ed. by Pablo Rodriguez, Ernst W. Biersack, Konstantina Papagiannaki, and Luigi Rizzo, ACM, 2009, pp. 111–122, DOI: 10.1145/1592568.1592583, URL: https://doi.org/10.1145/1592568.1592583.

[28] Shafi Goldwasser, Silvio Micali, and Charles Rackoff, "The knowledge complexity of interactive proof-systems", *in*: *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, ed. by Oded Goldreich, ACM, 2019, pp. 203–225, DOI: 10.1145/3335741.3335750, URL: https://doi.org/10.1145/3335741.3335750.

[29] Geoffrey Goodell, William Aiello, Timothy Griffin, John Ioannidis, Patrick D. McDaniel, and Aviel D. Rubin, "Working around BGP: An Incremental Approach to Improving Security and Accuracy in Interdomain Routing", *in*: *Proceedings of the Network and Distributed System Security Symposium, NDSS 2003, San Diego, California, USA*, The Internet Society, 2003, URL: https://www.ndss-symposium.org/ndss2003/working-around-bgp-incremental-approach-improving-security-and-accuracy-interdomain-routing/.

[30] Frank E. Grubbs, "Sample Criteria for Testing Outlying Observations", *in*: *The Annals of Mathematical Statistics* 21.*1* (1950), pp. 27–58, DOI: 10.1214/aoms/1177729885, URL: https://doi.org/10.1214/aoms/1177729885.

[31] Yi Guo, Hai-Xin Duan, Jikun Chen, and Fu Miao, "MAF-SAM: An effective method to perceive data plane threats of inter domain routing system", *in*: *Comput. Networks* 110 (2016), pp. 69–78, DOI: 10.1016/j.comnet.2016.09.017, URL: https://doi.org/10.1016/j.comnet.2016.09.017.

[32] Debayan Gupta, Aaron Segal, Aurojit Panda, Gil Segev, Michael Schapira, Joan Feigenbaum, Jennifer Rexford, and Scott Shenker, "A new approach to interdomain routing based on secure multi-party computation", *in*: *11th ACM Workshop on Hot Topics in Networks, HotNets-XI, Redmond, WA, USA - October 29 - 30, 2012*, ed. by Srikanth Kandula, Jitendra Padhye, Emin Gün Sirer, and Ramesh Govindan, ACM, 2012, pp. 37–42, DOI: 10.1145/2390231.2390238, URL: https://doi.org/10.1145/2390231.2390238.

[33] *Hacker Target - Autonomous System Lookup*, https://hackertarget.com/as-ip-lookup/.

[34] Andreas Haeberlen, Ioannis C. Avramopoulos, Jennifer Rexford, and Peter Druschel, "NetReview: Detecting When Interdomain Routing Goes Wrong", *in*: *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2009, April 22-24, 2009, Boston, MA, USA*, ed. by Jennifer Rexford and Emin Gün Sirer, USENIX Association, 2009, pp. 437–452, URL: http://www.usenix.org/events/nsdi09/tech/full%5C_papers/haeberlen/haeberlen.pdf.

[35] Gerhard P. Hancke and Markus G. Kuhn, "An RFID Distance Bounding Protocol", *in*: *First International Conference on Security and Privacy for Emerging Areas in Communications Networks, SecureComm 2005, Athens, Greece, 5-9 September, 2005*, IEEE, 2005, pp. 67–73, DOI: 10.1109/SECURECOMM.2005.56, URL: https://doi.org/10.1109/SECURECOMM.2005.56.

[36] Rahul Hiran, Niklas Carlsson, and Nahid Shahmehri, "Crowd-based detection of routing anomalies on the internet", *in*: *2015 IEEE Conference on Communications and Network Security, CNS 2015, Florence, Italy, September 28-30, 2015*, IEEE, 2015, pp. 388–396, DOI: 10.1109/CNS.2015.7346850, URL: https://doi.org/10.1109/CNS.2015.7346850.

[37] Thomas Holterbach, Stefano Vissicchio, Alberto Dainotti, and Laurent Vanbever, "SWIFT: Predictive Fast Reroute", *in*: *Proceedings of the Conference of the ACM Special Interest Group on Data Communication, SIGCOMM 2017, Los Angeles, CA, USA, August 21-25, 2017*, ACM, 2017, pp. 460–473, DOI: 10.1145/3098822.3098856, URL: https://doi.org/10.1145/3098822.3098856.

[38] Xin Hu and Zhuoqing Morley Mao, "Accurate Real-time Identification of IP Prefix Hijacking", *in*: *2007 IEEE Symposium on Security and Privacy (S&P 2007), 20-23 May 2007, Oakland, California, USA*, IEEE Computer Society, 2007, pp. 3–17, DOI: 10.1109/SP.2007.7, URL: https://doi.org/10.1109/SP.2007.7.

[39] Yih-Chun Hu, Adrian Perrig, and David B. Johnson, "Efficient Security Mechanisms for Routing Protocolsa", *in*: *Proceedings of the Network and Distributed System Security Symposium, NDSS 2003, San Diego, California, USA*, The Internet Society, 2003, URL: https://www.ndss-symposium.org/ndss2003/efficient-security-mechanisms-for-routing-protocols/.

[40] Yih-Chun Hu, Adrian Perrig, and Marvin A. Sirbu, "SPV: secure path vector routing for securing BGP", *in*: *Proceedings of the ACM SIGCOMM 2004 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, August 30 - September 3, 2004, Portland, Oregon, USA*, ed. by Raj Yavatkar, Ellen W. Zegura, and Jennifer Rexford, ACM, 2004, pp. 179–192, DOI: 10.1145/1015467.1015488, URL: https://doi.org/10.1145/1015467.1015488.

[41] Yiyi Huang, Nick Feamster, Anukool Lakhina, and Jun (Jim) Xu, "Diagnosing network disruptions with network-wide analysis", *in*: *Proceedings of the 2007 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS 2007, San Diego, California, USA, June 12-16, 2007*, ed. by Leana Golubchik, Mostafa H. Ammar, and Mor Harchol-Balter, ACM, 2007, pp. 61–72, DOI: 10.1145/1254882.1254890, URL: https://doi.org/10.1145/1254882.1254890.

[42] Josh Karlin, Stephanie Forrest, and Jennifer Rexford, "Pretty Good BGP: Improving BGP by Cautiously Adopting Routes", *in*: *Proceedings of the 14th IEEE International Conference on Network Protocols, ICNP 2006, November 12-15, 2006, Santa Barbara, California, USA*, IEEE Computer Society, 2006, pp. 290–299, DOI: 10.1109/ICNP.2006.320179, URL: https://doi.org/10.1109/ICNP.2006.320179.

[43] Stephen T. Kent, Charles Lynn, and Karen Seo, "Secure Border Gateway Protocol (S-BGP)", *in*: *IEEE J. Sel. Areas Commun.* 18.*4* (2000), pp. 582–592, DOI: 10.1109/49.839934, URL: https://doi.org/10.1109/49.839934.

[44] Craig Labovitz, G. Robert Malan, and Farnam Jahanian, "Internet routing instability", *in*: *IEEE/ACM Trans. Netw.* 6.*5* (1998), pp. 515–528, DOI: 10.1109/90.731185, URL: https://doi.org/10.1109/90.731185.

[45] Mohit Lad, Daniel Massey, Dan Pei, Yiguo Wu, Beichuan Zhang, and Lixia Zhang, "PHAS: A Prefix Hijack Alert System", *in*: *Proceedings of the 15th USENIX Security Symposium, Vancouver, BC, Canada, July 31 - August 4, 2006*, ed. by Angelos D. Keromytis, USENIX Association, 2006, URL: https://www.usenix.org/conference/15th-usenix-security-symposium/phas-prefix-hijack-alert-system.

[46] Matt Lepinski and Kotikalapudi Sriram, "BGPsec Protocol Specification", *in*: *RFC* 8205 (2017), pp. 1–45, DOI: 10.17487/RFC8205, URL: https://doi.org/10.17487/RFC8205.

[47] Jun Li, Dejing Dou, Zhen Wu, Shiwoong Kim, and Vikash Agarwal, "An internet routing forensics framework for discovering rules of abnormal BGP events", *in*: *Comput. Commun. Rev.* 35.5 (2005), pp. 55–66, DOI: 10.1145/1096536.1096542, URL: https://doi.org/10.1145/1096536.1096542.

[48] Jianning Mai, Lihua Yuan, and Chen-Nee Chuah, "Detecting BGP anomalies with wavelet", *in*: *IEEE/IFIP Network Operations and Management Symposium: Pervasive Management for Ubioquitous Networks and Services, NOMS 2008, 7-11 April 2008, Salvador, Bahia, Brazil*, ed. by Marcus Brunner, Carlos Becker Westphall, and Lisandro Zambenedetti Granville, IEEE, 2008, pp. 465–472, DOI: 10.1109/NOMS.2008.4575169, URL: https://doi.org/10.1109/NOMS.2008.4575169.

[49] *MF3D(H)x3, MIFARE DESFire EV*, https://www.nxp.com/docs/en/data-sheet/MF3DHx3_SDS.pdf.

[50] *MIFARE Plus EV2*, https://www.nxp.com/docs/en/data-sheet/MF1P(H)x2_SDS.pdf.

[51] Asya Mitseva, Andriy Panchenko, and Thomas Engel, "The state of affairs in BGP security: A survey of attacks and defenses", *in*: *Comput. Commun.* 124 (2018), pp. 45–60, DOI: 10.1016/j.comcom.2018.04.013, URL: https://doi.org/10.1016/j.comcom.2018.04.013.

[52] *NIST - Cryptographic Hash Functions*, https://csrc.nist.gov/glossary/term/cryptographic_hash_function.

[53] *NRO-Statistics-2021-Q4*, https://www.nro.net/wp-content/uploads/NRO-Statistics-2022-Q2-FINAL.pdf.

[54] Adrian Perrig, Pawel Szalachowski, Raphael M. Reischuk, and Laurent Chuat, *SCION: A Secure Internet Architecture*, Information Security and Cryptography, Springer, 2017, ISBN: 978-3-319-67079-9, DOI: 10.1007/978-3-319-67080-5, URL: https://doi.org/10.1007/978-3-319-67080-5.

[55] Jon Postel, "Internet Protocol", *in*: *RFC* 791 (1981), pp. 1–51, DOI: 10.17487/RFC0791, URL: https://doi.org/10.17487/RFC0791.

[56] Jon Postel, "User Datagram Protocol", *in*: *RFC* 768 (1980), pp. 1–3, DOI: 10.17487/RFC0768, URL: https://doi.org/10.17487/RFC0768.

[57] B. Aditya Prakash, Nicholas Valler, David G. Andersen, Michalis Faloutsos, and Christos Faloutsos, "BGP-lens: patterns and anomalies in internet routing updates", *in*: *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, France, June 28 - July 1, 2009*, ed. by John F. Elder IV, Françoise Fogelman-Soulié, Peter A. Flach, and Mohammed Javeed Zaki, ACM, 2009, pp. 1315–1324, DOI: 10.1145/1557019.1557160, URL: https://doi.org/10.1145/1557019.1557160.

[58] Jian Qiu, Lixin Gao, Supranamaya Ranjan, and Antonio Nucci, "Detecting bogus BGP route information: Going beyond prefix hijacking", *in*: *Third International Conference on Security and Privacy in Communication Networks and the Workshops, SecureComm 2007, Nice, France, 17-21 September, 2007*, IEEE, 2007, pp. 381–390, DOI: 10.1109/SECCOM.2007.4550358, URL: https://doi.org/10.1109/SECCOM.2007.4550358.

[59] Sophie Y Qiu, Fabian Monrose, Andreas Terzis, and Patrick D McDaniel, "Efficient techniques for detecting false origin advertisements in inter-domain routing", *in*: *2006 2nd IEEE Workshop on Secure Network Protocols*, IEEE, 2006, pp. 12–19, DOI: 10.1109/NPSEC.2006.320341.

[60] Tongqing Qiu, Lusheng Ji, Dan Pei, Jia Wang, and Jun (Jim) Xu, "TowerDefense: Deployment strategies for battling against IP prefix hijacking", *in*: *Proceedings of the 18th annual IEEE International Conference on Network Protocols, ICNP 2010, Kyoto, Japan, 5-8 October, 2010*, IEEE Computer Society, 2010, pp. 134–143, DOI: 10.1109/ICNP.2010.5762762, URL: https://doi.org/10.1109/ICNP.2010.5762762.

[61] Barath Raghavan, Saurabh Panjwani, and Anton Mityagin, "Analysis of the SPV secure routing protocol: weaknesses and lessons", *in*: *Comput. Commun. Rev.* 37.*2* (2007), pp. 29–38, DOI: 10.1145/1232919.1232923, URL: https://doi.org/10.1145/1232919.1232923.

[62] Barath Raghavan, Patrick Verkaik, and Alex C. Snoeren, "Secure and policy-compliant source routing", *in*: *IEEE/ACM Trans. Netw.* 17.*3* (2009), pp. 764–777, DOI: 10.1145/1569732.1569739, URL: http://doi.acm.org/10.1145/1569732.1569739.

[63] Yakov Rekhter and Tony Li, "A Border Gateway Protocol 4 (BGP-4)", *in*: *RFC* 1654 (1994), pp. 1–56, DOI: 10.17487/RFC1654, URL: https://doi.org/10.17487/RFC1654.

[64] *RIPE - Youtube Hijacking, a RIPE NCC RIS case study*, https://www.ripe.net/publications/news/industry-developments/youtube-hijacking-a-ripe-ncc-ris-case-study.

[65] *RIPE Network Coordination Center*, https://www.ripe.net/.

[66] Pavlos Sermpezis, Vasileios Kotronis, Petros Gigis, Xenofontas A. Dimitropoulos, Danilo Cicalese, Alistair King, and Alberto Dainotti, "ARTEMIS: Neutralizing BGP Hijacking Within a Minute", *in*: *IEEE/ACM Trans. Netw.* 26.*6* (2018), pp. 2471–2486, DOI: 10.1109/TNET.2018.2869798, URL: https://doi.org/10.1109/TNET.2018.2869798.

[67] Xingang Shi, Yang Xiang, Zhiliang Wang, Xia Yin, and Jianping Wu, "Detecting prefix hijackings in the internet with argus", *in*: *Proceedings of the 12th ACM SIGCOMM Internet Measurement Conference, IMC '12, Boston, MA, USA, November 14-16, 2012*, ed. by John W. Byers, Jim Kurose, Ratul Mahajan, and Alex C. Snoeren, ACM, 2012, pp. 15–28, DOI: 10.1145/2398776.2398779, URL: https://doi.org/10.1145/2398776.2398779.

[68] Victor Shoup, "Sequences of games: a tool for taming complexity in security proofs", *in*: *IACR Cryptol. ePrint Arch.* (2004), p. 332, URL: http://eprint.iacr.org/2004/332.

[69] Georgos Siganos and Michalis Faloutsos, "Neighborhood Watch for Internet Routing: Can We Improve the Robustness of Internet Routing Today?", *in*: *INFOCOM 2007. 26th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 6-12 May 2007, Anchorage, Alaska, USA*, IEEE, 2007, pp. 1271–1279, DOI: 10.1109/INFCOM.2007.151, URL: https://doi.org/10.1109/INFCOM.2007.151.

[70] Mitsuho Tahara, Naoki Tateishi, Toshio Oimatsu, and Souhei Majima, "A Method to Detect Prefix Hijacking by Using Ping Tests", *in*: *Challenges for Next Generation Network Operations and Service Management, 11th Asia-Pacific Network Operations and Management Symposium, APNOMS 2008, Beijing, China, October 22-24, 2008. Proceedings*, ed. by Yan Ma, Deokjai Choi, and Shingo Ata, vol. 5297, Lecture Notes in Computer Science, Springer, 2008, pp. 390–398, DOI: 10.1007/978-3-540-88623-5_40, URL: https://doi.org/10.1007/978-3-540-88623-5%5C_40.

[71] *University of Oregon Route Views Archive Project*, https://routeviews.org/.

[72] Iñigo Ortiz de Urbina Cazenave, Erkan Köşlük, and Murat Can Ganiz, "An anomaly detection framework for BGP", *in*: *2011 International Symposium on Innovations in Intelligent Systems and Applications*, IEEE, 2011, pp. 107–111, DOI: 10.1109/INISTA.2011.5946083.

[73] Tao Wan, Evangelos Kranakis, and Paul C. van Oorschot, "Pretty Secure BGP, psBGP", *in*: *Proceedings of the Network and Distributed System Security Symposium, NDSS 2005, San Diego, California, USA*, The Internet Society, 2005, URL: https://www.ndss-symposium.org/ndss2005/pretty-secure-bgp-psbgp/.

[74] Russ White, "Securing BGP through secure origin BGP (soBGP)", *in*: *Business Communications Review* 33.*5* (2003), pp. 47–47.

[75] Ian H. Witten, Eibe Frank, and Mark A. Hall, *Data mining: practical machine learning tools and techniques, 3rd Edition*, Morgan Kaufmann, Elsevier, 2011, ISBN: 9780123748560, URL: https://www.worldcat.org/oclc/262433473.

[76] Xiaowei Yang, David Clark, and Arthur W. Berger, "NIRA: a new inter-domain routing architecture", *in*: *IEEE/ACM Trans. Netw.* 15.*4* (2007), pp. 775–788, DOI: 10.1145/1295257.1295261, URL: http://doi.acm.org/10.1145/1295257.1295261.

[77] Heng Yin, Bo Sheng, Haining Wang, and Jianping Pan, "Keychain-Based Signatures for Securing BGP", *in*: *IEEE J. Sel. Areas Commun.* 28.*8* (2010), pp. 1308–1318, DOI: 10.1109/JSAC.2010.101008, URL: https://doi.org/10.1109/JSAC.2010.101008.

[78] Changxi Zheng, Lusheng Ji, Dan Pei, Jia Wang, and Paul Francis, "A light-weight distributed scheme for detecting ip prefix hijacks in real-time", *in*: *Proceedings of the ACM SIGCOMM 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Kyoto, Japan, August 27-31, 2007*, ed. by Jun Murai and Kenjiro Cho, ACM, 2007, pp. 277–288, DOI: 10.1145/1282380.1282412, URL: https://doi.org/10.1145/1282380.1282412.

COLLEGE DOCTORAL BRETAGNE
MATHS, TELECOMS
INFORMATIQUE, SIGNAL
SYSTEMES, ELECTRONIQUE

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES RENNES

Titre : Attaques par Relais sur Internet : Détection d'Anomalies par Mesures de Temps.

Mot clés : Attaques par relais, distance-bounding, protocole de détection de relai

Résumé : Le réseau Internet est rapidement devenu un outil permettant le partage des connaissances, la distribution de services et de divertissements, et la connexion de plusieurs milliards d'utilisateurs dans le monde entier. Le défi de correctement acheminer des informations sur un tel réseau implique une topologie complexe sans précédent. Sans pouvoir être exhaustif, cette topologie dépend de la position géographique des nœuds, d'alliances commerciales ou gouvernementales, ou encore du coût de construction et d'installation des câbles...
De nos jours, ce défi est relevé par une approche collaborative dans laquelle une communication de pair à pair permet la construction de table de routage reliant chaque équipement connecté. Dans ce contexte, une famille d'attaque a rapidement émergé : les attaques par détournement de trafic. Une attaque par détournement de trafic est définie par l'altération malveillante d'une ou plusieurs routes. Ce genre d'attaques peut être dérivé en 3 catégories principales : les attaques *Trou noir*, dans lesquelles les paquets sont détruits ou jetés avant d'atteindre leur destination, les attaques de *Redirection*, dans lesquelles les paquets sont routés vers la mauvaise destination, et les attaques par *Relai*, dans lesquelles les paquets traversent un ensemble de nœuds illégitimes avant d'arriver à leur destination. Qu'elles soient intentionnelles ou accidentelles, les occurrences de détournements de trafic sont de plus en plus fréquentes depuis quelques décennies. Ceci met clairement en évidence un besoin de repenser la façon dont sont supervisées nos communications.
Sur ce sujet, le monde de la recherche concentre ses efforts sur des solutions visant à empêcher ces attaques, soit en ajoutant des couches sécuritaires aux protocoles existants, soit en proposant une architecture de routage complètement nouvelle. Dans le premier cas, l'enjeu est de fournir une protection sans faille, tout en n'introduisant aucune latences supplémentaires. Le deuxième cas, quant à lui, implique un très long processus de standardisation, et surtout le besoin de convaincre le monde de passer d'une architecture à une autre.
Durant ces 3 ans, nous avons exploré une troisième option, visant à détecter rapidement et efficacement une attaque par relai. L'objectif est de construire un protocole au design simple, sans prise en compte la complexité de la topologie d'Internet, et pouvant être déployé quel que soit le protocole de routage sous-jacent. Notre proposition s'inspire d'un mécanisme dit de "distance bounding", une famille de protocoles permettant une authentification interactive, mesurant le temps aller-retours des messages pour décider si une attaque est en cours. Notre construction est soutenue par des mesures de temps à l'échelle mondiale, permettant aussi bien de montrer son applicabilité pratique que d'en évaluer les performances. Le protocole proposé est *efficace* - il n'utilise que 2 opérations cryptographiques par exécution, impliquant une charge négligeable pour les utilisateurs, et de faibles pertes en termes de débit, *applicable* - aucune mise à jour n'est requise pour les nœuds du réseau, *indépendant du protocole de routage* - la méthode de routage n'a pas d'impact sur notre schéma, *sans impact sur les performances réseau* - le volume de données supplémentaires en transit n'est que de l'ordre d'1.5%, et *sécurisé* - nous fournissons une preuve de sécurité complète dans le modèle de l'oracle aléatoire.

Title: Relay Attacks over the Internet: Anomaly Detection using Time Measurement

Keywords: Relay Attacks, distance-bounding, relay-detection protocol

Abstract: The Internet has grown to become a massive communication tool, spreading out knowledge, offering entertainment and services, and connecting billions of people worldwide. The challenge of properly routing information over such a network infers an unprecedentedly complex topology depending non-exhaustively on geographical position of the nodes, commercial or governmental alliances, or construction cost of physical links...
Nowadays, this challenge is overcome by a collaborative approach in which a permanent pear to pear communication allows to construct global routing tables linking every connected equipments together. In this context, a family of attacks rapidly emerged: the hijacking attacks. An hijacking attack is defined by any malicious alteration of the standard construction of one or several routes. Such attacks can be derivated in 3 main categories: *Black hole*, the packets are thrown before reaching their destination, *Redirection*, the packets are routed to the wrong destination, *Relay*, the packets travel through an undesired set of nodes before reaching their destination. Whether intentional or accidental, hijacking events have become more and more frequent over the last decades, highlighting a clear need to rethink the way we supervise our communications.
The responses on that matter are focusing on mitigation, whether by adding security layers to the current protocols, or by designing novel routing architecture from scratch. For the former, it requires providing a flawless protection without introducing latencies. For the latter, it requires a very long process of standardization, assuming the entire world agrees on one given architecture.
During these 3 years, we have decided to explore a third option, aiming to efficiently and quickly detect when a relay hijacking attack is ongoing. The goal of this research is to construct a protocol, simple in its design, that does not need to consider the topology of the Internet, and that could be deployed regardless of the underlying routing process. Our proposal relies on a distance-bounding mechanism that performs interactive authentication with a “Challenge-Response” exchange, and measures the round-trip time of messages to decide whether an attack is ongoing. Over the course of this manuscript, we explore the adaptability of the idea of distance bounding in the far more dynamic environment that is the Internet. Our construction is supported by worldwide experiments on communication time between multiple nodes, allowing us to both demonstrate its applicability and evaluate its performances. The final protocol is *efficient* - it requires only two cryptographic operations per execution, inducing negligible workload for users and very few losses of throughput, *scalable* - no software updates are required for intermediate network nodes, *routing protocol independent* - this means that any future update of the route selection process will not induce changes on our scheme, *network friendly* - the added volume of transiting data is only about 1.5%, and *secure* - we provide a complete security proof in the random oracle model.