Cryptographic mechanisms for device authentication and attestation in the internet of things

Aïda Abdou Diop

INSTITUT POLYTECHNIQUE DE PARIS

Thèse de doctorat

Protocoles Cryptographiques pour l'Authentification et l'Attestation Anonymes dans l'Internet des Objets

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à Télécom SudParis

École doctorale n°626 Institut Polytechnique de Paris (IP Paris)
Spécialité de doctorat : Informatique

Thèse présentée et soutenue à Palaiseau, le 30/11/2020, par

AÏDA ABDOU DIOP

Composition du Jury :

Samia Bouzefrane Professeur, CNAM	Présidente
Daniel Augot Directeur de recherche, INRIA Saclay	Rapporteur
Liqun Chen Professeur, University of Surrey	Examinatrice
Youssef Laarouchi Ingénieur de recherche, EDF	Examinateur
Maryline Laurent Professeur, Télécom SudParis	Directrice de thèse
Jean Leneutre Maître de conférences, Télécom Paris	Co-encadrant de thèse
Saïd Gharout Ingénieur de recherche, ARM	Invité
Jacques Traoré Ingénieur de recherche, Orange	Invité

Cryptographic Mechanisms for Device Authentication and Attestation in the Internet of Things

By

Aïda Abdou Diop

Supervisors: Maryline Laurent, Jean Leneutre & Jacques Traoré

Abstract

The new decentralized computing paradigm introduced by Machine-to-Machine (M2M) communications and the Internet of Things (IoT) ecosystem requires developing new security mechanisms and frameworks, adapted to this new decentralized architecture. Internet of Things applications are pervasive and present in the vast majority of industries, with the aim of increasing efficiency and safety in industrial processes and consumer-driven applications. The variety of IoT use cases includes applications leveraging low-level devices such as sensor or actuators, to applications deploying safety critical devices such as connected vehicles in Intelligent Transportation Systems (ITS). Devices are deployed as nodes in communication networks, and have become in recent years targets for attackers who exploit the resource-constrained nature of the devices in order to compromise the safety, security, and availability of the different applications. Two of the main challenges in this ecosystem are securing the communication between IoT devices, and ensuring that devices in the network have not been compromised or tampered with, thus attesting of the integrity of the entire network. The challenges are exacerbated by the nature of devices, which present stringent constraints, notably in terms of computational capabilities, storage space, and energy resource. In addition, new privacy concerns affecting users in IoT applications have risen, and require implementing privacy-friendly authentication and attestation mechanisms. Authentication mechanisms allow systems to identify themselves on the network, and provide solutions for the first challenge. Remote Attestation is a security mechanism which enables control systems to verify the software state of devices in the network, thus detecting any tampering or remote malware injection attacks.

In this thesis, we aim to contribute to the development of new and privacy-preserving authentication and attestation mechanisms, which are particularly adapted for implementation in constrained environments.

In the traditional computing architecture, secure authentication is addressed through the deployment of Public Key Infrastructures (PKI), which create and manage credentials used in the secure communication between different entities. PKIs notably include a trusted management entity called the Certification Authority (CA), in charge of generating credentials for each device and its public key. This centralized architecture introduces privacy concerns, as well as scalability challenges in the context of IoT applications.

In the first part of this thesis, we leverage a cryptographic mechanism deployed in trusted computing, namely Direct Anonymous Attestation (DAA), in order to provide decentralized, and privacy-preserving authentication protocols adapted for constrained environments. Our work contributes to the development of a variant of Direct Anonymous Attestation schemes, called pre-Direct Anonymous Attestation (pre-DAA), which achieves a trade-off between security and efficiency that was not previously achieved in the literature. In particular, our pre-DAA scheme is proven secure in the Random Oracle Model (ROM) under the q-Strong Diffie Hellman (q—SDH) assumption, while performing better than DAA schemes proven secure under an interactive assumption. The pre-DAA scheme is subsequently used in the development of two privacy-preserving

authentication protocols. The first application of our pre-DAA scheme consists in the design of a decentralized architecture for secure communication in vehicular ad hoc networks, which removes the need for a centralize Public Key Infrastructure. The second application of our pre-DAA scheme is the design of a mobile-based access control protocol for public transport systems, which addresses the issue of user traceability inherent to current access control control protocols for transport systems. Using our particular protocol, a user is able to anonymously authenticate on the public transport network in 173 ms with some precomputation.

In the second part of this thesis, we address the device integrity verification challenge by designing a remote attestation protocol which enables the secure and efficient attestation of groups (or swarms) of devices. Our attestation protocol verifies the integrity of every device in the network during a single attestation phase, by leveraging the aggregating properties of an aggregate algebraic MAC scheme. Compared to swarm attestation protocols in the literature, our contribution enables the detection of an erroneous attestation report in the aggregated result, thus allowing the identification of compromised devices.

Résumé

En introduisant de nouvelles exigences de communications décentralisées entre les systèmes d'informations, l'Internet des Objets (IdO) et le Machine-to-Machine (M2M) ont révolutionné l'architecture de sécurité classique des dits systèmes. Dans cette nouvelle configuration, la sécurité et la protection des données à caractère personnel échangées puis stockées par ces systèmes est devenu un enjeu primordial pour le déploiement de ces nouveaux écosystèmes. Ces objets présentent aussi des contraintes physiques fortes impactant leurs fonctionnalités, notamment en termes de capacité de calcul et de mémoire, d'énergie, et d'exigences de sécurité en fonction du cas d'usage et de l'application concernés. Ces nouveaux modèles de communications requièrent une nouvelle manière de penser la sécurité des systèmes, qui est notamment dû au déploiement physiques des objets dans le monde réel, et à grande échelle. Cet accès physique permet notamment à un attaquant d'observer un objet, et d'en perturber le fonctionnement. Nous nous intéresserons dans cette thèse aux problématiques liées à l'intégrité des communications entre les objets et les systèmes embarqués pour différents cas d'usages dans l'Internet des Objets, ainsi qu'à l'intégrité des micro-logiciels des dits systèmes. En d'autres termes, nous nous intéressons aux mécanismes d'authentification et d'attestation adaptés pour des environnements contraints.

Les solutions cryptographiques permettant d'assurer la confidentialité des communications pour les environnements contraints sont matures et pérennes. Elles ont notamment fait l'objet d'avancées concrètes par le développement de primitives cryptographiques à bas coût. En revanche, il existe des problématiques de sécurité concrètes concernant l'intégrité des communications, ainsi que l'intégrité des systèmes embarqués utilisés dans des environnements contraints.

D'une part, la question de l'intégrité des communications relève des mécanismes d'authentification des communications. Ces dernières sont basées sur des algorithmes de cryptographie à clé publique, et permettent à chaque nœud d'un réseau de communication de s'assurer de l'identité de chaque nœud sur le réseau, et ainsi d'authentifier les messages échangés.

D'un autre côté, la question de l'intégrité des objets eux mêmes, et notamment de leur micrologicielle, fait appelle à des mécanismes d'authentification à distance appelés "Attestation à distance" (ou Remote Attestation). Le mécanisme d'attestation à distance a été introduit dans l'informatique de confiance il y'a déjà deux décennies, notamment pour faire face au développement de nouveaux systèmes décentralisés qui viennent en rupture des structures centralisées des fonctions de sécurité. L'avènement de l'Internet des Objets a permis un regain d'intérêt pour ce mécanisme, notamment dans le but de vérifier l'intégrité logicielle des objets, et ainsi détecter des attaques et tentatives d'intrusion.

L'authentification et l'attestation des objets dans l'Internet des Objets requiert une identification forte et mutuelle entre les objets, ainsi qu'avec les stations de bases. En effet, pour pouvoir authentifier les messages d'un objet précis, qui dans certains cas peut être une commande de contrôle importante, il est nécessaire de s'assurer de l'identité de l'objet. Cette contrainte offre un avantage certains aux attaquants, qui peuvent exploiter cette information pour lancer des attaques ciblées compromettant ainsi la sûreté de ces systèmes, ou pour extraire des données à caractère

personnel sur les utilisateurs finaux dans certains cas d'usages.

Cette thèse se focalise sur le développement de mécanismes d'authentification et d'attestation anonymes et efficaces. Nous introduisons dans une première partie deux protocoles d'authentification se basant sur une nouvelle primitive d'attestation anonyme (pre-Direct Anonymous Attestation (pre-DAA)), pour les cas d'usages des véhicules connectés et d'authentification anonyme pour les pass de transports mobiles. Notre nouvelle primitive de pre-DAA est prouvée sûr dans le modèle de l'oracle aléatoire, sous une variante de l'hypothèse q—SDH . La deuxième partie de cette thèse se concentre ensuite sur le développement d'un mécanisme d'attestation d'essaims d'objets, avec une propriété de détection lorsque l'un des objets fournit une fausse attestation.

Remerciements

Je suis reconnaissante envers un grand nombre de personnes qui m'ont permis de mener à bien cette thèse. Tout d'abord, je tiens à remercier Maryline et Jean pour leur accompagnement tout au long de cette thèse. Merci de m'avoir donné une grande autonomie dans mon travail de recherche, tout en veillant à ce que je garde les objectifs finaux en tête. Je tiens à remercier Saïd, qui bien qu'ayant dû nous quitter en cours de route, aura été d'un grand soutien, notamment durant les périodes qui ont précédé le début de la thèse. Merci Saïd de m'avoir initié au monde de la normalisation, des acquis que j'emporte avec moi dans mes aventures futures. Les résultats présentés dans ce manuscrit ont été rendu possible grâce à l'encadrement plein d'enthousiasme de Jacques. Je te remercie Jacques pour ta disponibilité, ta patience, et tes encouragements durant ces trois dernières années. Je te suis reconnaissante de m'avoir initié aux multiples et diverses applications pratiques de la sécurité prouvée, et salue ta capacité à initier et accompagner une doctorante encore novice dans ce domaine.
Je remercie sincèrement Daniel Augot et Samia Bouzefrane, d'avoir tout d'abord accepté de faire partie de mon jury à mi-parcours, et aussi d'avoir accepté de rapporter ce mémoire, en me fournissant des remarques pertinentes qui m'auront permis de l'amélioer. I sincerely thank Liqun Chen and Youssef Laarouchi for accepting to be part of my thesis committee.
Durant ces trois dernières années, j'ai eu le grand plaisir d'effectuer mon travail de recherche au sein de l'équipe Security–Privacy–Innovation dirigée par Jean-François Misarsky. Je tiens à remercier Jean-François et toute l'équipe SPI pour leur gentillesse, leur bonne humeur, les pauses-café toujours pleines de sourires (et de gâteaux). Cette équipe, je l'ai tout d'abord intégré lors de mon stage sous l'encadrement de Sébastien et Marie, que je remercie de m'avoir non seulement fait découvrir le monde de la recherche, mais aussi pour l'aventure Cybercrypt à laquelle j'ai pris beaucoup de plaisir à participer. Les différents doctorants, post-doctorants, alternants, et stagiaires de l'équipe SPI contribuent à établir une atmosphère que j'ai grandement apprécié durant mon temps chez Orange. Plus particulièrement, je tiens à remercier les doctorants et post-doctorants avec qui j'ai partagé un bout de chemin, en commançant par ceux que j'ai rejoint en cours de route, Donald, Loïc, Quentin, Solenn, et Yacine; puis mes camarades de tranchées avec qui j'ai démarré cette thèse, et le soutien de qui m'aura été précieux pendant ces trois dernières années, Adel "les bons plans", Guigui, et Takoua; et ceux qui nous ont rejoint en cours de route, Maxime, Monir, Olivier (le doctorant au cœur de stagiaire), Paul, et Anaïs. Guillaume je pense qu'on peut enfin se permettre de le dire: "les nerfs sont dé–tendus!".
Je tiens enfin à remercier ma famille, qui a su m'accompagner avec patience et encouragements dans cette entreprise. À mes parents, qui auront malgré eux dû se familiariser avec la cryptographie, pour ensuite tenter d'expliquer ce qui m'aura occupé durant ces trois dernières années à qui voudra bien l'entendre. À Adou pour son soutien toujours plein d'enthousiasme, et à Mamito et Coumbo qui complètent les trois mousquetaires, et qui contribuent toujours de près ou de loin à toutes mes aventures.

Je tiens à dédier ce manuscrit à la mémoire d'Yvan Rafflé, qui aura laissé un grand vide dans les couloirs de l'équipe SPI.

Comment obtenir des protocoles d'attestation et d'authentification adaptés pour des objets contraints en terme de capacité de calcul, d'énergie et d'autonomie dans différentes applications de l'Internet des Objets? Comment assurer que de tels protocoles puissent préserver l'anonymat et la protection des données personnelles des utilisateurs, tout en assurant la responsabilité des différentes parties dans l'utilisation des services concernés? Nous allons tenter d'apporter des réponses à ces questions dans cette thèse. La problématique centrale consiste à trouver un compromis entre sécurité et efficacité, tout en considérant la question de la protection de l'anonymat comme étant une problématique intrinsèque de la sécurité des services dans l'Internet des objets. Ce chapitre introduit la problématique qui a abouti aux différentes contributions de cette thèse. Il présente ensuite les différentes solutions développées pour l'authentification et l'attestation anonymes, adaptées pour l'éventail d'objets dans différents cas d'usages de l'Internet des Objets.

Contexte

L'Internet des Objets (IdO) est un paradigme permettant de connecter des objets, des équipements, des capteurs, ou tout autre système doté d'une capacité de connexion, à des réseaux longues et courtes distance. Ce paradigme permet ainsi aux objets de communiquer entre eux de manière autonome, ainsi que de remonter des données spécifiques, souvent mesurées dans leur environnement de déploiement. L'IdO débute dans le monde physique, où des capteurs mesurent et font remonter des données provenant de leur environnement de déploiement telles que la température, le niveau de luminosité, la détection d'un objet, ou d'une personne entre autres, vers une application de traitement des données. Les actuateurs permettent d'agir sur l'environnement, en déclenchant par exemple une fonctionnalité. Ce paradigme a révolutionné plusieurs industries en terme d'applications pratiques, et cette révolution projette le déploiement de près de 29 milliards d'objets connectés d'ici 2022 [ERI]. L'un des enjeux fondamental de son déploiement à grande échelle est la question de la sécurité et de la protection des données personnelles. En effet, nombreuses applications de l'IdO traitent des données permettant par exemple de localiser des personnes et des objets, introduisant ainsi des problématiques de traçabilité. De plus, la sécurité des objets est devenue un enjeu primordial depuis quelques années, notamment dû aux nombreuses attaques perpétuées sur ces objets à faible capacité de calcul. Nous allons présenter dans ce chapitre les problématiques de sécurité et de protection des données personnelles ayant motivé les différentes contributions de cette thèse.

Applications et motivation

Les solutions de l'IdO peuvent être classifiées dans de multiples domaines d'application, tels que la maison connectée (ou maison intelligente), les réseaux électriques intelligents, la ville intelligente, ou les véhicules connectés. Un des domaine les plus prometteurs en terme d'applications pratiques est notamment l'IdO industriel. En effet, l'IdO industriel permet de mettre en place un écosystème permettant de contrôler les automates industriels à distance.

- **La maison connectée:** les appareils électroniques et systèmes de contrôle des paramètres tels que la luminosité, la température, et la qualité de l'air de la maison sont reliés à un système central. Le système peut aussi dans certains cas contrôler les différents niveaux d'accès de la maison, nécessitant ainsi l'implémentation de mécanismes assurant la sécurité et l'intégrité du système;

- **Le réseau électrique intelligent:** les réseaux électriques déployés aujourd'hui font appel à technologies de l'information et de la communication pour fournir un service efficace, sécurisé, et plus adapté aux contraintes écologiques actuelles. Ils permettent aussi le suivi détaillé de la consommation électrique individuelle et collective, notamment par le biais de relevés plus détaillés sur la consommation des foyers. Ceci conduit à des problématiques de protection des données des utilisateurs;

- **Les voitures connectées:** les systèmes de transport intelligents sont déployés au moyen de capteurs présents dans les véhicules, permettant la remontée des données provenant de l'environnement du véhicule, mais aussi la mise en place de réseaux de communications entre les véhicule nommé *Vehicular Ad Hoc Network* (VANET). Les réseaux VANETs permettent ainsi aux véhicules de transférer de l'information, sur par exemple l'état du trafic en temps réel. Ces nouvelles pratiques ne doivent en revanche pas compromettre l'anonymat du véhicule et de son utilisateur, notamment en permettant de tracer les personnes.

- **L'IdO industriel:** la gestion des systèmes de contrôle industriels est effectué par le biais d'automates de contrôle tels SCADA. Ils interviennent par exemple dans les centrales de gaz et d'électricité, ou dans les usines de fabrication de voiture. Garantir la sécurité et l'intégrité de ces automates est un enjeu primordial pour éviter des cyber-attaques avec des conséquences potentiellement désastreuses humainement et matériellement.

Problématique, défis, et enjeux

Besoins et défis

Les objets dans l'IdO varient en terme de capacité de calcul, de mémoire, de source d'énergie, et d'application. Les objets les plus contraints, tels que les capteurs et les actuateurs, peuvent être utilisés comme composant d'objets plus sophistiqués tels les voitures connectées. Ces derniers objets, moins contraints en terme de capacité de calcul, présentent aussi des contraintes propres à l'écosystème de l'IdO comme par exemple des contraintes en terme de bande passante du réseau

de communication, d'anonymat des communications. Dans cette thèse, nous allons en général faire référence à un objet défini de la manière suivante: un système embarqué avec des capacités de communication, exécutant une tâche bien définie de manière autonome. Les problématiques de sécurité des objets et des communications entre groupes d'objets, de protection des données personnelles, et de performance des protocoles de sécurité adoptés sont au cœur des travaux de cette thèse. Nous les présentons brièvement ci-dessous:

- **Intégrité:**
  L'intégrité des objets et des communications est primordial dans nombreuses applications de l'IdO, et notamment de l'IdO industriel. En effet, les systèmes de contrôle industriels traitent des données sensibles en terme de sécurité, et ont été la cible de cyber-attaques ces dernières années [FMC10, MD16]. Ces attaques sont perpétrées à distance, et consistent essentiellement à injecter des malware dans des systèmes, les rendant ainsi vulnérables ou tout simplement non fonctionnels. De plus, les données renvoyés par des systèmes compromis peuvent compromettre le fonctionnement de tout le réseau. Assurer l'intégrité des systèmes et des données contribue à assurer la confiance dans les réseaux et les applications de l'IdO, et est donc ainsi devenu un enjeu principal de la sécurité de l'IdO.

- **Disponibilité:**
  En fonction des cas d'usages, les objets dans l'IdO envoient et reçoivent des données en temps réel. Ceci introduit des contraintes de disponibilité des objets, pour notamment assurer l'intégrité des communications dans le réseau. Les solutions pour assurer la disponibilité des objets contribuent à la protection contre les attaques par déni de service (DoS) notamment.

- **Privacy:**
  L'adoption de nombreuses règlementations en matière de protection des données personnelles ces dernières années [EU, PIP16, OPC19] a notamment mis l'accent sur la nécessité d'adopter des processus d'authentification anonymes des objets et des utilisateurs dans différentes applications de l'IdO. En effet, les données récoltées en fonction du cas d'usage peuvent engendrer des problématiques de privacy telle que la traçabilité des personnes par exemple. Des solutions cryptographiques telles que les signatures de groupe ou les schémas d'attestation anonymes permettent d'assurer l'authentification respectueuse de la vie privée.

- **Responsabilité:**
  En parallèle de la protection de l'anonymat et des données des usagers, la responsabilité de chaque usager doit être engagée lors de l'utilisation des différents services de l'IdO. En effet, assurer l'anonymat ne doit pas permettre aux usagers de nier la responsabilité des communications qui leurs sont attribuées.

- **Performance:**
  Différentes applications de l'IdO font appel à différents objets aux capacités et fonctionnalités variées. Le consensus étant que tout développement de protocoles de sécurité pour l'IdO doit

tenir compte des contraintes en matière de capacité de calcul, de source d'énergie, et de mémoire. Lorsque l'on considère les objets comme des systèmes d'un réseau de communication, ces contraintes s'étendent aux capacités du réseau, notamment en terme de bande passante.

Dans cette thèse, nous nous intéressons aux problématiques d'authentification, de contrôle d'accès, et d'attestation de l'intégrité des objets, dans le respect de la vie privée. Nous nous intéressons notamment aux mécanismes cryptographiques d'authentification qui protègent contre la traçabilité tout en assurant la non-répudiation dans différents cas d'usages.

Dans la suite de ce chapitre, nous présentons les défis en terme de sécurité, de protection des données personnelles, ainsi que les solutions cryptographiques qui peuvent être adoptées. Nous introduisons ensuite les différentes contributions de cette thèse.

Défis et enjeux

Les multiples angles d'attaques des objets dans l'IdO en font des cibles idéales pour des attaques à distance. Les objectifs de sécurité identifiés pour les systèmes traditionnels sont valables pour les systèmes embarqués, avec des contraintes encore plus fortes.

Identification et Authentification. L'authentification est un mécanisme permettant de prouver l'identité d'une partie dans un protocole de communication. En effet, l'identification seule ne suffit pas pour s'assurer de l'intégrité des communications. Les parties doivent pouvoir s'authentifier mutuellement, de manière non ambiguë, assurant la responsabilité et la non-répudiation de chaque partie. Les solutions cryptographiques permettant l'authentification des différentes parties dans un protocole de communication sont les schémas de signatures. Il s'agit de cryptosystèmes à clé publique, permettant de vérifier et prouver l'identité assignée à un message donné, sur le réseau. La partie signataire n'est pas en mesure de renier sa signature une fois celle-ci générée sur un message. Dans l'IdO, l'authentification est une étape cruciale pour s'assurer non seulement de l'identité des objets remontant des données spécifiques, mais aussi pour s'assurer de l'intégrité des communications une fois cette identification établie. Les schémas de signature à clé publique posent la contrainte forte qu'ils nécessitent parfois des calculs couteux, qui ne sont pas à la portée des objets considérés dans l'IdO.

Attestation. En plus de s'assurer de l'intégrité des communications dans l'IdO, il est primordial de s'assurer de l'intégrité des objets eux-mêmes, notamment pour des applications dans l'IdO industriel, où la compromission des automates pourrait engendrer des conséquences graves. De plus, l'intégrité des communications et des données remontées par un objet repose sur l'hypothèse qu'il n'est pas victime d'une attaque à distance telle une injection de malware. Ceci pourrait en effet compromettre l'intégrité et donc le fonctionnement du réseau tout entier. L'intégrité des objets est mesurée par l'état interne de leur micro-logiciel. Cet état doit être surveillé périodiquement, pour pouvoir ainsi détecter de potentielles attaques. Le mécanisme d'attestation permet à un système donné de fournir la preuve de son état

interne. L'attestation à distance est un mécanisme de sécurité introduit par l'informatique de confiance, permettant à des entités de confiance de surveiller l'état des systèmes sous leur contrôle par le biais d'attestations périodiques fournies par les dits systèmes.

Solutions cryptographiques pour l'authentification et l'attestation

La cryptographie a introduit de nombreuses solutions pour l'authentification par le biais de schémas de signature électronique [Sha84a, Sch89, MPSW19, BLS01], et les protocoles d'authentification anonymes qui en découlent tels que les schémas de signatures de groupe [CH91, CS97], les schémas de signatures aveugles [Cha83], les protocoles d'accréditations anonymes [CL01], et les protocoles d'attestations anonymes [BCC04]. Ces solutions permettent de mettre en place des protocoles d'authentification anonymes sûrs, dont la sécurité peut-être prouvée de manière efficace basée sur des hypothèses mathématiques et cryptographiques classiques.
Dans cette thèse, nous nous intéressons particulièrement aux protocoles d'attestations anonymes (DAA), qui sont des protocoles cryptographiques largement déployés, notamment par le biais du protocole de DAA EPID proposé par Intel [BL11a].

Protocoles d'attestation pour l'authentification individuelle: L'attestation anonyme (ou DAA) permet à un système de prouver l'intégrité de son état interne à une autorité de confiance. Pour cela, le protocole contient deux parties, une première partie où le système s'authentifie et obtient une clé de groupe, et une deuxième partie où il peut émettre des signatures sans que celles-ci ne puissent être tracées. Comme nous allons le démontrer par la suite, les schémas de DAA peuvent donc être utilisés comme protocole d'authentification préservant l'anonymat des différentes parties. La première contribution de cette thèse consiste à développer un schéma d'attestation anonyme efficace, permettant l'authentification sûre dans des environnements contraints.

Protocoles d'attestation de groupe: L'architecture de l'IdO est telle que l'authentification et la vérification de l'intégrité des objets doivent être considérés au-delà du seul cas individuel. En effet, les objets sont souvent déployés en groupe, permettant ainsi l'exécution d'une ou plusieurs tâches nécessitant la collaboration de plusieurs systèmes. On parle alors d'essaims d'objets. Les protocoles d'attestation de groupe permettent de vérifier l'intégrité de groupe d'objet lors d'une seule exécution du protocole. Ils permettent une gestion efficace et coordonnée de la sécurité des objets. Dans la deuxième partie de cette thèse, nous développons un schéma d'attestation de groupe permettant de détecter de manière précise les objets fournissant des attestations erronées.

Contributions de cette thèse

Nous présentons ici les travaux effectués durant cette thèse, portant sur l'attestation et l'authentification anonymes des systèmes pour différents cas d'usages de l'IdO. Les différentes contributions seront présentées en détail dans la suite de ce mémoire.

Contribution C1: Une nouvelle primitive d'attestation anonyme

Les différentes contraintes en termes de capacité de calcul et de performance des objets dans l'IdO nous a conduit à développer une nouvelle primitive d'attestation anonyme adaptée pour des environnements contraints.
Le pré-DAA est une variante efficace des DAA ne nécessitant pas de délégation de calculs. En effet, lors de la génération des signatures de DAA, le module de sécurité délègue certains calculs couteux à l'objet qui le contient comme par exemple un téléphone ou un PC. Ce processus de délégation de calcul induit des préoccupations en terme de sécurité et de traçabilité, notamment en considérant le fait qu'un téléphone ayant été sujet à une attaque de malware pourrait introduire dans chaque signature d'attestation un élément permettant de tracer la signature à l'utilisateur.
Le chapitre 4 introduit un nouveau schéma de pré-DAA efficace, qui peut être exécuté par un module contraint en terme de capacité de calcul et de mémoire, tel une carte à puce. Nous prouvons la sécurité de notre nouveau schéma de pré-DAA dans le modèle de l'oracle aléatoire, en se basant sur une variante de l'hypothèse q-Strong Diffie-Hellman (q—SDH).
Dans la deuxième partie de ce mémoire, nous utilisons notre schéma de pré-DAA comme brique de base pour la construction de protocoles d'authentification efficaces et anonymes.

Contribution C2: Un protocole pour l'authentification anonyme dans VANET

Le chapitre 5 introduit un protocole d'authentification décentralisé et anonyme pour les communications de voiture à voiture (*Vehicle-to-Vehicle* (V2X)) dans les réseaux VANET. Notre protocole permet la transmission de messages CAM (*Cooperative Awareness Messages*), qui contiennent des informations sur le trafic, dans le réseau. L'authentification des voitures ne nécessite pas une autorité de certification centrale, permettant ainsi de résoudre les problématiques d'encombrement de la bande passante. Notre protocole est basé sur notre schéma de pré-DAA présenté au chapitre 4, assurant ainsi une racine de confiance de l'émission des messages à leur réception sur le réseau. L'authentification des messages est ainsi assurée, ainsi que leur intégrité. Les propriétés de sécurité du schéma sous-jacent assurent la propriété de non-répudiation. Les travaux de ce chapitre ont donné lieu à la publication [DDR+19].

Contribution C3: Un protocole de contrôle d'accès sur mobile dans les réseaux de transports

Le chapitre 6 introduit un protocole d'implémentation de pass de transport sur mobile. Notre protocole assure la non-traçabilité des usagers, par le biais d'un protocole anonyme d'authentification sur le réseau. Un usager souscrit à un abonnement au mois ou à l'année, et reçoit une accréditation sous forme de clé de groupe lui permettant de s'authentifier de manière anonyme sur le réseau. Le protocole est basé sur le schéma de pré-DAA introduit au chapitre 4. En effet, les propriétés de notre schéma de pré-DAA assurent les fonctionnalités suivantes:

- **Anonymat:** l'identité de chaque usager ayant validé un pass à un instant donné ne peut pas être retrouvée, même par l'opérateur de transport;

- **Non-traçabilité sauf en cas de détection de fraude:** différentes signatures générées par un même pass ne peuvent être liées entre elles, ou à un même utilisateur, sauf en cas de détection de fraude;
- **Détection de fraude:** la propriété de liaison du schéma de pré-DAA sous-jacent permet de détecter des tentatives de fraude, notamment l'utilisation consécutive d'un même pass par deux utilisateurs différents.

L'efficacité de notre protocole assure qu'un utilisateur est capable de s'authentifier en moins de 200 ms pour un mobile allumé. De plus, la propriété de non-délégation de calcul au téléphone stipule que la carte à puce intégrée effectue tous les calculs d'authentification de manière autonome, et peut ainsi s'authentifier même lorsque le mobile est éteint, bien qu'avec des performances plus dégradées dans ce cas précis. Les travaux de ce chapitre ont été présentés au symposium Real World Crypto 2020, et font l'objet d'une soumission dans [DDT].

Contribution C4: Un protocole d'attestation de groupe d'objets

Dans la troisième et dernière partie de ce mémoire, nous nous intéressons de plus près à l'intégrité des objets eux-même. En effet, l'utilité première des schémas d'attestation est de prouver l'intégrité des systèmes physiques. Ces schémas ne s'appliquent cependant qu'au cas de l'attestation individuelle. Asokan et.al. [ABI+15] ont introduit le premier schéma d'attestation d'essaims d'objets, permettant de vérifier l'intégrité collective d'un groupe d'objets formant un réseau. Les différentes solutions depuis ont été basés sur différents schémas de signature et de MAC agrégés, fournissant ainsi une réponse quand à l'état global du groupe. Cependant, aucun schéma d'attestation d'essaim d'objets ne permettait jusque-là d'identifier l'origine d'une fausse attestation dans le groupe. Dans le chapitre 7, nous introduisons CoRA, un nouveau schéma d'attestation d'essaim d'objets, basé sur un schéma de MAC algébrique agrégé $\mathsf{MAC}_{\mathsf{BLS}}$. CoRA permet de détecter l'origine de l'attestation erronée dans le cas où la vérification de l'intégrité du groupe échoue. Notre solution se base sur les propriétés algébriques du schéma $\mathsf{MAC}_{\mathsf{BLS}}$ sous-jacent. Les travaux de ce chapitre ont été publié dans [DLLT20].

Table of Contents

ACRONYMS

CA Certification Authority.

CAM Cooperative Awareness Messages.

CDH Computational Diffie-Hellman.

CPS Cyber Physical Systems.

DAA Direct Anonymous Attestation.

DDH Decisional Diffie-Hellman.

DL Discrete Logarithm.

DoS Denial-of-Service.

ECC Elliptic Curve Cryptography.

HSM Hardware security module.

ICS Industrial Control System.

IdO Internet des Objets.

IIoT Industrial Internet of Things.

IoT Internet of Things.

ITS Intelligent Transportation System.

LRSW Lysyanskaya Rivest Sahai Wolf.

MAC Message Authentication Code.

NFC Near Field Communication.

NIZKPK Non-Interactive Zero-Knowledge Proof of Knowledge.

OBU On-Board Unit.

PKI Public Key Infrastructure.

PLC Programmable Logic Controller.

PUF Physically Unclonable Function.

q**—SDH** sq-Strong Diffie-Hellman.

RA Remote Attestation.

RoT Root of Trust.

SA Swarm Attestation.

SGX Software Guard Extension.

SoK Signature of Knowledge.

TCG Trusted Computing Group.

TEE Trusted Execution Environment.

TPM Trusted Platform Module.

UICC Universal Integrated Circuit Card.

V2X Vehicle-to-Everything.

VANET Vehicular Ad Hoc Networks.

ZKPK Zero-Knowledge Proof of Knowledge.

NOTATION

$x \in X$	The element x is in the set X
$x \xleftarrow{\$} X$	x is chosen uniformly at random from the set X
$\{0,1\}^*$	Set of all binary strings of arbitrary finite length
$\{0,1\}^k$	Set of all binary strings of length k
$\mathbb{Z}_p$	Set of positive integers less than $p-1$
$\{x_i\}_{i=1}^{l}$	The set of elements $(x_1, x_2, ..., x_l)$
$x \leftarrow y$	x is assigned the value y
$\mathbb{G}$	Cyclic group of prime order
$1_{\mathbb{G}}$	Multiplicative identity element of group $\mathbb{G}$
$0_{\mathbb{G}}$	Additive identity element of group $\mathbb{G}$
$k\mathbb{G}_i$	k exponentiations in the group $\mathbb{G}_i$
$k\mathbb{G}_i^j$	k $j-$multi-exponentiations in the group $\mathbb{G}_i$
λ	Security parameter
1^λ	Security parameter in binary notation
pp	Public parameters
$Pr[A]$	Probability of event A
ID_U	Identifier of user U
$\mathscr{I}$	Issuer
$\mathscr{SE}$	Secure element (SIM card)
$\mathscr{V}$	Verifier
$\mathscr{A}$	Adversary
$\mathscr{O}$	Query oracle

$\mathscr{A}^{O}(x)$	Adversary $\mathscr{A}$ takes as input x and has access to oracle $\mathscr{O}$
$\mathscr{B}$	Reduction
$\mathscr{C}$	Challenger

CHAPTER 1

INTRODUCTION

The Internet of Things introduced a new computing paradigm which places resource-constrained devices at the heart of the computing framework. Its applications range from industrial to consumer-driven, including commercial and services, providing us with new and efficient ways of controlling our environment. This new ecosystem requires a new analysis of the security of the systems, as well as the privacy of users and their data. As with the traditional computing ecosystem, building secure IoT systems should not hinder the efficiency of the corresponding applications and the privacy of end users. The widespread deployment of IoT applications, specifically consumer-driven applications, is hindered by the privacy-related concerns of existing security solutions. The aim of this thesis is to overcome these challenges by developing new privacy-preserving authentication protocols that are optimally efficient for embedded systems, as well as ensuring the integrity of said systems with a new device attestation construction.
In this chapter, we provide the motivations behind developing privacy-preserving authentication and attestation protocols in order to build secure IoT systems.

Contents

1.1 A New Computing Paradigm

1.1.1 Next-generation industries and control systems

The Internet of Things (IoT) describes an ecosystem in which a collection of embedded and mobile devices, that can sometimes scale up to thousands of devices, form a wireless network with communication, detection, and actuation capabilities. Devices are equipped with a unique identifier (UID), and are capable of sending and receiving data autonomously across the Internet without any human intervention. IoT applications are growing fast and are increasingly permeating every aspect of our lives, as the number of connected objects is expected to reach 29 billion by 2022 [ERI]. The underlying control systems provide a link between the physical world and the digital world, and their application domains are as vast as there are industries. IoT applications can be classified by solution domain, namely Smart Home, Smart Grid, Smart City, and Automotive. A more recent field has emerged for the management of Industrial Control Systems (ICS) for utility providers, notably controlling processes and systems such as gas centrifuges. It is known as the Industrial Internet of Things (IIoT).

1.1.1.1 Smart Home

On the consumer-specific end of IoT applications, home automation systems control all aspects of home life including lighting, temperature, air quality, appliances, secure access control. Robust security mechanisms must be implemented in order to prevent outside attackers from compromising the control systems [JBC16, KSH16], as well as to protect user privacy from unlawful data collection and location monitoring [CWC13]. Specific Smart Home applications include:

- *Access Control* implements a security system that defines a fine-grained access control policy for the home, by level of permission;

- *Remote Appliance Monitoring* enables the remote control (turn on/off, status check) of devices via a connected user interface. It also integrates with the Smart Grid via a user interface for smart meter control;

- *Leakage Detection* will implement an intelligent gas, smoke, or water leak detection system controlled by the central control system.

1.1.1.2 Smart Grid

The deployment of new intelligent electrical grids has led to more efficient energy production, transportation, and distribution for utility services. It also grants users a more hands on control over their energy consumption data. Smart Grid Applications include:

- *Smart Energy Production Management* implements a system for exchanging data between energy production sites and the transportation and distribution sites. This allows the different actors to streamline the new production systems, notably those participating in the energy production from renewable sources;

- *Smart Metering* grants end users fine-grained access to their consumption data. Indeed, by way of new and connected meters, consumer data can be monitored at all times, providing more accurate consumption data, which in turns results in a more efficient billing system for utility providers;

- *Electric Vehicle Charging (EVC)* allows users to locate the nearest charging station in their vicinity. EVCs enable more efficient energy management by tailoring users' charging need to the nearest power supply.

1.1.1.3 Smart City

The development of Smart Cities with the monitoring of urban infrastructures and traffic, has revolutionized services and their impact in urban areas. Smart City applications include:

- *Smart Street Lighting* enables the automated control of street lighting with respect to the environmental data forwarded by sensors;

- *Environmental Control* implements air quality control, as well as pollution level monitoring in cities by way of sensors;

- *Traffic Control* applications aim to collect and process transportation data collected from vehicles and roadside units, in order to reduce travel time. The transportation data may notably be used to provide parking applications indicating the closest parking spaces to users, which results in a better management of infrastructures and resources;

- *Smart Building* applications enable the tracking of user and environmental data in order to for example tailor electricity supply to the demand (e.g. thermostats with motion sensors). It also include access control systems in order to ensure the security and privacy of people and data.

1.1.1.4 Intelligent Transportation Systems

The automotive industry has undoubtedly benefited the most from the Internet of Things, from production lines being impacted by intelligent control systems, to the deployment of networks of connected vehicles increasing safety and efficiency on the road. Automotive applications include:

- *Connected Vehicles* increase safety on the road by implementing a communication network between vehicles equipped and roadside units. They are equipped with Internet access and cellular radio, and are able to send real-time Cooperative Awareness Messages (CAM) including road hazard notifications, crash notifications, and congested areas. This novel communication system is also known as Vehicle-to-Everything (V2X) communication;

- *Pay-As-You-Drive (PAYD)* is a usage-based insurance service which determines a user's vehicle insurance plan based on the type of vehicle, as well as driving style based on speed and driving data collected in real-time;

- *Smart Ticketing* applications allow commuters to purchase and store and use transport tickets and subscribe to transit passes using the NFC capabilities of their mobile phones. Smart ticketing provides a flexible and secure ticketing system, with advantages for both the transport operator and the user. Indeed, the transport operator collects validation data, which allows him to perform data analysis on validation data at different stations in order to optimize the public transport system. It is for example possible to more accurately anticipate peak hours at different stations. It also provide users with a fast, accurate, secure, and paperless validation process;

- *Electrical Toll (eToll)* enables vehicles equipped with a radio transponder to pay their toll fairs without waiting at tollbooths. This eliminates delays and congestion at tollbooths;

- *Fleet Management* helps minimizing the risks inherent to applications with a significant vehicle investment. It improves efficiency and productivity by leveraging traffic data and vehicle location. For example, asset tracking allows cities to provide trash collectors with the most efficient routes hence reducing transportation time and cost;

- *Stolen Vehicle Tracking (SVT)* is a service which allows users to track their stolen vehicle using IoT networks.

1.1.1.5 Intelligent Industrial Control Systems

Industrial Internet of Things applications encompass all applications mentioned above. In this thesis, we mainly consider the Industrial Internet of Things (IIoT) to include all applications affecting the management of Industrial Control Systems ICS. This includes the automation of production lines in factories, as well as the introduction of intelligent control systems in industrial power plants. Indeed, industrial parks are complex, and employ a vast number of control systems such as Supervisory Control And Data Acquisition (SCADA), to monitor other electrical and embedded systems deployed in the field.

1.1.2 IoT systems: requirements and key challenges

Devices in IoT vary in terms of hardware specifications, applications, networking properties, computational capabilities and memory. They range from constrained sensors and actuators, to

vehicles with built-in sensors and high-end secure trusted hardware. The baseline being that all devices have been assigned an IP address and have the ability to collect and transfer data over a network without any need for human intervention. A connected device is a public application of an embedded system with communication capabilities. Embedded systems are physical and computational systems which are able to autonomously perform a specific task. They are part of a larger ecosystem with specific functional and security requirements. As depicted in Table 1.1, when designing security schemes for IoT applications, we must consider different characteristics of the application by order of priority. We distinguish the following requirements and key challenges in the deployment of devices in IoT applications:

Data and Device Integrity

The integrity of embedded systems is notably of critical importance in IIoT applications [SWW15, LXL+12]. Indeed, ICSs process security-critical and privacy-sensitive data, which makes them vulnerable to targeted cyber attacks [FMC10]. These systems are deployed in sometimes inaccessible and uncontrolled environments, further complicating the implementation of efficient security mechanisms to protect them against remote malware attacks. Moreover, the large number of deployed devices requires more scalable methods to verify the integrity of devices and the data they communicate over the network. IoT devices are the target of well known malware attacks due to these constraints. In 2010, a 500-kilobyte computer worm Stuxnet [FMC10] infected the software of SCADA systems and Programmable Logic Controllers used to control industrial processes and gas centrifuges in Iran. The Dyn Distributed Denial-of-Service (DDoS) attack [AvRDN19] was a botnet composed of a collection of infected IoT devices. Critical security challenges in IoT applications primarily result from attackers exploiting the hardware constraints of the devices themselves [Pol10, ZCNC11, MR12, MV14, HABJ].

Availability

The secure access to data in all application domains must be ensured in real-time. For example, in order to prevent any delay in production (which would induce loss of productivity), devices in industrial production must be reachable and in a trustworthy state. Availability ensures that authorized entities always have access to devices and data when needed. This property is particularly important in IIoT applications, as the lack thereof could potentially result in safety issues. Methods to ensure availability include protection against Denial-of-Service (DoS) attacks. Availability is closely linked to data integrity, as interactions must be associated with a specific user, with no ability to forge authentication or to tamper with authenticated data.

Accountability

Users of a system must be accountable for their own actions. This property is especially critical in ITS applications, where it must not be possible for users to deny sending specific messages or having been granted access to services or data for liability issues.

Privacy

The variety of applications imply that IoT devices process varied and large amounts of data. Such data may be sensitive in nature, for example providing information on user location and habits in the case of Home Automation and Intelligent Transportation System applications. Exploiting said data may lead to the invasion of user privacy in consumer-driven applications. The privacy of users and their data must be ensured in the long run. The need for privacy-preserving solutions for device authentication and attestation in IoT is increasing. Such solutions not only allow end users to have control over their data, but they also allow service providers who leverage IoT technologies to provide better services to be compliant with new privacy regulations. Indeed, new laws have been adopted in recent years in order to protect user data privacy. An example of such laws is the European General Data Protection Regulation [EU] adopted in 2016 by the European Commission. These regulations aim to protect user personal data from unauthorized collection and exploitation by public and private services alike. They motivated the development of privacy-enhancing cryptographic techniques for user authentication and access control, but also for ensuring data integrity in order to prevent tampering. Considering these new regulations, new protocols designed to secure communications between IoT devices must include "privacy-by-design" mechanisms.

Efficiency

Devices in IoT applications are optimized for a specific task, and as opposed to traditional computing systems, they present a multitude of constraints such as production cost, energy consumption, computational capabilities and limited memory. When deployed as part of a larger IoT application, they also present specific connectivity and communication constraints pertaining to the network bandwidth, as well as the radio technology used in said application.

In this thesis, we will take a closer look at the specific privacy problem of personal location, whether it pertains to the autonomous devices in IoT networks, or the end users who might be affected. In particular, the aim is to develop privacy-preserving cryptographic solutions for device authentication, access control, and attestation, that reveal the minimal amount of personal information required for the secure deployment of each service. In addition, these new mechanisms provide solutions for the efficiency challenges inherent to the IoT ecosystem.

The remaining of this chapter is organized as follows. We introduce the research challenges in designing secure and privacy-preserving cryptographic protocols for IoT devices, and discuss the current cryptographic mechanisms used to build said solutions. We then summarize the contributions of this thesis.

Application	Communication model	Limited bandwidth	Limited energy	Scalability	Delay Constraint	Privacy
Pay-as-you-drive	One-to-One				✓	
eToll					✓	✓
Smart Ticketing			✓		✓	✓
Smart metering	One-to-Many	✓	✓	✓	✓	✓
ICS	Many-to-One	✓	✓	✓	✓	
Smart Home		✓	✓			
Smart Grid					✓	✓
Fleet management				✓		✓
Smart city	Many-to-Many	✓	✓	✓		
Connected vehicles		✓		✓	✓	✓

Table 1.1: IoT Applications Characteristics and Requirements

1.2 Security and Privacy Challenges and Cryptographic Solutions

IoT systems present security challenges due to the variety of attack surfaces, from the device layer to the application layer. In this thesis, we focus on application layer security and privacy. In this section, we first formulate the security and privacy challenges which motivated the contributions in this thesis, as well as the existing cryptographic solutions and security frameworks, prior to formulating the subsequent research goals.

1.2.1 Security and Privacy Objectives

Confidentiality. Cryptographic mechanisms are used to provide data confidentiality, data integrity, and secure device authentication of mobile and embedded devices in different IoT applications. Devices exchange sensitive data on wireless communication channels, which are vulnerable to eavesdropping attacks. In traditional systems and networks as well as IoT networks and systems, *symmetric key* encryption schemes ensure data confidentiality. A symmetric key encryption scheme enables two parties to exchange data without any other external party being able to read said message. The encrypted messages are indistinguishable from a random string to external parties, and the encryption process makes use of a common and agreed upon secret session key. *Public key* algorithms enable the distribution of the secret session key. In public key cryptography, each party possesses a public/private key pair, the former is known by every party in the system, while

the latter is only known to the owner. Implementing cryptographic schemes on embedded devices comes with its own set of challenges. Indeed, cryptographic algorithms require significant computational power, energy, and memory, which are not always met by embedded devices. Cryptographic schemes ensuring data confidentiality, also known as encryption schemes, are mature and robust. Depending on the application, standardized encryption algorithms are used to provide secure communications between IoT devices as well. In fact, in order to incorporate the efficiency, computational capacity and storage space of these new computational system, new sets of standards have given way to what is commonly known as *Lightweight Cryptography*, which includes lightweight block ciphers [BKL+07, BSS+13, MBSM16], and lightweight public key schemes [HPVP11, PH12]. The ISO/IEC 29192 standard [ISO] specifies three block ciphers suitable for IoT/M2M applications which require lightweight cryptography implementations.
Confidentiality is a low priority challenge in many IoT applications due to the development of lightweight encryption schemes. In fact, ensuring the authenticity of devices as well as the authenticity and integrity of the data exchanged between devices in various IoT applications is an important challenge [RPH06, PBH+08, SWW15]. The remaining of this section will discuss the importance of authentication and attestation as security services in the IoT ecosystem.

Identification. In the IoT ecosystem, mutual identification between devices is a key security property. The aim being devices to be able to identify each other and the data being sent on the communication channel, in order to trust those data. For controlling devices, identification allows controllers to identify which devices they control, as well as to accurately forward a certain control command to the concerned device. In access control applications, it is mandatory to identify which device or user is requesting access, in order to grant the proper access credentials. Essentially, the ecosystem is composed of autonomous devices, where trust is derived from complete knowledge of the systems involved.

Authentication. Authentication is a security mechanism which enables devices and users to prove their identities in the communication process. Indeed, identification alone does not provide complete trust in the system, as the system must prove said identity to be correct. From a security standpoint, authentication can be provided by signature schemes in the public key model. Signature schemes are public key schemes which allow a sending entity to authenticate with a verifying entity, hence ensuring to each party involved in the protocol that messages originate from the expected sender. The signature verification step is public, and can be undertaken by any party. A key property of signature schemes is the fact that the party generating the signature scheme should not be able to *repudiate* its signature upon its generation. In our case, IoT devices must prove their identity in a secure manner, in order to ensure trust in the data communicated over the network. The authentication proof is verified by a mutually trusted authority, who can be for example a certification authority in the case of public key schemes.

Data/Device Software Integrity Finally, ensuring that messages originating from an authenticated device have not been tampered with is a key security requirement in IoT. In a given application, the integrity of data exchanged between devices must be ensured in order to trust the overall system. In addition to data integrity, the internal software state of devices in an IoT

application must be periodically verified. Devices in a number of IoT use cases are deployed in remote locations which are inconveniently accessible, rendering the process of local attestation (e.g. Secure boot) often impossible. Remote attestation allows control systems to remotely monitor the software state of their devices. It is a security service at the core of the IoT ecosystem, and its analysis constitutes an important part of the work in this thesis. We discuss the concept of remote attestation in Section 1.2.4.

Authorization. In access control applications, once identified and authenticated, devices can be authorized to access trusted systems or data. A number of IoT applications implement authorization and access control functionalities which leverage embedded devices, including Home Automation, Smart Vehicles, and Mobile Access Control Systems. These systems must identify, authenticate, and grant access to authorized parties, whilst protecting the systems from fraudulent behavior, and from unlawful data collection which can lead to privacy breaches.

Efficiency. Cryptographic protocols designed for embedded systems must minimize the required hardware size (circuit size, ROM/RAM sizes), the computational cost, and energy consumption. They must also provide optimal processing speed (throughput, delay), while minimizing computational and communication overhead. Indeed, these protocols by nature generate overhead, which causes scalability problems in networks and systems incorporating embedded devices. Considering the trade-off between security and efficiency is therefore an important aspect to factor in the design of secure and privacy-preserving protocols.

1.2.2 Security architectures for embedded systems

The variety of IoT applications and the embedded systems deployed in those applications result in a rich body of literature on security mechanisms and architectures for embedded IoT systems. In this thesis, we particularly focus on the following systems:

- Trusted Execution Environments (TEE) at the root of Intel and ARM security architectures. Different security architectures exist for these systems: Trusted Computing based on secure hardware (e.g. Trusted Platform Module (TPM) [TPM19]), Physically Unclonable Function (PUF) [MS11, MBM+17], and Intel Software Guard Extension (SGX) [MAB+13a, CD16].

- Low-end embedded systems such as smart cards, and more specifically Universal Integrated Circuit Cards (UICC) commonly known as Subscriber Identity Module (SIM) cards [ETS]. Smart cards are embedded in many devices (mobile phones, smart meters), and provide an architecture capable of implementing cryptographic algorithms and security services for devices that are not equipped with high-end hardware modules. In particular, in this thesis we will introduce cryptographic mechanisms that leverage the features of constrained trusted environments such as SIM cards, to provide services such as device attestation, that are built-in more sophisticated systems.

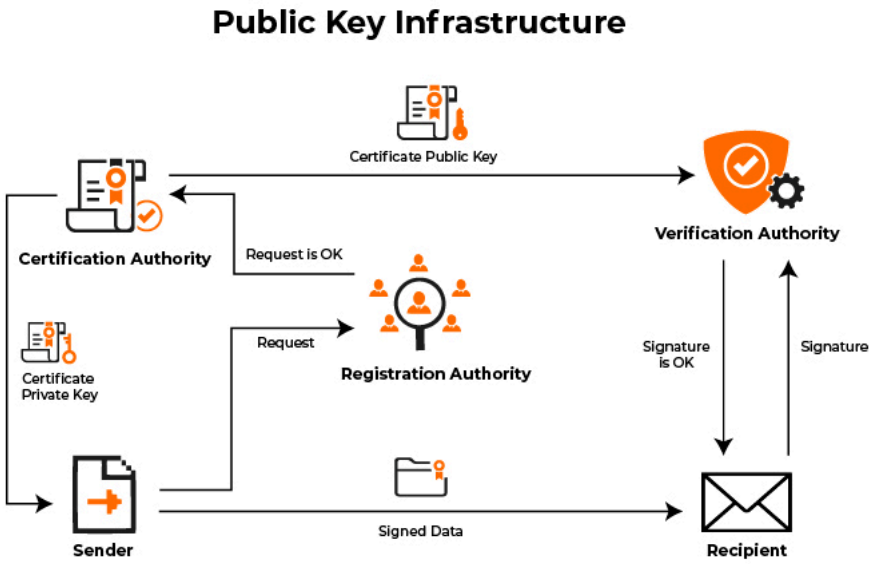


Figure 1.1: Public Key Infrastructure Architecture

1.2.3 Authentication

The first part of this thesis provides secure and privacy-preserving authentication solutions for mobile and embedded devices. In this section, we briefly describe authentication based on cryptographic primitives. We discuss the security and privacy challenges, as well as the efficiency challenges induced by the implementation of cryptographic algorithms on the aforementioned systems.

1.2.3.1 Security architecture

Public Key Infrastructures. Threats affecting IoT systems include impersonation, whereby attackers take control of a given device, and communicate with other devices using the identifier of an honest user. Cryptography provides us with the tools to enforce mutual authentication, requiring each party to securely prove its identity prior to sending messages over the communication network. A Public Key Infrastructure (PKI) is a set of hardware, software, policies, and procedures required to create, manage, store, and distribute digital certificates and public keys. The PKI is an essential service in authentication schemes and provide the elements for a secure and trusted environment for the Internet of Things, as it establishes the identity of devices and users, as well as a controlled access to systems and services. A PKI operates with a centralized architecture as depicted in Figure 1.1, where public keys are bound to their users using digital certificates. It comprises a Certification Authority (CA), which essentially acts as a trusted third party, in charge of generating the credentials used to identify devices. They bind a given device to its public key in a digital certificate. Certificate management in the IoT world presents a number of challenges.

Challenges. The millions of devices potentially involved in IoT applications lack a centralized and scalable system for managing keys and identities. The periodic enrollment of devices

requires a highly scalable identification and authentication system, which does not induce delay and overhead over the communication system. Traditional PKIs require a scalable certificate management system, which is exacerbated in the case of millions of devices requesting certificates and authenticating at the same time on the network. As we will introduce in Part II of this thesis, the management of millions of certificates for Verification Authorities hinders the large scale deployment of vehicular ad hoc networks for example.
In addition, PKIs introduce privacy concerns, as they bind a device to a given certificate, revealing his identity each time a device signs a message on the network, or requests access to a system. The particular privacy issues are increasingly being monitored, in order to comply with new laws regarding user data privacy [EU].

1.2.3.2 Privacy-preserving and decentralized authentication

There is a two-fold privacy concern in the deployment devices in IoT applications: (1) ensuring the authenticity of devices and communications in order to comply with the new trust model, without revealing any additional information about devices and their users; (2) verifying the integrity of devices' software states without revealing their respective identities in the process, notably in order to avoid targeted malware attacks.
Different cryptographic schemes allow for decentralized and anonymous authentication in the public key setting. Group signatures [CH91] enable devices to obtain a group signing key, which allows them to autonomously generate signatures which are publicly verifiable, without the need for certificate management. However, a completely anonymous authentication scheme such as a group signature scheme might lead to particular attacks such as Sybil attacks [Dou02], which are significant threats in applications such as vehicular ad hoc networks [GD07, ZCNC11]. Identity-based cryptosystems [Sha84a] attempt to solve the efficiency and scalability issues that emanate from having a centralized binding authority such as a CA. Identity-based signature schemes [Hes02] derive each signer's keys from their identity, removing the need for a centralized key management system. They however reveal the identity of the signer to the verifier, leading to the potential tracing of devices and subsequent privacy issues.
Direct Anonymous Attestation (DAA) (Figure 1.2) is a cryptographic primitive which enables the decentralized and anonymous authentication of trusted systems. It has been standardized by the Trusted Computing Group for the TPM [TPM19]. A DAA scheme enables embedded systems to anonymously obtain group signing keys, which they can subsequently use to authenticate while remaining anonymous to both the verifying entities and the device manager. Intel has developed the most widespread DAA scheme in practice for their SGX implementation, namely the DAA scheme known as EPID [BL11b]. We will discuss the security and efficiency limitations of EPID in Chapter 3 of this thesis. Existing DAA schemes in the literature are not efficient for low-end embedded systems, hindering their implementation in a number of constrained environments.
The second part of this thesis develops a novel DAA scheme, which satisfies the stringent security and efficiency constraints for low-end embedded systems. We will then provide two protocols which are based on our DAA scheme for (1) mobile-based authentication in public transport systems and

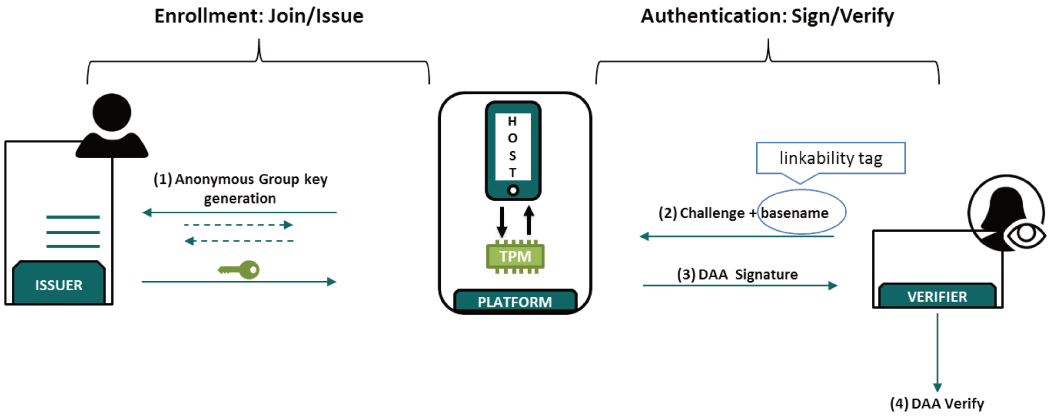


Figure 1.2: Authentication based on Direct Anonymous Attestation

(2) message authentication in Vehicle-to-Everything (V2X) communication.

1.2.4 Remote Attestation

The second part of this thesis provides attestation solutions for mobile and embedded devices, which are agnostic to the specific embedded system. In this section, we introduce the literature on remote attestation, as well as the security and efficiency challenges.

1.2.4.1 Security architecture

Remote Attestation (RA) is a security service that verifies the integrity of the software state of embedded systems. The protocol is run between a (trusted) **verifier** and a (potentially untrusted) **prover**. At the end of the protocol, RA can establish a root of trust between a verifier and a prover. Critical IoT applications implement mechanisms to monitor the internal state of devices in order to detect potential remote attacks. Devices are the provers and must provide an attestation of the software code they run. Various RA protocols have been proposed over the years, essentially falling into three categories: *hardware-based attestation, software-based attestation, and hybrid attestation*.

1.2.4.2 Swarm attestation

Over the years, research has been active in developping RA protocols for the single prover - single verifier architecture. However, in many IoT applications, groups of objects work together to efficiently harvest and communicate data and accomplish specific tasks. As such, they are referred to as device swarms. Such swarms include for example devices in smart factories, in home automation systems, and vehicles in vehicular ad-hoc networks. The naive approach consisting in combining individual RA attestation responses from each device in a large swarm results in efficiency and scalability issues. A swarm attestation protocol enables the efficient collective attestation of a group of devices. Existing collective (swarm) attestation protocols present security and privacy issues.

The third part of this thesis proposes an efficient, secure, and scalable collective attestation protocol, with enhanced security and privacy properties.

1.3 Requirements Considered in this PhD Thesis

The aim of this thesis is to provide solutions satisfying the following requirements.

1.3.1 Efficiency requirements

- **Implementation on low-end embedded systems**. Authentication and attestation solutions must be efficient enough for being implemented on low-end embedded systems.
- **Scalability**. The nature of IoT devices and networks requires scalable protocols, which account for the disruptive nature of the networks.

1.3.2 Privacy requirements

- **Privacy-preserving access control.** We consider authentication services on mobile devices, which can be used for access control in applications requiring user anonymity, untraceability, whilst ensuring accountability.
- **Concealed user identity.** In many authentication schemes, the identity of the signer is revealed. An authentication scheme should be able to authenticate a user, without retrieving or revealing any additional information.
- **Device identity and location privacy.** Adversaries may target specific control systems based on their identities or locations. An authentication scheme must enable privacy-preserving communication.
- **Revocation.** Any authentication protocol must support revocation for compromised or misbehaving devices in the network. A privacy-preserving scheme must not hinder revocation capabilities in such cases.

1.3.3 Security requirements

The design of authentication and attestation protocols satisfying the requirements specified above must provide strong security guarantees.

- **Cryptographic assumptions**. The security of each protocol and the underlying cryptographic primitive must be based on weak (in the cryptographic sense) assumptions. An example of such assumptions include *q-type* assumptions [BB04], where the parameter "q" bounds the number of solutions to the underlying mathematical problem an adversary may request while still not being able to derive a new solution to the problem.

- **Security models and proofs**. Each designed primitive and protocol must provide a rigorous security model, upon which relies the security of the overall system. Each primitive and protocol must be provably secure, based on the corresponding cryptographic assumptions.

1.4 Contributions and Outline

The aim of this thesis is to enable privacy-preserving authentication and attestation of mobile and embedded devices, with applications to a variety of Internet of Things use cases. To achieve this goal, we develop different protocols based on provably-secure cryptographic primitives, that are efficient enough for being implemented in constrained environments and particularly for low-end embedded systems. In this section, we provide an overview of the different approaches and results detailed in this thesis. Each contribution is treated in an independent chapter based on peer-reviewed publications, and can be read independently.

1.4.1 Part I. Background and Preliminaries

The first part of this thesis provides the mathematical and cryptographic preliminaries. Chapter 2 presents the primitives and protocols used in the development of our contributions, as well as the mathematical assumptions upon which we prove their security. In Chapter 3, we introduce the literature on remote attestation, as well as the literature on cryptographic tools for privacy-preserving authentication and attestation.

1.4.2 Part II. Privacy-Preserving Authentication and Access Control

The second part of this thesis provides two solutions offering privacy-preserving authentication and access control in specific constrained environments with a trusted security module. The cryptographic building block of our solutions is a novel Direct Anonymous Attestation construction, which provides strong security and efficiency guarantees.

Contribution C1: A novel Direct Anonymous Attestation scheme

Chapter 4 introduces our novel pre-Direct Anonymous Attestation scheme (pre-DAA), which provides a trade-off between security and efficiency. A pre-DAA scheme is a standalone DAA variant presented in more details in Chapter 2. Essentially, it allows the construction of a secure attestation scheme which can be undertaken by a trusted element (e.g. a SIM card), in order to strongly authenticate the device it is embedded in. DAA primitives are split between two categories based on the underlying mathematical assumptions: an interactive assumption, namely the Lysyanskaya Rivest Sahai Wolf (LRSW) assumption, versus a non-interactive *q-type* assumption, namely the sq-Strong Diffie-Hellman (q—SDH) assumption. The latter provides more robust security guarantees as we will explain in Section 4.1. We prove our new pre-DAA scheme secure under the q—SDH assumption in the random oracle model. Our new pre-DAA scheme is the more efficient construction to date, and is suitable for low-end devices, providing the required privacy-preserving trust for any authentication or attestation process the device might partake in.

Contribution C2: A privacy-preserving pseudonym scheme for V2X communications

We introduce in Chapter 5 a decentralized and privacy-preserving pseudonym scheme for Vehicle-to-Everything (V2X) communications in VANET. Our protocol enables the privacy-friendly transmission of Cooperative Awareness Messages (CAM) between vehicles. The authentication step does not require a centralized certification authority, inducing a more scalable authentication protocol for VANETs. The underlying cryptographic primitive of our solution is our pre-DAA scheme introduced in the previous chapter. By using a pre-DAA scheme to register and authenticate on the network, the vehicle's on-board unit provides a root of trust, whereby every other vehicle in the network is ensured that the vehicle sending a given message is trustworthy, and possesses valid credentials granted by a trusted entity. The protocol enables vehicles to anonymously send messages over the network, whilst enforcing user responsibility by way of the traceability property of the underlying pre-DAA scheme.

Contribution C3: A privacy-preserving access control protocol in public transport networks

In Chapter 6, we leverage the development of short-range communication technologies such as Near Field Communication (NFC) and dedicated trusted execution environments (TEEs) in smartphones to introduce a privacy-preserving mobile transit pass protocol. The challenge in using such technologies for identification and access control is the inherent privacy breaches induced by each user being uniquely traceable. Indeed, authenticated transactions by the same user are linked, which can therefore allow authorities to retrieve the user's identity. In the case of mobile transit passes, the challenge is to ensure that users can use their smartphones as transit passes, without being traced or uniquely identified. In Chapter 6, we build a privacy-friendly mobile transit pass service, which enables users to remain anonymous on the network, and to only be traceable when there is an attempt to travel without a valid and unique transit pass. Our solution leverages the anonymity and traceability properties of our pre-DAA scheme to provide the necessary security, privacy, and accountability guarantees. The efficiency of our pre-DAA scheme ensures that our solution can be implemented on a SIM card in a standalone manner, making it suitable for any mobile authentication application with stringent privacy requirements.

1.4.3 Part III. Collective Device Attestation

The third part of this thesis focuses on the concept of remote attestation itself, for the purpose of proving the integrity of computationally constrained devices. DAA schemes were introduced to provide the secure and privacy-preserving hardware assisted attestation of individual devices. The remote attestation process for IoT devices is mostly geared towards groups of devices (also known as device swarms), providing a protocol for verifying the integrity of large groups of devices in a scalable manner. In this final part of the thesis, we look closely into developing an efficient and scalable solution to verify the integrity of device swarms, with enhanced security capabilities.

Research challenge:	C1	C2	C3	C4
Security model and proof	•	•	•	•
Privacy-preserving and decentralized authentication	•	•	•	·
Privacy-preserving traceability	•	•	•	·
Authentiation scheme for the Many-to-Many device/user model	·	•	·	·
Scalable attestation	·	·	·	•

Table 1.2: Correlation between Research Challenges and the Contributions of this Thesis

Contribution C4: A swarm attestation protocol with erroneous aggregation detection

We introduce in Chapter 7 a novel solution to the challenge of providing a scalable swarm attestation protocol. Indeed, the challenge for swarm attestation protocols is to identify the compromised device in the case where the attestation response does not correspond to the expected collective state. Swarm attestation protocols often use aggregate public and private key cryptosystems in order to build a scalable attestation protocol. The issue with aggregate schemes resides in identifying individual signers from the aggregate result (this would in fact result in breaking the security of the scheme). We provide a solution to this problem by introducing CoRA, a swarm attestation protocol which enables the verifier to sequentially detect the author of an erroneous attestation report of its internal software state. Our solution leverages the algebraic properties of an aggregate algebraic Message Authentication Code (MAC), in order to build a sequential detection process.

Part I

Background and Preliminaries

CHAPTER 2

CRYPTOGRAPHIC DEFINITIONS AND PRELIMINARIES

This chapter introduces the mathematical notations and cryptographic preliminaries used in the remaining of this thesis.

Contents

2.1 Mathematical Notations

2.1.1 Groups, Rings, and Fields

Definition 2.1. (Group). A group $(\mathbb{G}, \cdot)$ is a set $\mathbb{G}$ equipped with a binary operation $\cdot : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}$ satisfying the following properties:

- **Associativity:** for all $(g_1, g_2, g_3) \in \mathbb{G}^3$, $(g_1 \cdot g_2) \cdot g_3 = g_1 \cdot (g_2 \cdot g_3)$;
- **Identity:** there exist an element $e \in \mathbb{G}$, such that for all element $g \in \mathbb{G}$, $e \cdot g = g \cdot e = g$. The element e, denoted $1_{\mathbb{G}}$ in the multiplicative notation, is referred to as the *identity* element;
- **Inverse element:** for all $g \in \mathbb{G}$, there exist an element $g' \in \mathbb{G}$, denoted g^{-1} in the multiplicative notation, such that $g \cdot g' = g' \cdot g = e$, where e is the identity element.

Definition 2.2. (Subgroup). Let $(\mathbb{G}, \cdot)$ be a group and $\mathbb{H}$ a subset of $\mathbb{G}$. $(\mathbb{H}, \cdot)$ is a subgroup of $(\mathbb{G}, \cdot)$ if it satisfies the following requirements:

- $1_{\mathbb{G}} \in \mathbb{H}$;
- $\forall (x, y) \in \mathbb{H}^2, x \cdot y \in \mathbb{H}$;
- $\forall x \in \mathbb{H}, x^{-1} \in \mathbb{H}$.

Any subgroup of a group is itself a group.

In the remaining, we will refer to the group $(\mathbb{G}, \cdot)$ as $\mathbb{G}$

Definition 2.3. (Commutative Group). A commutative group, also referred to as abelian group, is a group $\mathbb{G}$ with the following additional property on the binary operation:

- **Commutativity:** for all elements $(g_1, g_2) \in G^2$, $g_1 \cdot g_2 = g_2 \cdot g_1$.

Definition 2.4. .

- **Subgroup generated by an element:** Let x be an element of the group $\mathbb{G}$. The set $\{x^n, n \in Z\}$, denoted by $\langle x \rangle$, is called the *subgroup* generated by x;
- **Cyclic group:** A group $\mathbb{G}$ is *cyclic* if $\exists x \in G$ such that $\mathbb{G} = \langle x \rangle$. x is called the *generator* of group $\mathbb{G}$;
- **Finite group:** A group $(\mathbb{G}, \cdot)$ is *finite* if the set $\mathbb{G}$ is finite;
- **Order of a group:** The number of elements of a finite group $\mathbb{G}$, denoted by $|\mathbb{G}|$, is called the *order of the group*;
- **Order of an element:** The *order of an element* $x \in \mathbb{G}$, denoted by $|x|$, is the order of the finite subgroup $\langle x \rangle$ generated by x, i.e. the least positive integer n such that $x^n = 1$ (if it exists).

Any group $\mathbb{G}$ of prime order p is cyclic, and any element $x \in \mathbb{G} \backslash \{1_{\mathbb{G}}\}$ is a generator of $\mathbb{G}$.

Definition 2.5. (Ring). A ring $(\mathbb{R}, +, \cdot)$ is a set $\mathbb{R}$ equipped with two binary operations $+ : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ and $\cdot : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ satisfying the following properties:

- $(\mathbb{R}, +)$ is a commutative group, and $0_{\mathbb{R}}$ is the identity element for the binary operation "+";
- the multiplicative operation $\cdot$ is associative, and $1_{\mathbb{G}}$ is the identity element for the binary operation "$\cdot$" ;
- the multiplicative operation $\cdot$ is distributive with respect to addition, i.e. for all $(g_1, g_2, g_3) \in G^3$, $g_1 \cdot (g_2 + g_3) = (g_1 \cdot g_2) + (g_1 \cdot g_3)$.

$\mathbb{Z}$ denotes the set of integers, and $\mathbb{N}$ the set of positive integers. Let $n \in \mathbb{N}$, the ring $\mathbb{Z}_n = \mathbb{Z}/\mathbb{Z}_n$ denotes the ring of integers modulo n.

Definition 2.6. (Field). A field $(\mathbb{F}, +, \cdot)$ is a commutative ring with the following additional property:

- For every element $g \in (\mathbb{F}^*, +, \cdot)$, there exists a unique multiplicative inverse g^{-1} such that $g \cdot g^{-1} = 1_{\mathbb{F}}$;

A field $(\mathbb{F}, +, \cdot)$ is finite if $\mathbb{F}$ is a finite set. The number of elements of $\mathbb{F}$ denoted by $|\mathbb{F}|$, is called the *order* of the field. The *characteristic* of a field, when it exists, is the smallest positive integer n such that $\underbrace{1 + 1 + ... + 1}_{n \text{ times}} = 0$. If there is no such integer, the field is said to have a characteristic equal to 0.

- The order of $(\mathbb{F}^*, \cdot)$ is power of a prime, i.e. in the form p^n where p is the characteristic and n an integer such that $n \geq 1$. For each prime p and any positive integer $n \geq 1$, there exists a unique finite field (up to an isomorphism) $(\mathbb{F}, +, \cdot)$ such that $|\mathbb{F}| = p^n$. Such a field is denoted by $\mathbb{F}_{p^n}$. If p is prime, $\mathbb{F}_p = \mathbb{Z}_p$, however whenever $n > 1$, $\mathbb{F}_{p^n} \neq \mathbb{Z}_{p^n}$ as $\mathbb{Z}_{p^n}$ is not a field.

For a finite field $\mathbb{F}$, the set $\mathbb{F}^* = \mathbb{F} \backslash \{0_K\}$ equipped with the multiplicative operation $\cdot$ is a cyclic group of order $p - 1$.

The security of the cryptographic algorithms developed in this thesis is based on hardness assumptions, the majority being based on variants of the Discrete Logarithm Problem in finite fields of order p, where p is a prime integer. We consider in such cases a subgroup of prime order q of $\mathbb{F}_p^*$, where q divides $p - 1$. The size of the subgroup of order q must be large enough to be resistant to attacks attempting to solve the discrete logarithm problem, notably the "baby-step giant-step" algorithm [Sha71]. In practice, cryptographic algorithms today make use of bilinear groups, formed by points on an elliptic curve.

2.1.2 Elliptic curve groups

The use of elliptic curves in cryptography was introduced independently by Koblitz [Kob87] and Miller [Mil85]. Elliptic curve cryptography (ECC) is used in the construction of practical cryptosystems, due to the relatively small parameters and key sizes of said cryptosystems compared to other non-ECC cryptosystems for equivalent security [CMRR19]. It is therefore especially

practical for implementing cryptographic algorithms in constrained environments. ECC leverages the mathematical properties of elliptic curves, notably an algebraic operation in elliptic curve groups called bilinear map or *pairing*.

Definition 2.7. (Elliptic curve over a finite field). An elliptic curve E over a finite field F_{p^n}, denoted by $E(F_{p^n})$, consists of a *point at infinity* $\mathcal{O}$ along with a set of points (x, y) verifying the following *Weierstrass* equation:

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \text{ where } a_1, a_2, a_3, a_4, a_6 \in \mathbb{F}_{p^n}$$

If the characteristic $p \neq 2, 3$, the *Weierstrass* equation can be simplified as follows:

$$E : y^2 = x^3 + ax + b \text{ where } (a, b) \in \mathbb{F}_{p^n}^2 \text{ and } 4a^3 + 27b^2 \neq 0$$

The condition $4a^3 + 27b^2 \neq 0$ ensures that the curve is *non-singular*. In particular, it ensures that we can compute the tangent at every point except $\mathcal{O}$. Otherwise the curve is said to be *singular*.

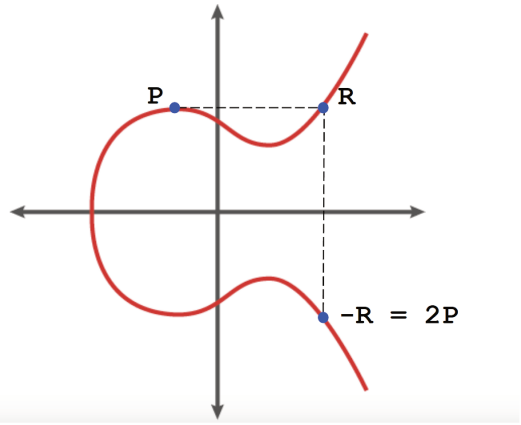


Figure 2.1: Doubling of the point P on an elliptic curve.

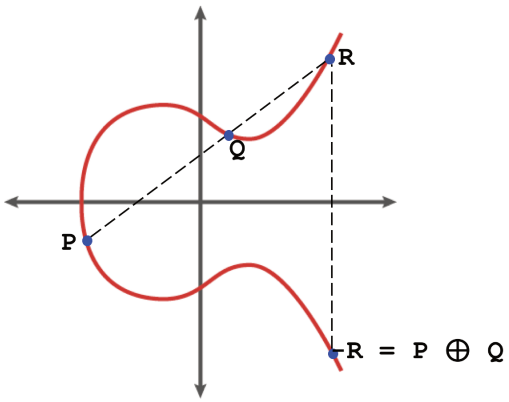


Figure 2.2: Addition of two distinct points P and Q on an elliptic curve.

Points on an elliptic curve equipped with the addition operation denoted by +, form an algebraic group with the point at infinity $\mathcal{O}$ as the identity element. The algebraic operation is defined as follows:

Definition 2.8. (Additive law). Let P and Q be two different points on the elliptic curve $E(F_{p^n})$:

- The inverse of $P = (x_P, y_P) \neq \mathcal{O}$ is denoted by $-P$ whose coordinates are $(x_P, -y_P) \in E(\mathbb{F}_{p^n})$. The inverse of $-\mathcal{O}$ is equal to $\mathcal{O}$, and $\mathcal{O} + P = P$;

- As depicted in Figure 2.1, if $P \neq -P$, the tangent line at point P intersects the curve E at a second point $R = (x_R, y_R) \in E(\mathbb{F}_{p^n}) \backslash \mathcal{O}$. $-R$ is equal to the sum $P + P = [2]P$;

- If $P \neq -Q$, the line through P and Q intersects the curve E at a third point $R = (x_R, y_R) \in E(\mathbb{F}_{p^n}) \backslash \mathcal{O}$. As depicted in Figure 2.2, $-R$ is equal to the sum of P and Q.

$E(\mathbb{F}_{p^n}, +)$ is an abelian group, and the additive law can be described algebraically as follows: the sum of points $P = (x_P, y_P) \neq \mathcal{O}$ and $Q = (x_Q, y_Q) \neq \mathcal{O}$ is the point $R = (x_R, y_R)$ with:

$$x_R = \lambda^2 - x_P - x_Q \text{ and } y_R = \lambda(x_P - x_R) - y_P$$

where λ is defined as

- $\lambda = \frac{y_Q - y_P}{x_Q - x_P}$ if $P \neq \pm Q$ and $P, Q \neq \mathcal{O}$

- $\lambda = \frac{3x_P^2 + a}{2y_P}$ if $P = Q$ and $P \neq \mathcal{O}$

Definition 2.9. (Bilinear groups) Let $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ be three cyclic groups of the same prime order p, with respective generators g_1, g_2 and g_T. The tuple $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ is called a bilinear group if the map $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$, also called *pairing*, satisfies the following properties:

- For all $(x, y) \in \mathbb{G}_1 \times \mathbb{G}_2$ and for all $(a, b) \in \mathbb{Z}_p^2$, $e(x^a, y^b) = e(x, y)^{ab}$ (**bilinearity**);

- For all $(x, y) \in \mathbb{G}_1 \times \mathbb{G}_2$, if $e(x, y) = 1$ then $x = 1_{\mathbb{G}_1}$ or $y = 1_{\mathbb{G}_2}$ (**non-degeneracy**);

- For all $(x, y) \in \mathbb{G}_1 \times \mathbb{G}_2$, there exists an efficient algorithm which computes $e(x, y)$ (**efficiency**).

A pairing is called *symmetric* if $\mathbb{G}_1 = \mathbb{G}_2$. Otherwise it is called *asymmetric*.

In practice, $\mathbb{G}_1$ is the group formed by points on an elliptic curve E equipped with a group operation, denoted by $E(\mathbb{F}_p)$. $\mathbb{G}_2$ is a subgroup of a related elliptic curve group $E(\mathbb{F}_p^k)$, whilst $\mathbb{G}_T$ is a subgroup of the finite field $\mathbb{F}_p^k$. Since the successful construction of a one-round 3-party Diffie-Hellman key exchange protocol by Joux [Jou00] using pairings, they have been used in the design of a number of relatively efficient cryptographic algorithm. Most notably, they were used in the design of Identity-based [Hes02, BGLS03], and short [BLS01] signature schemes. Different elliptic curves yield different security and efficiency guarantees for the pairing. In particular, in 2008, Galbraith, Paterson, and Smart[GPS08] classified pairings in three different types described as follows:

- **Type 1 (symmetric):** there exists two efficiently computable isomorphisms $\phi_1 : \mathbb{G}_1 \rightarrow \mathbb{G}_2$ and $\phi_2 : \mathbb{G}_2 \rightarrow \mathbb{G}_1$;

- **Type 2 (asymmetric):** there exists an efficiently computable isomorphism $\phi : \mathbb{G}_2 \to \mathbb{G}_1$, but no efficiently computable isomorphism from $\mathbb{G}_1$ to $\mathbb{G}_2$;
- **Type 3 (asymmetric):** there exists no efficiently computable isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$.

Given the absence of isomorphism between type 3 pairings, they yield more secure cryptosystems. They also yield more efficient constructions, and have thus been favored in the construction of cryptographic algorithms in recent years. The algorithms developed in the remaining of this thesis make use of type 3 pairings.

2.2 Cryptographic Preliminaries

In this section, we briefly recall basic complexity notions, and present some basic functions used in cryptography.

2.2.1 Complexity Tools

2.2.1.1 Security parameter and adversary's running time

In order to formalize security notions in cryptography, the attacking powers of the adversary must be bounded. Indeed, an adversary with an unlimited amount of computing power and time is able to break any cryptosystem. Proving the security of cryptographic systems must therefore be done with respect to a *computationally bounded* adversary. We denote by t the upper-bound on the adversary's running time. The *security parameter* on the other end, denoted by λ, is a positive integer used to express the input size of the underlying computational problem. Informally, it is used to measure how "hard" it is for the adversary to break a cryptographic scheme based on said computational problem. The security parameter is often expressed in its unary representation 1^λ.

2.2.1.2 Turing machines

The *Turing machine*, introduced in 1936 by Alan Turing, is an abstract stateful machine that can simulate any computer algorithm regardless of its complexity. The machine processes a set of instructions by writing on a *tape*, and moving "left" or "right" on said tape according to the instructions and a finite set of *states*. Formally, a Turing machine is defined as follows:

Definition 2.10. (Turing machine). A Turing machine is a 7-tuple $T = (\Sigma, Q, \sigma, \delta, \Delta, q_0, \mathscr{F})$ where:

- Σ is a finite, non-empty set of alphabet symbols;
- Q is a finite and non-empty set of states;
- $\sigma : Q \times \Sigma \to \Sigma$ is the writing function;
- $\delta : Q \times \Sigma \to Q$ is the state changing function;

- $\Delta : Q \times \Sigma \rightarrow \{L, R\}$ is the transition function where L corresponds to a left shift, and R to a right shift;

- $q_0 \in Q$ is the initial state, i.e. the state of the *tape head* at the beginning of the computation;

- $\mathscr{F} \subset Q$ is the set of final states

In the remaining of this thesis, we focus on algorithms that have time complexity $TM(x)$, i.e. algorithms whose time complexity corresponds to the number of Turing machine steps taken from the initial to the final state on input x.

2.2.1.3 Complexity definitions

The efficiency of cryptographic algorithms is defined with respect to the complexity notions defined as follows:
An algorithm $\mathscr{A}$ is said to have *polynomial time complexity* if there exists a polynomial $p(.)$ such that for all $x \in \{0,1\}^*$, $\mathscr{A}$'s running time on input x is bounded by $p(|x|)$, where $|x|$ denotes the size of x. More formally, polynomial time complexity is defined as follows:

Definition 2.11. (Polynomial time complexity) Let $T_M(n) = sup\{T_{M(x)}, |x| = n\}$ where M is a Turing machine. The time complexity of M is said to be polynomial if

$$\exists n_0 \in \mathbb{N}, \exists c \in \mathbb{N}^*, \text{ such that } \forall n \geq n_0, T_M(n) \leq n^c$$

An algorithm with polynomial time complexity is considered to be *efficient* and a problem is said to be *hard* if there is no polynomial time algorithm that can solve it.
A polynomial time algorithm is said to be *probabilistic* if it "flips" a polynomial number of random coins (or bits), and uses the result of these coin tosses to determine the next state in his computation. Conversely, an algorithm is said to be *deterministic* if, given the same input (i.e. without any randomness), it will always produce the same output.

Definition 2.12. (Negligible function). A function $\epsilon : \mathbb{N} \rightarrow \mathbb{R}$ is said to be *negligible* if:

$$\forall c > 0, \exists k_c > 0 \text{ such that } \forall k \geq k_c, |\epsilon(k)| < \frac{1}{k^c}$$

Definition 2.13. (Negligible probability). Let P be a probability that depends on the security parameter λ. P is said to be negligible if it is a negligible function of λ.

Definition 2.14. (Overwhelming probability). A probability P is said to be overwhelming if $1 - P$ is negligible.

2.2.2 Basic Functions

A number of basic functions are used in the definition of cryptographic schemes. We define here some basic functions used in the remaining of this thesis.

2.2.2.1 One-Way Function (OWF)

A one-way function $f : \{0,1\}^* \to \{0,1\}^*$ is a function that is easy to compute but hard to invert. More formally, a one-way function is defined as follows:

Definition 2.15. (One-Way Function). Let $f : \{0,1\}^* \to \{0,1\}^*$ be a function. f is said to be *one-way* if it verifies the following properties:

- **Efficient evaluation:** there exists a polynomial time algorithm that can efficiently compute $f(x)$ on input $x \in \{0,1\}^*$;

- **One-wayness:** for all polynomial time algorithms and all $y = f(x)$, the probability of finding $x' \in \{0,1\}^\lambda$ such that $y = f(x')$ is negligible in λ. More formally, for any efficient algorithm $\mathscr{A}$ (the adversary), for all sufficiently large λ, the following probability is negligible:

$$\mathsf{Pr}[x \leftarrow \{0,1\}^\lambda; y \leftarrow f(x); x' \leftarrow \mathscr{A}(1^\lambda, y) : f(x') = y]$$

2.2.2.2 Hash Function

Hash functions are one-way functions with additional properties. They take bit strings of arbitrary length and output bitstrings of fixed length λ where λ is the security parameter.

Definition 2.16. (Cryptographic hash function) A hash function $H : \{0,1\}^* \to \{0,1\}^\lambda$ is said to be *cryptographically secure* if it verifies the following properties:

- **Pre-image resistance (one-way):** given an element $y \in \{0,1\}^\lambda$, the probability of finding x such that $H(x) = y$ is negligible, i.e. the following probability is negligible;

$$\mathsf{Pr}[y \leftarrow \{0,1\}^\lambda; x \leftarrow \mathscr{A}(1^\lambda, y) : y = H(x)]$$

- **Second pre-image resistance:** given $x_1 \in \{0,1\}^*$, the probability of finding $x_2 \neq x_1$ such that $H(x_1) = H(x_2)$ is negligible, i.e. the following probability is negligible;

$$\mathsf{Pr}[x \leftarrow \{0,1\}^*; x' \leftarrow \mathscr{A}(1^\lambda, x) : H(x') = H(x) \text{ and } x \neq x']$$

- **Collision resistance:** the probability of finding $(x_1, x_2) \in (\{0,1\}^*)^2$ with $x_2 \neq x_1$ such that $H(x_1) = H(x_2)$ is negligible, i.e. the following probability is negligible;

$$\mathsf{Pr}[(x, x') \leftarrow \mathscr{A}(1^\lambda); H(x') = H(x)]$$

In this thesis, we only consider cryptographically secure hash functions.

2.2.2.3 Pseudo-Random Function (PRF)

A pseudo-random function is a function that is computationally indistinguishable from a random function. Let $F : \mathcal{K} \times \mathcal{D} \rightarrow \mathcal{R}$ be a computational function where $\mathcal{K}$ is the key space of size $|\mathcal{K}| = \lambda_{\mathcal{K}}$, $\mathcal{D}$ the domain space of size $|\mathcal{D}| = \lambda_{\mathcal{D}}$, and $\mathcal{R}$ the range of size $|\mathcal{R}| = \lambda_{\mathcal{R}}$. For $K \in \mathcal{K}$, we denote F_K the function that is the partial evaluation of F on K, i.e.:

$$F_K : \mathcal{D} \rightarrow \mathcal{R}$$
$$x \rightarrow F(K, x)$$

F_K is a pseudo-random function if it verifies the following properties:

- **Efficient evaluation:** given K and x, there exists a polynomial time algorithm that can efficiently compute $F_K(x)$;
- **Pseudo-randomness:** for all polynomial time algorithms $\mathcal{A}$ bounded by the number of oracle queries to F_K, the advantage of $\mathcal{A}$ in distinguishing F_K from a random function is negligible.

2.2.3 Hardness Assumptions

In order to prove the security of cryptographic schemes, we make use of a set of *computationally hard assumptions*, defined in cyclic groups of prime order. A problem is considered to be *hard* if its resolution by a polynomial time algorithm (also known as the adversary) is computationally unfeasible. In other words, an adversary's probability of solving the problem (also known as his advantage) is negligible for a given security parameter. In this thesis, we make use of three categories of hardness assumptions.

2.2.3.1 Assumptions based on the Discrete Logarithm problem

The security of the cryptographic algorithms developed in this thesis is based on variants of the Discrete Logarithm (DL) problem. In the remaining of this thesis, we consider groups where such problems are considered to be *hard*, such as finite fields or elliptic curve groups.

Definition 2.17. (Discrete Logarithm Assumption (DL)) Let $\mathbb{G}$ be a cyclic group of prime order p. The *Discrete Logarithm* (DL) assumption states that given a random generator $g \in \mathbb{G}$ and an element $h \in \mathbb{G}$, a polynomially-bounded adversary $\mathcal{A}$ is able to compute $x \in \mathbb{Z}_p$ such that $h = g^x$ only with negligible probability. The integer x, written $x = \log_g(h)$, is called the discrete logarithm of h in base g.

A variant of this assumption consists in evaluating the probability of finding the discrete logarithm of an element h while having had previous access to a set of elements $(h_1, ..., h_t)$ and their discrete logarithms $(x_1, ..., x_t)$ in base g. The assumption is referred to as the One-More Discrete Logarithm (OMDL) assumption, and was formalized by Bellare, Namprepre, Pointcheval, and Semanko [BNPS03].

Definition 2.18. (One-More Discrete Logarithm Assumption (OMDL)) Let $\mathbb{G}$ be a cyclic group of prime order p. Given a random generator g, a challenge oracle $\mathcal{O}_1$ that returns a random element $h_i \in \mathbb{G}$ when queried, and a discrete logarithm oracle $\mathcal{O}_2$, that returns the discrete logarithm x_i of h_i when queried. The *One-More Discrete Logarithm* (OMDL) assumption states that after t queries to $\mathcal{O}_1$ (where t is chosen by the adversary), and at most $t-1$ queries to $\mathcal{O}_2$, the adversary $\mathcal{A}$ has negligible probability in recovering the discrete logarithms of all t elements h_i for $i \in \{1,...,t\}$.

Diffie and Hellman introduced the first key exchange protocol based on a hardness assumption derived from the DL assumption, namely the Diffie-Hellman (DH) assumption [DH76]. The two variants of the Diffie-Hellman assumption are presented below.

Definition 2.19. (Computational Diffie-Hellman Assumption (CDH)) Let $\mathbb{G}$ be a cyclic group of prime order p. The *Computational Diffie-Hellman* (CDH) assumption states that given a random generator $g \in \mathbb{G}$ and two elements $(g^a, g^b) \in \mathbb{G}^2$, where $(a,b) \in Z_p^2$, an adversary $\mathcal{A}$ is able to compute g^{ab} only with negligible probability.

In 1998, Boneh formalized the *decisional* version of the CDH assumption, called the Decisional Diffie-Hellman Assumption (DDH) [Bon98].

Definition 2.20. (Decisional Diffie-Hellman Assumption (DDH)) Let $\mathbb{G}$ be a cyclic group of prime order p. The *Decisional Diffie-Hellman* (DDH) assumption states that given a random generator $g \in \mathbb{G}$, two elements $(A = g^a, B = g^b) \in \mathbb{G}^2$ where $(a,b) \in Z_p^2$, and a random element $X \in \mathbb{G}$, an adversary $\mathcal{A}$ is able to decide whether $X = g^{ab}$ only with negligible probability.

The triplet (g^a, g^b, g^{ab}) is called a DDH triplet. The DDH assumption can also be defined as follows:

Definition 2.21. (Decisional Diffie-Hellman Assumption' (DDH')) Let $\mathbb{G}$ be a cyclic group of prime order p. The *Decisional Diffie-Hellman* (DDH) assumption states that given two random generator $g, h \in \mathbb{G}$, and two elements $(g^a, h^b) \in \mathbb{G}^2$ where $(a,b) \in Z_p^2$, an adversary $\mathcal{A}$ is able to decide whether $a = b$ only with negligible probability.

2.2.3.2 Assumptions based on the Discrete Logarithm problem in bilinear groups

Discrete logarithm-based assumptions have variants that hold in bilinear groups. Notably, it is easy to show that the DDH assumption is easy in bilinear groups while the CDH problem remains intractable. Indeed, for a type 1 bilinear group $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$, an adversary $\mathcal{A}$ is able to verify the equality $e(A,B) = e(X,g)$, and win the DDH game with probability 1, while CDH remains hard. In type 3 bilinear groups, the DL and the CDH problems are intractable in both $\mathbb{G}_1$ and $\mathbb{G}_2$, while DDH is only intractable in $\mathbb{G}_1$. This assumption, called the eXternal Diffie-Hellman (XDH) assumption, was formalized by Boneh, Boyen, and Shacham [BBS04].

Definition 2.22 (XDH (eXternal Diffie-Hellman) Assumption)**.** Let $\mathbb{G}_1, \mathbb{G}_2$ and $\mathbb{G}_T$ be three cyclic groups of prime order p, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ a type 3 bilinear map. The *eXternal Diffie-Hellman* (XDH) assumption states that the DDH assumption holds in $\mathbb{G}_1$.

Boneh and Boyen [BB04] introduced an additional assumption for type 3 bilinear groups.

Definition 2.23 (q—SDH assumption)**.** Let $\mathbb{G}_1$, $\mathbb{G}_2$, $\mathbb{G}_T$ be three type 3 bilinear groups of prime order p. Let g_1 (respectively g_2) be a generator of $\mathbb{G}_1$ (respectively $\mathbb{G}_2$). The q—SDH assumption states that given a $(q+2)-$tuple $(g_1, g_2, g_2^x, g_2^{(x^2)}, ..., g_2^{(x^q)})$, no adversary can efficiently output a pair $(c, g_1^{\frac{1}{x+c}}) \in \mathbb{Z}_p^* \times \mathbb{G}_1$.

2.2.3.3 Miscellaneous assumptions

In 1999 Lysyanskaya, Rivest, Sahai, and Wolf [LRSW99] introduced a new hardness assumption.

Definition 2.24. (LRSW assumption) Let $\mathbb{G}$ be a cyclic group and g a generator. Given a tuple $(A = g^x, B = g^y)$, an oracle $\mathcal{O}_{LRSW}$ can be queried on (A, B), which returns a triple (a, a^y, a^{x+mxy}) for a random group element $a \in \mathbb{G}$. The LRSW problem consists in generating a triplet $(b, b^y, b^{x+m'xy})$ for a message m' that has not been previously submitted to the oracle. The LRSW assumption states that a polynomial time adversary can output such a triplet only with negligible probability.

A variant of the LRSW assumption for type 3 bilinear groups was introduced by Pointcheval and Sanders [PS16] in 2016. The assumption was proved in the generic group model presented in Section 2.3.2.3.

Definition 2.25. (Pointcheval-Sanders Assumption1) Let $\mathbb{G}$ be a cyclic group and g a generator. Given a tuple $(A = g^x, B = g^y)$, an oracle $\mathcal{O}_{LRSW}$ can be queried on (A, B), which returns a triple (a, a^y, a^{x+mxy}) for a random group element $a \in \mathbb{G}$. The LRSW problem consists in generating a triplet $(b, b^y, b^{x+m'xy})$ for a message m' that has not been previously submitted to the oracle. The LRSW assumption states that a polynomial time adversary can output such a triplet only with negligible probability.

In 2018, Pointcheval and Sanders [PS18] introduced a variant of the q—SDH assumption for type 3 bilinear groups, namely the q—MSDH assumption, proven secure in the generic group model.

Definition 2.26. (q—MSDH assumption) Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ be a bilinear group of type 3, with g_1 (respectively g_2) a generator of $\mathbb{G}_1$ (respectively $\mathbb{G}_2$). Given $\{g_1^{x^i}, g_2^{x^i}\}_{i=0}^q$ and $(g_1^a, g_2^a, g_2^{a \cdot x})$, for random $a, x \in \mathbb{Z}_p^*$, an adversary $\mathcal{A}$ can only output with negligible probability a tuple $(w, P, h^{\frac{1}{x+w}}, h^{\frac{a}{P(x)}})$ for some $h \in \mathbb{G}_1^*$, where P is a polynomial of degree at most q and w is a scalar such that $(X + w)$ and $P(X)$ are relatively prime.

Finally, the security of the Paillier cryptosystem described in Section 2.5.1.3 is based on the Decisional Composite Residuosity (DCR) Assumption [Pai99].

Definition 2.27. (Decisional Composite Residuosity Assumption (DCR)) There exists no probabilistic polynomial time distinguisher for n-th residues modulo n^2. In other words, there is no probabilistic polynomial time adversary that can distinguish the set $\mathbb{CR}$ from $\mathbb{Z}_{n^2}^*$, where $\mathbb{CR} = \{z \in \mathbb{Z}_{n^2}^*, \exists y \in \mathbb{Z}_{n^2}^* : z = y^n \bmod n^2\}$.

2.3 Provable security

The security of a cryptographic protocol must be proven with respect to a robust security model. The security of a cryptographic primitive is based on how hard it is for a class of attackers to break the primitive.

2.3.1 Methods to generate security proofs

In cryptography, the security of a cryptographic scheme is determined with respect to given a security notion. The security proof should show that if an adversary is able to break the security of the scheme, a second adversary (called challenger) is able to solve the underlying *hard* problem. Starting from the assumption that the problem is intractable, the adversary's probability of breaking the scheme with respect to a given security property is therefore negligible. In this thesis, we therefore prove the security of our schemes by attempting to prove that the adversary's ability to break the scheme reduces to solving the underlying problem, which is considered to be hard. This method called *reduction*, was introduced by Goldwasser and Micali [GM84]. There are two methods to generate security proofs, namely *game-based* and *simulation-based*.

2.3.1.1 Game-based security

In the game-based approach, a security experiment $\text{Exp}^{P}_{\mathscr{A}}$ formalizes a security property P between an adversary $\mathscr{A}$ and a challenger $\mathscr{C}$. In the experiment, $\mathscr{A}$ has access to a set of query oracles that model his attacking power and his real life capabilities. The adversary's goal is to solve the game with non-negligible probability. His *advantage* in winning the game is defined as the difference between his probability of winning the game and the probability of winning the game through random guessing. We say that the cryptographic primitive satisfies the initial security property P if the adversary's advantage is negligible. Shoup [Sho04] formalized the game-based approach by introducing what is now known as the Shoup *game-hopping technique*. The method consists in modeling an attack for a given security property as an initial attack game (Game 0) between the adversary and the challenger. Then defining a sequence of subsequent games (Game 1, Game 2,..., Game i), where adversary $\mathscr{A}$ cannot detect a change between two consecutive games. For each game (Game i), S_i denotes the event that the adversary wins Game i. The probability $\Pr[S_i]$ must therefore be negligible, and the probability $\Pr[S_{i+1}]$ must be negligibly close to $\Pr[S_i]$. The last game (Game n) is built so that $\Pr[S_n]$ is negligibly close to the target probability (i.e. the probability of a random guess). The probability of the initial attack game $\Pr[S_0]$ being therefore negligibly close to $\Pr[S_n]$, we are able to prove the security of the scheme.
This method is useful to prove the security of complex cryptographic primitives, and is the method used to prove the security of cryptographic primitives in this thesis.

2.3.1.2 Simulation-based security

In the simulation-based approach, the security of a cryptographic primitive relies on the *real world*/*ideal world* paradigm. The adversary mounts an attack against security property of the

scheme in the real world. In parallel, this approach considers an ideal world where an *ideal functionality*, denoted by $\mathscr{F}$, is defined such that all interactions between different parties (oracles, challenger) are made through $\mathscr{F}$. In this approach, a protocol is considered secure if the probability of distinguishing between its execution in the real world, denoted $\text{view}_{\text{Real}}$ and its execution in the ideal world, denoted $\text{view}_{\text{Ideal}}$, is negligible. This means that the interactions between different parties are computationally indistinguishable in both worlds. The ideal world being modeled as a setting where no attack can successfully be mounted, this proves the security of the scheme. To prove the security of the scheme, one must build a *simulator* $\mathscr{S}$ that can emulate the adversary's interactions in the ideal world. A cryptographic protocol's security is then evaluated by its ability to emulate the ideal process.
This approach presents advantages compared to the game-based approach, as it allows to prove the security of protocols as stand-alone protocols, as well as part of larger complex protocols. This notion is also known as *universal composability (UC)* [Can01]. In the UC model, the challenge of formulating a security model capturing all the threats against the cryptosystem, as well as the possible execution scenarios are solved by introducing an ideal process running the protocol in a secure way. In the ideal world, all parties in the protocol execution hand their inputs to a trusted party (the ideal functionality) who runs the protocol as expected, and returns each party's output.

2.3.2 Security Models

The attacks of an adversary $\mathscr{A}$ must be defined with respect to a specific model defining the attack environment. We describe here the three major security models.

2.3.2.1 Random Oracle Model (ROM)

In 1993, Bellare and Rogaway [BR93] formalized a model which exploits the properties of random oracles [FFS88] in order to prove the security of efficient schemes. In the random oracle model, hash functions (defined in Section 2.2.2) are modeled as ideal functions (or random oracles), which when queried on an input M, return an element T indistinguishable from a random element. A hash function modeled as a random oracle thus acts as a perfectly random function. In practice, a hash function is not a completely random function, which has drawn criticism from the cryptographic community [CGH04]. However, the attacks designed against schemes in the ROM have been especially designed to perform in a specific way which is unrealistic when compared to real world schemes. Moreover, schemes proven in the random oracle modeled are generally very efficient, and are notably suitable for implementation in constrained environments.

2.3.2.2 Standard Model

In the standard model, the security of a scheme is proven solely based on the hardness of the underlying problem. The environment does not consider any idealized model for the groups or functions to which an adversary $\mathscr{A}$ has access. Proofs generated in this model are often complex and schemes proven secure in the standard model are often inefficient.

2.3.2.3 Generic Group Model (GGM)

The generic group model [Mau05, JS08] is an ideal model where the adversary $\mathscr{A}$ cannot exploit a specific group structure in order to break the scheme. In order to perform a specific group operation, he has to query an oracle which returns the expected answer.

2.4 Symmetric-Key Cryptographic Primitives

In cryptography, the primary goal is to ensure the *confidentiality* of messages. *Encryption* is the process of transforming an original input message called plaintext, into an encoded output called ciphertext. The process of transforming a plaintext into an encoded version is called *Encryption*, and the reverse process of retrieving a message from a ciphertext is called *Decryption*. Both processes involve a secret encoding parameter called the (encryption or decryption) *key*. The transformations and substitutions operated on the plaintext during encryption are dependent on the key. Introduced by Kerckhoff in the 19^{th} century [Ker83], and later formalized by Shannon [Sha49], the principle known as "Kerckhoff's Principle" states that a cryptosystem should be secure even if everything about the system, except the key, is public knowledge. Cryptographic systems are divided into two categories, namely *symmetric* and *asymmetric*. Symmetric key cryptosystems use the same secret key for both encryption and decryption. Such schemes therefore require a preliminary key exchange protocol, at the end of which they both obtain the secret key used in the encryption process. The key must be kept secret, and the key exchange protocol must ensure that no adversary is able to obtain the secret key during the exchange process.

2.4.1 Symmetric key encryption

A symmetric key encryption scheme involves an encryption algorithm Enc, which takes as input a secret key k and a plaintext m, and outputs the ciphertext c defined as follows: $c = \mathsf{Enc}(k, m)$. The decryption algorithm Dec takes as input the secret key k and a ciphertext c generated using Enc, and outputs the original plaintext $m = \mathsf{Dec}(k, c)$. The correctness of the encryption scheme ensures that the following equation holds:

$$m = \mathsf{Dec}(k, \mathsf{Enc}(k, m))$$

2.4.2 Message Authentication Code

A Message Authentication Code (MAC) is a symmetric key primitive that provide message authentication and integrity. A MAC scheme consists in appending a tag (also referred to as the MAC) to the message using a secret key sk shared between the sender and the receiver.

2.4.2.1 Definition and security notions

Definition 2.28. (Message Authentication Code). A MAC scheme consists of the following algorithms:

- $\mathsf{Setup}(1^\lambda)$: a probabilistic algorithm that takes as input the security parameter 1^λ, and outputs the public parameters pp;

- $\mathsf{Keygen}(pp)$: a probabilistic algorithm that takes as input pp, and outputs a secret key sk;

- $\mathsf{Mac}(pp, sk, m)$: an algorithm that takes as input the public parameters pp, the secret key sk, and a message m, and outputs the tag τ;

- $\mathsf{Verify}(pp, sk, m, \tau)$: a deterministic algorithm that takes as input the public parameters pp, the secret key sk, the message m, and the tag τ, and outputs 1 (accept) or 0 (reject).

A MAC scheme must satisfy the following properties:

- *Authenticity*: A valid tag can only be generated or verified by a party holding the secret key sk;

- *Integrity*: Any modification made to the original message m upon generating the tag renders the tag invalid with overwhelming probability;

- *Validity*: If the tag τ is valid with respect to both m and sk, then $\mathsf{Verify}(pp, sk, m, \tau)$ returns 1 with overwhelming probability.

MAC schemes are usually constructed from hash functions (HMAC) or block ciphers (CBC-MAC). The security of MAC scheme varies according to its type, namely whether it is a *deterministic* or a *probabilistic* MAC.

Deterministic MAC. A MAC scheme is said to be *deterministic* if for a given message m, there exists a unique valid tag τ. The security notion for a deterministic MAC is that of *unforgeability under chosen message attack* (UF-CMA). This notion encapsulates the fact that an adversary $\mathscr{A}$, which has access to a Mac oracle, can output a valid tag τ on a message m that has not been queried only with negligible probability. The Mac oracle is a query/response oracle where any entity can submit a message m, and the oracle sends back the corresponding tag τ. The UF-CMA security experiment is depicted in Figure 2.3.

$\mathrm{Exp}^{\text{UF-CMA}}_{\mathscr{A}}(1^\lambda)$:

1. $pp \leftarrow \mathsf{Setup}(1^\lambda)$
2. $sk \leftarrow \mathsf{Keygen}(pp)$
3. $(m, \tau) \leftarrow \mathscr{A}^{\mathscr{O}_{\mathsf{Mac}}}(pp)$ ($\mathscr{A}$ has access to a tag generation oracle $\mathscr{O}_{\mathsf{Mac}}$)
4. If m has already been queried to $\mathscr{O}_{\mathsf{Mac}}$, then return 0
5. Return $\mathsf{Verify}(pp, sk, m, \tau)$

Figure 2.3: UF-CMA security experiment.

Adversary $\mathscr{A}$'s probability of success in the UF-CMA game, also known as his *advantage*, is denoted by $\mathsf{Adv}^{\text{UF-CMA}}_{\mathscr{A}}(1^\lambda)$. It is defined as $\mathsf{Adv}^{\text{UF-CMA}}_{\mathscr{A}}(1^\lambda) = \mathsf{Pr}[\text{Exp}^{\text{UF-CMA}}_{\mathscr{A}}(1^\lambda) = 1]$.

Probabilistic MAC. A MAC scheme is said to be *probabilistic* if for a given message m, there multiple corresponding valid tags. The security notion for a probabilistic MAC is that of *unforgeability under chosen message and verification attack* (UF-CMVA). It is a stronger security notion as is encapsulates the fact that an adversary $\mathscr{A}$, which has access to a Mac **and** a Verify oracle, can output a valid tag τ on a message m that has not been queried to the Mac oracle only with negligible probability. The Verify oracle is a query/response oracle where any entity can submit a message/tag pair (m,τ), and the oracle sends back 1 (if the tag is valid) or 0 otherwise. The UF-CMVA security experiment is depicted in Figure 2.4.

$\text{Exp}^{\text{UF-CMVA}}_{\mathscr{A}}(1^\lambda)$:

1. $pp \leftarrow \mathsf{Setup}(1^\lambda)$
2. $sk \leftarrow \mathsf{Keygen}(pp)$
3. $(m,\tau) \leftarrow \mathscr{A}^{\mathscr{O}_{\mathsf{Mac}},\mathscr{O}_{\mathsf{Verify}}}(pp)$ ($\mathscr{A}$ has access to both a tag generation and verification oracle)
4. If m has already been queried to $\mathscr{O}_{\mathsf{Mac}}$, return 0
5. Return $\mathsf{Verify}(pp,sk,m,\tau)$

Figure 2.4: UF-CMVA security experiment.

The adversary's advantage is defined as $\mathsf{Adv}^{\text{UF-CMVA}}_{\mathscr{A}}(1^\lambda) = \mathsf{Pr}[\text{Exp}^{\text{UF-CMVA}}_{\mathscr{A}}(1^\lambda) = 1]$.

2.4.2.2 Algebraic MACs

The concept of an algebraic MAC introduced by Dodis et al. [DKPW12], and later generalized by Chase et al. [CMZ14] generalized algebraic MACs to support n attributes, i.e. generate a MAC on n message blocks $(m_1,...,m_n)$. An algebraic MAC defines a MAC using group operations rather than block ciphers or hash functions. The security is based on number-theoretic assumptions.

Let $\mathbb{F}_p$ be the finite field of prime order p, G a cyclic group of order p, and pp the public parameters. The algebraic MAC scheme introduced in [DKPW12] is described as follows:

- $\mathsf{KeyGen}(pp)$: choose $(x_0,x_1) \in \mathbb{F}_p^2$ and set the secret key $sk = (x_0,x_1)$;
- $\mathsf{Mac}(sk,m)$: to compute the MAC of a message $m \in \mathbb{F}_p$, choose $u \in G$ and compute $u' = u^{x_0+x_1 \cdot m}$. Set the tag as (u,u');
- $\mathsf{Verify}(sk,m,(u,u'))$: to verify the tag (u,u') for a message m, check whether $u' = u^{x_0+x_1 \cdot m}$.

Algebraic MACs are used in settings where the signer and the verifier share a secret key, thus providing constructions with a more efficient verification step. In such settings, algebraic MACs can be used instead of public key signatures as building blocks for efficient authentication protocols, most notably anonymous credentials or direct anonymous attestation.

2.4.2.3 Aggregate MACs

Following the introduction of aggregate signatures by Boneh et al. [BGLS03], aggregate MACs were introduced by Katz and Lindell [KL08] as the symmetric-key alternative. In the symmetric-key setting, a collection of n users share a secret key sk_i with the final verifier. An *aggregate* MAC scheme allows n users to generate n tags on n potentially different messages. An *aggregate algebraic MAC* is an aggregate MAC construction instantiated with an *algebraic MAC*. Each tag is an element of an algebraic group, and the key generation function also outputs a public parameter $iparam = pk_i$ related to each secret key sk_i.
Let λ be the security parameter.

Definition 2.29. (Aggregate MAC) An aggregate MAC scheme comprises the following tuple of probabilistic polynomial-time algorithms (KeyGen, Mac, Verify, Agg, AggVerify):

- KeyGen: takes as input the security parameter 1^λ and returns the pair (sk_i, pk_i) for a particular sender, where sk_i is the secret key and pk_i the public parameter.
- Mac/Verify: Mac and Verify are the same as in a standard message authentication scheme.
- AggMac: upon receiving two sets of messages $M_1 = \{m_1^1, ..., m_{n_1}^1\}$ and $M_2 = \{m_1^2, ..., m_{n_2}^2\}$, associated with tags τ_1 and τ_2 respectively, outputs an aggregate tag τ on $M = M_1 \cup M_2$;
- AggVerify: takes as input a set of keys $sk = \{sk_1, ..., sk_t\}$, a set of messages $M = \{m_1, ..., m_n\}$, and a tag τ. It returns a bit $b \in \{0,1\}$.

An aggregate MAC scheme is complete if the following two conditions are verified:

- For any $\lambda \in \mathbb{N}$, any $(sk_i, pk_i) \leftarrow \mathsf{KeyGen}(1^\lambda)$, any message m, $\mathsf{Verify}(sk_i, m, \mathsf{Mac}(sk_i, m)) = 1$.
- Let M_1 and M_2 be two sets of messages with $M_1 \cap M_2 = \emptyset$, let Sk_1 and Sk_2 be two sets of keys, and let $M = M_1 \cup M_2$ and $Sk = Sk_1 \cup Sk_2$. If $\mathsf{AggVerify}(Sk_1, M_1, \tau_1) = 1$ and $\mathsf{AggVerify}(Sk_2, M_2, \tau_2) = 1$, then $\mathsf{AggVerify}(Sk, M, \mathsf{AggMac}(M_1, M_2, \tau_1, \tau_2)) = 1$.

Security definition. The security definition for an aggregate algebraic MAC is *existential unforgeability under adaptive chosen-message and verification attack* (EUF-CMVA). Existential unforgeability in this case means that no adversary should be able to forge an aggregate MAC on a set of messages of his choice, by a set of n users, considering he knows at most $n-1$ of those secret keys. Let $\mathscr{A}$ be an adversary against the EUF-CMVA security of an aggregate algebraic MAC. $\mathscr{A}$ has access to two oracles: $\mathscr{O}_{\mathsf{AggMac}}$ that takes as input a set of messages and generates a valid tag, and $\mathscr{O}_{\mathsf{Verify}}$ that takes as input a set of secret keys, a set of messages and an aggregate tag, and returns $b \in \{0,1\}$.

Definition 2.30. (EUF-CMVA Security) For an aggregate MAC (KeyGen, Mac, Verify, AggMac, AggVerify), we define a probabilistic polynomial-time adversary $\mathscr{A}$'s advantage $Adv_{\mathscr{A}}^{\text{EUF-CMVA}}(1^{\lambda})$ to be his probability of success in the following experiment:

$\text{Exp}_{\mathscr{A}}^{\text{EUF-CMVA}}(1^{\lambda})$:

1. *Setup:* $\mathscr{C}$ initializes a list KeyList. He then runs Setup to generate pp and KeyGen to generate the algebraic MAC key pair (sk^*, pk^*). He sends $\mathscr{A}$ the public parameter pk^*;
2. *Join Queries:* $\mathscr{A}$ adaptively requests to append public parameters pk_i to KeyList;
3. *Aggregate Tag Queries:* $\mathscr{A}$ adaptively asks for the aggregate tag on at most q messages $(\{m_1, ..., m_q\})$ under the challenge secret key sk^*. The queries work as follows: for each query, $\mathscr{A}$ provides an aggregate tag τ_i on $(m_{i,1}, ..., m_{i,n_i})$ under the secret keys $(sk_{i,1}, ..., sk_{i,n_i})$ where the corresponding public keys $(pk_{i,1}, ..., pk_{i,n_i})$ belong to KeyList. $\mathscr{C}$ returns the aggregate tag on $(m_{i,1}, ..., m_{i,n_i}, m_i)$ produced using sk^*.
4. *Verification Queries:* $\mathscr{A}$ adaptively asks for the verification on a tag τ_i. If τ_i was not generated using sk^*, $\mathscr{C}$ returns the result of the verification function $b \in \{0, 1\}$;
5. *Output:* Eventually, $\mathscr{A}$ outputs an aggregate tag τ on messages $(m_1^*, ..., m_n^*)$ generated under $(sk_1, ..., sk_n)$. He wins the game if the following conditions hold:
   - $\text{AggVerify}(\{sk_1, ..., sk_n\}, \{m_1, ..., m_n\}, \tau) = 1$;
   - For all $pk_i \neq pk^*$, $pk_i \in$ KeyList;
   - There exists an index $j^* \in \{1, ..., n\}$ such that $pk_{j^*} = pk^*$ and $m_{j^*}^*$ has never been queried to the aggregate MAC oracle (i.e. for $i \in \{1, ..., q\}, m_i \neq m_{j^*}^*$).

Figure 2.5: EUF-CMVA security experiment.

2.5 Public-Key Cryptographic Primitives

In this section, we introduce public key cryptographic primitives, which are used in a variety of applications, and notably in the design of secure protocols in Part II of this thesis.

2.5.1 Public key encryption

The solution to the secure key exchange problem mentioned in Section 2.4 is provided by public key (or *asymmetric*) encryption. Indeed, public key encryption allows different parties to securely exchange the symmetric key used in the message encryption process. Public key cryptosystems have a wide range of applications in cryptography, notably, they allow the construction of primitives

such as *digital signatures*, used to authenticate messages by emulating real life signatures. Public key cryptosystems make use of a *public key* (known to every entity) in the encryption process and a *private key* (known only to the recipient of the encrypted message) in the decryption process.

2.5.1.1 Definition

Public key encryption, introduced by Diffie and Hellman [DH76], allows each entity to possess a pair of public/private key. A user distributes his public key, which is used to encrypt messages, and uses his private key to decrypt messages received from other entities. In this thesis, we use public key cryptosystems in order to build secure protocols for distributed architectures.

Definition 2.31. (Public key encryption scheme) A public key encryption scheme comprises the following algorithms:

- $\mathsf{Setup}(1^\lambda)$: A probabilistic algorithm which takes as input 1^λ, where λ is a security parameter, and outputs the public parameters of the system denoted by pp;

- $\mathsf{Keygen}(pp)$: A probabilistic algorithm which takes as input the public parameters pp, and generates a public and private key pair (pk, sk);

- $\mathsf{Encrypt}(pp, pk, m)$: A deterministic or probabilistic algorithm which takes as input the public parameters pp, the public encryption key pk, a message m, and outputs the ciphertext c. If $\mathsf{Encrypt}$ is *probabilistic*, it involves a random coin r and is denoted $\mathsf{Encrypt}(pp, pk, m, r)$;

- $\mathsf{Decrypt}(pp, sk, c)$: A deterministic algorithm which takes as input the public parameters pp, the private decryption key sk, and a ciphertext c. It outputs the message m.

2.5.1.2 Security notions

The security of cryptographic algorithms is evaluated based on the attacker's goal, as well as his attacking power. The two main attack goal for public key encryption schemes is *Indistinguishability* [PP04].

Indistinguishability (IND). Informally, the *indistinguishability* property encapsulate the notion that given two messages, the adversary is able to determine which one corresponds to a given ciphertext only with negligible probability.
The indistinguishability game is depicted in Figure 2.6. Let $\pi = (\mathsf{Keygen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ be an encryption scheme. Let $\mathscr{A}$ be a probabilistic adversary whose running time is bounded by t. We say that π is (λ, ϵ)-IND secure if any probabilistic polynomial time adversary $\mathscr{A}'s$ advantage $Adv^{\mathrm{IND}}_{\mathscr{A}}(1^\lambda)$, defined as $\mathsf{Adv}^{\mathrm{IND}}_{\mathscr{A}}(1^\lambda) = 2 \cdot |\mathsf{Pr}_{b,r}[\mathrm{Exp}^{\mathrm{IND}}_{\mathscr{A}}(1^\lambda) = 1] - \frac{1}{2}|$ is less than ϵ, where ϵ is a negligible function.

$\text{Exp}^{\text{IND}}_{\mathscr{A}}(1^\lambda)$:

1. $pp \leftarrow \text{Setup}(1^\lambda)$
2. $(pk, sk) \leftarrow \text{Keygen}(pp)$
3. $(m_0, m_1) \leftarrow \mathscr{A}(pk)$ ($\mathscr{A}$ outputs two messages m_0 and m_1 such that $|m_0| = |m_1|$)
4. $b \leftarrow \{0,1\}$ (the challenger selects a bit $b \in \{0,1\}$ at random)
5. $c \leftarrow \text{Encrypt}(pp, pk, m_b, r)$
6. $b' \leftarrow \mathscr{A}(m_0, m_1, c)$
7. Output 1 if $b' = b$ and 0 otherwise

Figure 2.6: IND security experiment.

In real life, the attacker may have access to a number of advantages that can be qualified as his attacking power. We distinguish the following notions based on the adversary's power.

Chosen Plaintext Attack (CPA). In public key cryptography, the adversary, as anybody, has access to the encryption key and can therefore encrypt messages of his choice. This precise attack is called *Chosen Plaintext Attack* (CPA);

Chosen Ciphertext Attack (CCA1). Also known as *Lunchtime Attack*, this attack model provides the adversary with a decryption oracle before he is given the challenge ciphertext. He can therefore decrypt any ciphertext of his choice, up until he is given the challenge;

Adaptive Chosen Ciphertext Attack (CCA2). The adaptive chosen ciphertext adversary has access to a decryption oracle at any moment during the attack, with the restriction that he can decrypt any ciphertext of his choice, except the challenge ciphertext (which would render the attack trivial), and thus adapt his queries during the attack.

In order to define the security of a scheme, we must combine the desired goal with the attacking power of the adversary when building the security model. A scheme is only secure with respect to a security notion XX-YY defined by the combination of an attack goal XX, where XX=IND and the attacker's power YY, where YY∈{CPA,CCA1,CCA2}. The relations between the different security notions was first formalized by Bellare et al. [BDPR98]. The ideal security goal for an encryption scheme is IND-CCA2 security.

2.5.1.3 Threshold cryptosystems

Threshold cryptosystems [DSDFY94, FS01, HMR+19] are special public key cryptosystems, where the decryption key is shared between different decryption entities. Compared to traditional encryption schemes, a threshold encryption scheme has two additional components, namely a distributed generation of the public key and sharing of the corresponding decryption key, as well as a decryption from a shared representation of the key. Threshold cryptosystems are used in applications

where it is not suitable to assign the decryption capability to a single entity. In cases for example where some recipients are only partially trusted. The decryption key in a threshold cryptosystem is generated by and shared among n participants. Let $t \leq n$ be the threshold, in order to decrypt a message, at least t participants must take part in the decryption process with their respective shares of the decryption key.

In the design of our protocols, we make use of the threshold version of the Paillier cryptosystem [Pai99] in the two-party malicious setting. The security of the scheme is based on the Decisional Composite Residuosity (DCR) assumption described in Section 2.2.3. The Paillier cryptosystem and the RSA cryptosystem share the same public/private key structure, namely a composite n and its factorization. Therefore, a distributed RSA key generation mechanisms provides a distributed algorithm for a Paillier key generation protocol. Hazay et al. [HMR+19] have proposed the first threshold Paillier cryptosystem in the two-party setting, with notably a secure distributed generation of an RSA composite function, a distributed generation of the secret key shares function, as well as a distributed decryption function.

Paillier cryptosystem. The Paillier cryptosystem is defined as follows:

- Keygen: choose a, b be two large primes such that $a, b > 2$, $a \nmid (b-1)$ and $b \nmid (a-1)$. Compute $n = ab$, $\lambda = lcm(a-1, b-1)$, and $g = (1+n)$. The public key is $pk = (n, g)$, and the secret key $sk = (\lambda)$;
- Encrypt(m,pk): select $r \in \mathbb{Z}_n^*$, and compute the ciphertext $c = g^m \cdot r^n \bmod n^2$;
- Decrypt(c,sk): compute $\frac{L(c^{sk} \bmod n^2)}{L(g^{sk} \bmod n^2)} \bmod n = m$, where the function L is defined as $L(x) = \frac{x-1}{n}$.

Homomorphic property. The Paillier cryptosystem is additively homomorphic. Indeed, given two ciphertexts $c_1 = g^{m_1} \cdot {r_1}^n \bmod n^2$ and $c_2 = g^{m_2} \cdot {r_2}^n \bmod n^2$, the product of the ciphertext $c = c_1 \cdot c_2 = g^{m_1+m_2} \cdot (r_1 \cdot r_2)^n \bmod n^2$ encodes the message $(m_1 + m_2)$.

2.5.2 Digital Signature

A digital signature is the public key equivalent of a MAC. A signature scheme allows any entity to authenticate a message by generating a publicly verifiable signature on the message. The scheme guarantees the authenticity of the message, as well as its integrity. The signer obtains a public/private key pair (pk, sk), and generates signatures using his private key sk. The recipient of a signed message is able to publicly verify the validity of the signature using the public key pk.

2.5.2.1 Definition and security notions

Definition 2.32. (Digital Signature) A digital signature scheme is comprised of the following algorithms:

- Setup(1^λ): a probabilistic algorithm that takes as input the security parameter 1^λ, and generates the system parameters pp;

- Keygen(pp): a probabilistic algorithm that takes as input the system parameters pp, and generates the signer's public/private key pair (pk, sk);

- Sign(pp, sk, m): an algorithm takes as input the public parameters pp, a message to be signed m, and the private signing key sk. It then returns a signature σ on m, generated with sk;

- Verify(pp, pk, m, σ): takes as input the public parameters pp, the message m, the signature σ, and the public verification key pk. It outputs 0 or 1 depending on whether the verification step succeeds or not.

A digital signature scheme must satisfy the following properties:

- *Validity*: if a signature σ was generated on a message m using the private key sk, then Verify(pp, pk, m, σ) outputs 1;

- *Integrity*: if any change is made to the message m upon generating the signature σ, then Verify(pp, pk, m, σ) outputs 0 with overwhelming probability;

- *Non-repudiation*: given a signature σ generated with sk, Verify(pp, pk, m, σ) outputs 1 with overwhelming probability. In other words, a signer cannot deny signing a message upon generating the signature.

The adversary against a MAC or a digital signature scheme is called a *forger*. Similarly to encryption schemes, the security of a signature scheme depends on the forger's goal, and his power. We distinguish the following forger goals:

Total Break: the forger can recover the private key, and therefore generate a digital signature on any message of his choosing;

Universal Forgery (UF): the forger can generate a valid signature on any message without knowing the private key;

Selective Forgery (SF): the forger can produce a valid signature on a message m of his choosing, without knowing the private key, and *prior* to starting the attack;

Existential Forgery (EUF): the goal of the forger is to generate a signature on a random message of his choice without access to the private key.

The attacking power of the adversary can be classified in one of the following categories:

No Message Attack (NMA): the attacker only knows the signer's public key;

Known Message Attack (KMA): the attacker knows the public key of the signer, as well as a set of valid message/signature pairs;

Chosen Message Attack (CMA): the attacker in the security game has access to a signing oracle that outputs valid signatures on messages of his choice, and can adapt his signatures with respect to previously generated signatures. As with the previous notions, the attacker has also access to the signer's public key.

A digital signature scheme is considered to be secure if it is *existentially unforgeable under chosen message attack* (EUF-CMA). In other words, an adversary that has access to a signing oracle should not be able to output a valid signature σ on a random message m that has not been queried to the oracle. The formal EUF-CMA security experiment is depicted in Figure 2.7.

$\mathsf{Exp}^{\text{EUF-CMA}}_{\mathscr{A}}(1^\lambda)$:

1. $pp \leftarrow \mathsf{Setup}(1^\lambda)$
2. $(sk, pk) \leftarrow \mathsf{Keygen}(pp)$
3. $(m, \sigma) \leftarrow \mathscr{A}^{\mathscr{O}_{\mathsf{Sign}}}(pk)$ ($\mathscr{A}$ has access to a signing oracle $\mathscr{O}_{\mathsf{Sign}}$)
4. If m has already been queried to $\mathscr{O}_{\mathsf{Sign}}$, then return 0
5. Return $\mathsf{Verify}(pp, pk, m, \tau)$

Figure 2.7: EUF-CMA security experiment.

The adversary $\mathscr{A}$'s advantage is defined as $\mathsf{Adv}^{\text{EUF-CMA}}_{\mathscr{A}}(1^\lambda) = \mathsf{Pr}[\mathsf{Exp}^{\text{EUF-CMA}}_{\mathscr{A}}(1^\lambda) = 1]$.

For the construction of our protocols in this thesis, we will make use of the following digital signature schemes:

Schnorr signature. The Schnorr signature [Sch89] is a simple an efficient signature scheme, proven secure under the discrete logarithm assumption. The scheme is defined as follows:

- $\mathsf{Setup}(1^\lambda, 1^{\lambda'})$: generates the public parameters $pp = (p, g, g, H)$, where p, q are two prime integers such that q divides $(p-1)$, $q \geq 2^\lambda$ and $p \geq 2^{\lambda'}$. g is an element of order q in $\mathbb{Z}_p$ and $H : \{0,1\}^* \leftarrow \mathbb{Z}_q$ is a collision-resistant cryptographic hash function;

- $\mathsf{Keygen}(pp)$: randomly select an element $x \leftarrow \mathbb{Z}_q^*$ and computes the corresponding public key $y = g^x \bmod p$. Sets the private key $sk = x$, and the public key $pk = y$;

- $\mathsf{Sign}(pp, m, x)$: randomly select an element $r \leftarrow \mathbb{Z}_q$ and compute $t = g^r \bmod p$, $c = H(t||m)$, and $s = r + cx \bmod q$. Sets the signature $\sigma = (c, s)$;

- $\mathsf{Verify}(pp, m, \sigma, y)$: verifies the signature by computing $t' = g^s y^{-c}$ and by computing that the equality $c = H(t'||m)$ holds.

The Schnorr signature is the non-interactive version of the Schnorr proof of knowledge of a discrete logarithm described in Section 2.5.4.1, applied to message m.

Boneh-Lynn-Shacham signature. The digital signature with the shortest output was proposed by Boneh, Lynn, and Shacham [BLS01] (BLS). The security of the BLS signature is based on the CDH assumption in elliptic curve groups where the DDH problem is easy but the CDH problem is intractable (also known as gap Diffie-Hellman groups). The signature generation step consists in performing a multiplication on an elliptic curve over a finite field. The signature verification step consists in computing a bilinear pairing on the curve. The scheme is defined as follows:

- $\mathsf{Setup}(1^\lambda)$: generates the public parameters $pp = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_3, g_1, g_2, e, H)$ where $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ are cyclic groups of prime order p (more precisely gap Diffie-Hellman groups where there exists an efficiently computable isomorphism from $\mathbb{G}_2$ to $\mathbb{G}_1$), g_1, g_2 are fixed generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively, and $e : \mathbb{G}_1 \times \mathbb{G}_2 \leftarrow \mathbb{G}_T$ is an efficiently computable bilinear map. $H : \{0,1\}^* \rightarrow \mathbb{G}_1$ is a full-domain hash function;

- $\mathsf{KeyGen}(pp)$: select a random $x \xleftarrow{\$} \mathbb{Z}_p$ and compute $v = g_2^x$. Set x as the private key and $v \in \mathbb{G}_2$ as the corresponding public key;

- $\mathsf{Sign}(pp, M, sk)$: to sign a message $M \in \{0,1\}^*$, compute $h = H(M)$ and $\sigma = h^x$. The signature is $\sigma \in \mathbb{G}_1$;

- $\mathsf{Verify}(pp, \sigma, M, pk)$: given the public key pk, the message M and the signature σ, compute $h = H(M)$ and check that the equality $e(h, v) = e(\sigma, g_2)$ holds. If so return 1, otherwise return 0.

Pointcheval-Sanders signature. The Pointcheval-Sanders signature [PS16] is a randomizable signature scheme proven secure in the generic group model. We describe the n-vector message version of the signature below:

- $\mathsf{Setup}(1^\lambda)$: takes as input a security parameter k, outputs $pp \leftarrow (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ the description of a type 3 pairing;

- $\mathsf{Keygen}(pp)$: selects $\tilde{g}_2 \xleftarrow{\$} \mathbb{G}_2^*$ and $(x, y_1, ..., y_{r+1}) \xleftarrow{\$} (\mathbb{Z}_p^*)^{r+2}$, computes $(\tilde{X}, \tilde{Y}_1, ..., \tilde{Y}_{r+1}) \leftarrow (\tilde{g}^x, \tilde{g}^{y_1}, ..., \tilde{g}^{y_{r+1}})$, and sets $sk \leftarrow (x, y_1, ..., y_{r+1})$ and $pk \leftarrow (\tilde{X}, \tilde{Y}_1, ..., \tilde{Y}_{r+1})$;

- $\mathsf{Sign}(sk, \mathbf{m} = (m_1, ..., m_r))$: selects $h \xleftarrow{\$} \mathbb{G}_1^*$, and $m' \xleftarrow{\$} \mathbb{Z}_p$ and outputs $\sigma \leftarrow (m', h, h^{(x + \sum_{j=1}^{r} y_j \cdot m_j + y_{r+1} \cdot m')})$;

- $\mathsf{Verify}(pk, \sigma, \mathbf{m} = (m_1, ..., m_r))$: parses σ as (m', σ_1, σ_2) and checks that $\sigma_1 \neq 1_{\mathbb{G}_1}$ and $e(\sigma_1, \tilde{X} \cdot \prod_{j=1}^{r} \tilde{Y}_j^{m_j} \cdot \tilde{Y}_{r+1}^{m'}) = e(\sigma_2, \tilde{g})$ are satisfied. If yes the algorithm outputs 1, otherwise it outputs 0.

The scheme is suitable for implementing privacy-preserving protocol due to the randomization property.

2.5.2.2 Group signature

The concept of a group signature was introduced by Chaum and Van Heyst [CH91]. A group signature scheme enables any member of a group to generate a digital signature on behalf of the group without revealing his personal identity. The construction is ideal for privacy-preserving schemes, as it allows signers to authenticate messages, whilst staying anonymous as members of a group. A trusted revocation authority can revoke a signer's anonymity in certain circumstances. A group signature scheme comprises a *group manager* in charge of member registration, *group members* that first register with the group manager to obtain their group signing keys prior to being able to generate anonymous signatures, and a *revocation authority* that is able to recover a group member's identity from a given signature. Formally, a group signature scheme is defined as follows:

Definition 2.33. (Group signature). A group signature scheme is comprised of the following algorithms:

- Setup(1^λ): a probabilistic algorithm that takes as input the security parameter 1^λ, and generates the system parameters pp;
- Keygen(pp): a probabilistic algorithm that takes as input the system parameters pp, and generates the public group key gpk, the group manager's secret key sk_{gm}, and the revocation authority's secret key sk_{ra};
- Join: an interactive protocol between a user and the group manager, at the end of which the user obtains his group signing key. In other words, on input pp and gpk, the algorithm outputs to the user his group signing key gsk_u, along with a group membership certificate ζ;
- Sign(pp, gsk_u, m, ζ): a probabilistic algorithm that on input a message m, a membership certificate ζ, and a user's group signing key gsk_u, outputs a signature σ on m;
- Verify(pp, gpk, m, σ): a deterministic algorithm that on input a group signature σ and the group public key gpk, outputs 1 (accept) or 0 (reject);
- Open($pp, gpk, sk_{ra}, m, \sigma$): a deterministic algorithm that on input a valid group signature σ, the corresponding message m, the group public key gpk, and the revocation authority's secret key sk_{ra}, outputs the identity of the group member who generated σ.

In addition to the properties of a basic digital signature scheme, a group signature scheme must satisfy the following properties:

- *Anonymity*: no entity should be able to recover a signer's identity except the revocation authority. In addition, no entity should be able to link two signatures generated by the same signer;
- *Traceability*: the Open algorithm should be able to trace any group signature to one of the signers;

- *Non-frameability*: no entity, even a set of colluding group members and the group manager, should be able to falsely link a signature to an honest group member who has not generated said signature.

2.5.2.3 Aggregate signature

Aggregate signatures allow a set of n signers to generate a single aggregate signature on their respective messages $(m_1,...,m_n)$. The verification of the aggregate signature is a proof of validity of all n signatures generated on all n messages.
The first aggregate signature scheme was introduced by Boneh, Gentry, Lynn, and Shacham (BGLS) [BGLS03]. Their scheme makes use of a full-domain hash function $\mathscr{H}: \{0,1\}^* \to \mathbb{G}_1$ modelled as a random oracle in the security proof. The BGLS aggregate signature is comprised of the following algorithms: (Keygen, Sign, Verify, AggSign, AggVerify), and is defined as follows:

Boneh-Gentry-Lynn-Shacham Aggregate Signature.

- KeyGen. Each user picks a random $x \xleftarrow{\$} \mathbb{Z}_p$, and computes $v \leftarrow g_2^x$. The secret key sk is x and the public key pk is set as $v \in \mathbb{G}_2$;
- Sign. For a particular user, given sk, the message $M \in \{0,1\}^*$, compute $h \leftarrow \mathscr{H}(M)$; $\sigma \leftarrow h^x$. The signature is $\sigma \in \mathbb{G}_1$;
- Verify. Given the user's public key $v \in \mathbb{G}_2$, the message M, and a signature σ, the algorithm computes $h \leftarrow \mathscr{H}(M)$, accepts if $e(h,v) = e(\sigma, g_2)$;
- AggSign. For each user $u_i \in U$ that provided a signature σ_i on $M_i \in \{0,1\}^*$ distinct messages of its choice, compute $\sigma \leftarrow \prod_{i=1}^{k} \sigma_i (= \prod_{i=1}^{k} \mathscr{H}(M_i)^{x_i})$, where $k = |U|$. The aggregate signature is $\sigma \in \mathbb{G}_1$;
- AggVerify. Given the public keys v_i of each user $u_i \in U$, the set of messages $\{M_1,...,M_k\}$, and the aggregate signature σ:
  - Check that all M_i's are distinct;
  - Accept if $e(\sigma, g_2) = \prod_{i=1}^{k} e(h_i, v_i)$, where $h_i \leftarrow \mathscr{H}(M_i)$.

Security model for aggregate signatures The security model for aggregate signature schemes is based on the inability of any polynomial-time adversary to existentially forge an aggregate signature (EUF-CKA) against chosen key attacks. The adversary is not capable of forging an aggregate signature on messages of his choice by some set of users. Boneh et al. formalised this notion with the aggregate chosen-key security model. In this model, the adversary $\mathscr{A}$ can choose all public keys except the challenge public key, and a signing oracle on the challenge key.

Setup. $\mathscr{A}$ is provided with a public key pk_1 chosen at random.

Queries. $\mathscr{A}$ adaptively requests signatures with pk_1 on messages of his choice.

Response. $\mathscr{A}$ outputs $k-1$ additional public keys $pk_2,...,pk_k$ where $pk_1,...,pk_k$ are included in $\mathscr{A}$'s aggregate signature. $\mathscr{A}$ also outputs k distinct messages $M_1,...,M_k$ and an aggregate signature σ by the k users each on its respective message.
$\mathscr{A}$ wins if σ is a valid aggregate signature on $M_1,...,M_k$ under the keys $pk_1,...,pk_k$, and $\mathscr{A}$ has not previously requested a signature on M_1 by pk_1 to the signing oracle. He's advantage $\mathsf{Adv\ AggSig}_{\mathscr{A}}$ is taken over the random tosses of the key generation algorithm and of $\mathscr{A}$.

2.5.3 Commitment scheme

Commitment schemes were first formalized by Brassard, Chaum, and Crépeau [BCC88] in 1988. Intuitively, commitment schemes can be described as lockable boxes. In the first phase called the *commitment phase*, the sender locks a message in the box and sends it to the receiver. The second phase called the *opening phase*, the sender sends the decommitment key to the receiver who is then able to open the box and read the message. Prior to sending the decommitment key, the receiver is not able to read the message, while the sender is not able to change the message after the commitment phase. The first property is called *hiding*, and the second is the *binding* property. A commitment scheme is a said to be *non-interactive* if it is defined with only two algorithms, namely **Commit** and **Open**, as opposed to the *interactive* definition where both the commitment and opening phases are executed as cryptographic protocols between the sender and the receiver.

2.5.3.1 Definition

Definition 2.34. (Commitment scheme). A non-interactive commitment scheme is defined by the following three algorithms:

- $\mathsf{Setup}(1^\lambda)$: outputs the public parameters of the system for security parameter λ;
- $\mathsf{Commit}(\mathsf{pp}, m)$: takes as input public parameters pp and a message m. Outputs the commitment value C along with the opening value d;
- $\mathsf{Open}(C, m, d)$: takes as input the commitment C, the opening value d, and the message m. Outputs "yes" or "no" depending on whether the verification succeeds or not.

In accordance with the *hiding* property, the opening value d is only sent to the receiver at the opening time. The hiding property states that the commitment C does not reveal any information about the message m. The *binding* property states that no adversary can output a message $m' \neq m$ along with an opening value d' such that C opens to both (m,d) and m',d'.

2.5.3.2 Security

More formally, we describe two security properties as follows:

- **Perfectly Hiding, Computationally binding:** a commitment scheme is *perfectly hiding* and *computationally binding* if the hiding property holds against a computationally unbounded receiver $\mathscr{R}$, while the binding property holds against a bounded sender $\mathscr{S}$;

- **Computationally Hiding, Perfectly Binding:** a commitment scheme is *computationally hiding* and *perfectly binding* if the hiding property holds against a computationally bounded receiver $\mathscr{R}$, while the binding property holds against any sender $\mathscr{S}$.

The notion of perfectly-binding was formalized by Naor [Nao91], and the first perfectly-hiding commitment scheme was introduced by Naor, Ostrovsky, Venkatesan, and Yung [NOVY92]. A commitment scheme cannot be both perfectly hiding and perfectly binding [Dam98].

An example of a perfectly hiding and computationally binding commitment scheme is the Pedersen commitment [Ped91].

Definition 2.35. (Pedersen commitment).

- $\mathsf{Setup}(1^\lambda)$: outputs the public parameters $pp = (p, \mathbb{G}, g, h)$ where $\mathbb{G}$ be a cyclic group of prime order p, and $g, h \in \mathbb{G}$ two random generators of $\mathbb{G}$;

- $\mathsf{Commit}(pp, m)$: generates a random element $r \in \mathbb{Z}_p$. Computes the commitment $C = g^m h^r$, and outputs C;

- $\mathsf{Open}(pp, m, (C, r))$: verifies the commitment by checking the equality $C = g^m h^r$.

Commitment schemes are extremely useful when designing privacy-preserving schemes. Notably, they are useful building blocks in construction efficient zero-knowledge protocols [CD97].

2.5.4 Zero knowledge proof

First introduced by Goldwasser, Mical, and Rackoff [GMR85, GMR89], and later formalized by Feige, Fiat, and Shamir [FFS88], a zero-knowledge proof of knowledge is a protocol by which an entity (called prover) proves to another entity (called verifier) that he knows a value x, which belongs to a given language $\mathscr{L}$, without revealing any information about x. They are practical protocols that allow to prove the knowledge of secrets without leaking any information about said secret. As such, they are useful building blocks for privacy-preserving cryptographic protocols.

2.5.4.1 Zero-Knowledge Proof of Knowledge

Definition 2.36. (Zero-Knowledge Proof of Knowledge (ZKPK)). A zero-knowledge proof of knowledge is an interactive protocol between a prover and a verifier, where the verifier can attest that the prover knows a secret which verifies a given statement. The protocol verifies the following three properties:

- **Completeness:** a proof generated by an honest prover is accepted by the verifier with overwhelming probability;

- **Soundness:** a proof generated by a prover who does not know the secret is accepted by the verifier with negligible probability;

- **Zero-knowledge:** given a true statement, a verifier learns nothing more than the fact that the statement is true. This is modeled by showing that every verifier has a simulator S, that on input the statement to be proven, can output a transcript that is indistinguishable from the actual protocol interaction between an honest prover and the verifier (i.e. without actually knowing the secret).

We use the Camenisch, Stadler [CS97] notion for proofs of knowledge, namely:

$$\text{PoK}\{\alpha,\beta,...\text{:statement on } \alpha,\beta,...\}$$

Whereby $\text{PoK}\{\alpha,\beta : \text{statement on } \alpha,\beta\}$ denotes a proof of knowledge of secrets α,β, the statement on the right side of the colon corresponds to the statement of knowledge about the secrets.

In this thesis use honest verifier ZKPK, also known as $\Sigma-$protocols, which are three-move ZKPK protocols between a prover and verifier. We present an example of $\Sigma-$protocol below.

Schnorr identification protocol.

Introduced by Schnorr in 1990 [Sch91], this identification protocol presents a number of security and efficiency advantages for building privacy-preserving authentication protocols. In recent years (since the expiration of the proprietary patent), it has gained additional interest in the cryptographic community, notably for applications such as transaction authentication and validation in Bitcoin [MPSW19]. The scheme allows a prover to prove the knowledge of a discrete logarithm x in base g of a public value $y = g^x$. The protocol is described as follows:

Given the public parameters (p,q,g) where p, q are two prime integers such that q divides $q-1$, and g is an element of $\in \mathbb{Z}_p^*$ of order q. The public parameter $y = g^x \bmod p$ is used by the prover P to prove knowledge of the secret $x \in \mathbb{Z}_q^*$.The verifier V also has access to the public parameters.

1. P selects a random $a \leftarrow \mathbb{Z}_q^*$ and sends $t = g^a \bmod p$ to V;
2. V selects a challenge $c \leftarrow \mathbb{Z}_q$ and sends it to P;
3. P sends the response $z = a + cx \bmod q$ to V. V checks that $t = g^z y^{-c} \bmod p$ and accepts if and only if it is the case.

2.5.4.2 Non-Interactive Zero-Knowledge Proof of Knowledge (NIZKPK)

In this thesis, we are interested in non-interactive zero-knowledge proofs of knowledge (NIZKPK), for their ability to generate efficient one-pass authentication schemes.

Fiat-Shamir heuristic

In 1986, Fiat and Shamir [FS86] introduced a new method, now known as the Fiat-Shamir heuristic, to transform interactive zero-knowledge proofs of knowledge into the corresponding non-interactive version. In the non-interactive version, the prover does not wait for the challenge from

the verifier, and instead computes it using a hash function. non-interactive zero-knowledge proofs of knowledge (NIZKPK) are also called Signatures of Knowledge (SoK), as they are essentially signature schemes.

Non-interactive Schnorr identification protocol.

A non-interactive version of the Schnorr identification protocol (or the Schnorr signature of knowledge) is described as follows:

Given the public parameters (p, q, g) where p, q are two prime integers such that q divides $q-1$, and g is an element of $\in \mathbb{Z}_p^*$ of order q. The public parameter $y = g^x \bmod p$ is used by the prover P to prove knowledge of the secret $x \in \mathbb{Z}_q^*$.The verifier V also has access to the public parameters. Both parties are given the description of a collision resistant hash function $H : \{0,1\}^* \to \mathbb{Z}_q$.

1. P selects a random $a \leftarrow \mathbb{Z}_q^*$ and computes $t = g^a \bmod p$. He then computes a challenge $c = H(t||g||y)$, and finally the response $z = a + cx \bmod q$ which he sends to V along with the challenge c;

2. V computes the value $t' = g^z y^{-c} \mod p$ and checks that $c = H(t'||g||y)$. V accepts if and only if all verifications are successful.

2.6 Conclusion

This chapter introduces the mathematical tools, as well as the cryptographic notions and primitives used in the construction of protocols in this thesis. We notably present the concept of provable security, which provides a formalized method for proving the security of cryptographic schemes. In the next chapter, we provide review the concept of *attestation* and its security guarantees. We review how attestation schemes can be leveraged to build secure and privacy-preserving authentication schemes suitable for resource-constrained environments using secure cryptographic protocols.

CHAPTER 3

BACKGROUND ON ATTESTATION

In the context of Internet of Things applications, where devices are widely deployed and process potentially sensitive data, protection against malware attacks has become increasingly important. It is a challenging goal for a number of IoT applications, as now more than ever, these embedded systems are exposed to threats, with potentially devastating impact on the processes and applications. In parallel, devices process a large amount of private data on users depending on the application, and the privacy of such data must also be ensured. In this chapter, we describe security mechanisms for ensuring the authenticity and integrity of devices and communications in a variety of IoT applications. In particular, we describe different applications of the attestation mechanism to ensure the authenticity of devices and users, as well as the integrity of data. We finally elaborate on its application to building secure and privacy-preserving authentication protocols.

Contents

3.1 Building trust in IoT systems

3.1.1 Context

The Internet of Things encompasses a variety of application domains ranging from large-scale applications such as Vehicular Ad Hoc Networks (VANET), to applications involving personal wearable devices such as heart monitoring devices in Smart health. Devices in those applications are resource-constrained low-end systems with limited storage space. They are limited in terms of computational capabilities, hence affecting the implementation of cryptographic algorithms in such environments. Attacks on IoT devices may have potentially devastating repercussions, as they may affect the entire control system as it was the case with the *Stuxnet* attack [FMC10]. Verifying the integrity of IoT devices, i.e. checking that their internal software has not been targeted by a malware, is an important security goal. In this thesis, we analyze a security technique for verifying the integrity of systems called *remote attestation*. A second part of this thesis consists in leveraging attestation mechanisms to securely authenticate IoT devices in a privacy-friendly manner.

Privacy concerns

The pervasive deployment of devices in consumer-driven applications induces privacy concerns, which have had legal ramifications for the past few years. Data protection regulations and laws have been ratified by special commissions in many countries worldwide [OPC19, PIP16, USD17, EU]. These regulations impose strict restrictions on the exploitation of user personal data by public and private companies institutions alike. Companies that collect user personal data, such as identification information, location, and mobility data for their applications, must provide guarantees that the collection, storage, and processing is done using privacy-friendly methods. In cryptography, a number of tools can be used to implement privacy-preserving protocols, and notably privacy-friendly authentication. In the context of IoT, these mechanisms must be designed for the limitations imposed by IoT devices.

The goal in this thesis is to provide solutions based on tailored cryptographic primitives to address the security and privacy challenges. We describe in Sections 3.1.2 and 3.1.3 specific challenges for critical IoT application domains.

3.1.2 Challenges in Industrial Internet of Things

Traditional manufacturing systems, production engineering, and automation have now evolve to form what is known as the industrial Internet of Things (IIoT). Industrial control systems in IIoT are comprised of Cyberphysical Systems (CPS), that control physical processes in lieu of the traditional digital Programmable Logic Controllers (PLC). Cyberphysical systems communicate

over industrial networks, as well as the Internet, rendering them vulnerable to remote attacks [MR12, FMC10]. The utility sector in particular can be a specific target with potential consequences on the safety and security of systems and processes. In IIoT, the most important security challenges include preventing *system failure* [ZG13, SWW15] and ensuring system *availability*. These requirements imply protecting systems against denial-of-service attacks. Preventing these failures and detecting remote attacks requires preserving the integrity of control systems.

3.1.3 Challenges in Vehicular Ad Hoc Networks

A critical part of the deployment of Intelligent Transportation Systems (ITS) is the deployment of a secure and scalable communication network, denoted Vehicular Ad Hoc Network (VANET). The secure deployment of VANET requires assessing all security properties, namely *data confidentiality* and *integrity*, *trust* in the sender's data and *availability* on the network, *anonymity*, and *sender authentication* [DFGTR11, DFGMGTB14]. In addition to the security properties inherent to other communication networks, the enforcement of *road safety* and *user privacy* is a critical requirement in VANET [RPH06, PBH+08, CPHL07, EBPQ14]. In fact, a key component of VANET security is the trade-off between driver *privacy* and *liability*. Indeed, vehicles in the network must be able to trust messages stating the state of the road (e.g. roadblocks, traffic information, road hazards) emanating from each other and the roadside units. This security requirement can be solved using traditional authentication mechanisms such as Public Key Infrastructures (PKI). However, along with each authenticated message, vehicles provide a set of private information related to their location for instance. Such data can be used to trace users, or attempt targeted attacks. We distinguish the following security properties specific to the VANET application domain:

- **Message Authentication and Integrity:** vehicles must be able to identify and authenticate messages communicated over the network, as well as trust that said messages have not been tampered with;

- **User/Vehicle Authentication and Integrity:** receiving vehicles must trust that they are receiving messages from an authenticated vehicle in real-time. In other words, that messages have not been tampered with and were indeed generated within a reasonable time interval around the current time;

- **Non-repudiation:** a vehicle cannot deny sending a given message, it is therefore *liable* for sending message in a specific time frame;

- **Privacy:** this property encompasses two notions, namely data privacy and location privacy. Communications in VANET must ensure personal data privacy for the driver (prevent the ability to link a vehicle to a specific user), as well as location privacy (prevent the tracking of a specific user/vehicle).

In order to address the privacy/liability trade-off, two cryptographic mechanisms have been extensively studied, namely *vehicular mix-zones* and *PKI-based pseudonym schemes*. Both mechanisms

adopt a centralized approach to certificate and key management through a Certification Authority (CA).

3.2 Trusted Computing

In trusted computing , a computer system or device (also known as *platform*), such as a PC, smartphone or tablet, possesses a hardware-based Root of Trust (RoT). Via the root of trust, an entity interacting with the device has some assurance that the device is indeed in a trustworthy state and is behaving as expected. The RoT is used by the device to measure its internal state, which consists of hashes of its application code, additional verification code, and data stored in memory. The RoT leverages a tamper-proof hardware module to store the measurements, which are then securely forwarded to the entity during what is called the *attestation* process. The goal for a remote attacker often being to hijack a specific device by injecting malware in the system's code, providing proofs of the correct behavior of the device involves verifying its internal software state. The Trusted Computing Group is a standardization body specialized in the standardization of the specific functionalities to be implemented in trusted platforms.

Attestation is the security service at the core of trusted computing. The attestation process provides a secure and privacy-preserving way of guaranteeing the system's protection from external adversaries.

3.2.1 Trusted Execution Environment (TEE)

The attestation process leverages the secure crypto processors embedded in most modern computing systems to implement *Trusted Execution Environments* (TEE). The presence of a TEE ensures that a program runs in a secure enclave, isolating it from other applications running on the platform. The attestation process consists in the platform proving that it possesses a TEE. The TEE is embedded into a host platform, and provides the following guarantees:

- **Isolation:** a secure enclave separate from applications run on the host;
- **Secure code execution:** the atomic execution of any process in the TEE, notably attestation;
- **Secure storage:** storage only accessible by the TEE, and notably stores the cryptographic keys.

We distinguish the following implementations of a TEE:

ARM TrustZone (TZ)

TrustZone is the implementation of a secure cryptoprocessor mainly present in mobile devices. It enables the execution of trusted code in a secure execution environment, as well as secure storage of user credentials in various applications[ARM09, Win08].

Intel Software Guard Extension (SGX)

First introduced in 2013 [MAB+13b], then later deployed in the sixth generation Intel Core microprocessors [Cor15], the Intel SGX secure processor implementation provides software attestation by using containers as secure execution environments. Implemented on a host system, it proves to a user (or device) that the host she is communicating with is located in a secure container in a trusted hardware. The proof is a digital signature on a certificate using an attestation key.

Trusted Platform Module (TPM)

The Trusted Platform Module (TPM) is an ISO/IEC-standardized secure cryptoprocessor developed by the Trusted Computing Group (TCG). It is implemented as a chip mounted on a host system. The host system communicates with the TPM through a low-performance interface such as Low Pin Count (LPC). A TPM implements roots of trust for measurement, storage, and reporting, which use certificates generated on the TPM's public key, as well as an attestation protocol, to prove the accuracy of the data. The attestation protocol standardized by the Trusted Computing Group (TCG) is the Direct Anonymous Attestation (DAA) protocol introduced by Brickell, Camenisch, and Chen [BCC04], and described in Section 3.4.1.

Mobile TEE

Smartphones are being used to implement security-sensitive services such as mobile banking, physical access control [ABL13], and mobile ticketing [DPWD11, Gem18a, AGL+13, ALT+15]. The GSM Association notably considers that mobile phones offer secure, robust, and agile solutions for communication in Intelligent Transportation Systems (ITS) [Wal15]. Indeed, smartphones today are equipped with secure short-range communication capabilities such as Near Field Communication (NFC), enabled by Radio Frequency Identification (RFID). In addition, the **Universal Integrated Circuit Card (UICC)** (i.e. the SIM card) present in most smartphones constitutes a dedicated TEE with the following security and functional guarantees:

- Inexpensive compared to other trusted modules such as Hardware Security Modules (HSM);
- A programmable secure execution environment;
- Tamper protection and resistance against physical attacks (a tamper-evidence is generated whenever the card is physically tampered with);
- Protection against side-channel attacks such as fault injection, power, or timing attacks;
- A Secure Random Number Generator for various operations in randomized cryptographic schemes such as key generation;
- Certified to high levels of security: CC EAL5+ [Cri09] and FIPS 140-2 [FIP07];
- Specifications for secure remote provisioning [GSM19].

In this thesis, we will leverage the guarantees of a SIM card to build secure authentication, physical access control, and attestation solutions, that are suitable for constrained environments.

3.3 Remote Attestation

Remote Attestation (RA) is a two-party security protocol between an untrusted prover P and a trusted verifier V. The protocol is designed to remotely verify the internal software state of systems. The internal state of a device is composed of the application code, a hash of the memory, and Input/Output commands. As depicted in Figure 3.1, the attestation process consists in the verifier generating a challenge c, to which the prover replies with a hash value r of its internal application code. Remote attestation is particularly adapted to verify the *integrity* of devices in IoT applications, and the protocol must guarantee the following properties:

- **Authenticity:** the attestation response represents the *real* state of the system;

- **Freshness:** the attestation result represents the *current* state of the system.

In recent years, different remote attestation techniques have been studied for verifying the integrity of low-end embedded devices. They vary in terms of security guarantees, adversarial model, assumptions about the hardware features, and the potential application domain.

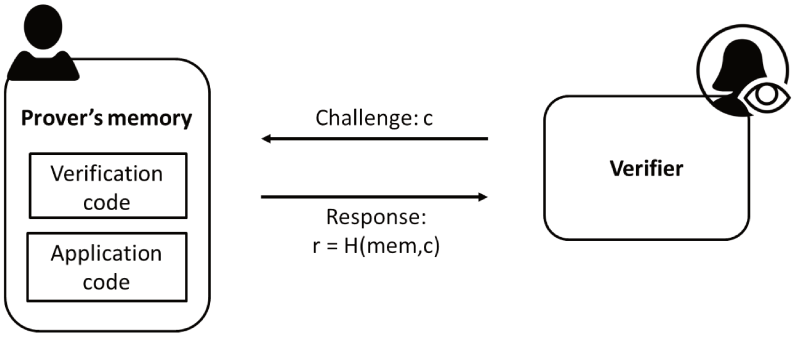


Figure 3.1: Single prover attestation protocol.

3.3.1 Software-based attestation

Software-based attestation protocols do not leverage any hardware module for the attestation process. The process therefore does not implement any cryptographic key management mechanism. Software-based attestation protocols necessitate that the verifier communicates directly with the prover, which requires a direct communication channel. Such configurations are impractical and hardly scalable. In addition, in software-based attestation, strong assumptions are made regarding the adversarial model [CFPS09]. This technique is however the most cost-effective as it requires no hardware module.

3.3.2 Hybrid attestation

Hybrid attestation has been introduced to overcome the challenges of software-based attestation. Hybrid attestation techniques employ a software/hardware architecture, with no specialized crypto processor, but only a read-only memory (ROM) to store attestation code [PT10]. The SMART [ETFP12] architecture is a hybrid architecture which provides a root of trust for low-end embedded devices without a specialized secure enclave. Cryptographic keys are therefore stored in a memory region that can only be accessed through SMART attestation code located in the ROM. Francillon et al. [FNRT14] proposed a different architecture relying on such minimal hardware. TrustLite [KSSV14] is a generalized version of SMART, where the memory protection unit (MPU) makes all the access control decisions to the data stored in the ROM. TyTAN [BEMS$^+$15] is a hybrid system with sender and receiver authentication, as well as inter process communication.

3.3.3 Hardware-based attestation

Attestation protocols that leverage the security properties of a hardware module (e.g. Trusted Platform Module (TPM)) are classified in the hardware-based attestation category. They provide strong security guarantees as we will show in Section 3.3.1, and are suitable for high-end devices. Due to the vulnerable nature of IoT devices, hardware-based security has become an attractive field when designing security protocols for different applications [SKAZ19]. The underlying hardware is mostly used for embedded system-based device identification, authentication, attestation, and key generation and management. The following trusted hardware modules are employed in different use cases:

- **Physically Unclonable Functions (PUF):** secure authentication in sensitive use cases such as VANET may require the use of non-volatile electrically erasable programmable read-only memory (EEPROM), or static random access memory (SRAM). They implement cryptographic operations in hardware. Physically Unclonable Functions (PUF) [GCDD02, HYKD14] are considered to be the hardware equivalent of cryptographic one-way functions, and have been used as a cheaper alternative to EEPROMs [SRR16];

- **Hardware Security Module (HSM):** designed to manage the lifecycle of cryptographic keys, HSMs are specialized crypto processors used for strong authentication. They are tamper-proof modules, ideal for implementing attestation protocols in applications such as VANET. They are however not cost-efficient for applications requiring the deployment of millions of devices.

- **Trusted Platform Module (TPM):** as described in Section 3.2.1, TPMs are used to provide security guarantees for the host system it is embedded in, notably the proof that a number of operations are being executed inside a secure enclave. The result of the attestation protocol convinces the verifier of the fact that the prover embeds a TPM whose public key has been certified by a trusted authority.

3.3.4 Swarm attestation

The single prover/single verifier model is not suitable for many IoT applications, as it is often the case that devices accomplish a specific task as part of a larger group of devices to be attested. For instance in the Industrial Internet of Things, many cyberphysical collaborate to monitor and control safety-critical processes. Combining the result of individual attestation protocols for each system would result in an inefficient and hardly-scalable process. The concept of *swarm attestation* (SA) was developed to address this issue. *Device swarms* are groups of devices forming a network, and collaborating to perform a given task. In most applications, a control or base station is the verifier, verifying the integrity of the device swarms. The secure deployment of a swarm attestation protocol presents additional challenges:

- **Dynamic network topology:** devices and systems may be added or removed from the swarm, and the attestation process must adapt to the new network configuration;

- **Denial of service:** the verifier might be a target of denial of service attacks when a node in the network becomes compromised. Indeed, even though the attestation process is initiated by the verifier, a malicious attacker who takes over one device may constantly send back an erroneous attestation response, hence triggering the verifier into performing the attestation protocol in a loop.

State of the art

Introduced in 2015, SEDA [ABI+15] was the first swarm attestation protocol. The SEDA model is based on the strong (in the cryptographic sense) assumption that it is unfeasible for an adversary to tamper with the attestation mechanism, or to forge an integrity measurement report. In practice, an adversary might attempt to tamper with the attestation mechanism itself by obtaining rogue credentials. Carpent, El Defrawy, Rattanavipanon, and Tsudik [CDRT17] then introduced swarm attestation protocols that attempt to provide more practical alternatives to SEDA. Indeed, the authors first introduce their classification of swarm attestation protocol models, namely *Quality of Swarm Attestation (QoSA)*, prior to presenting two swarm attestation protocols with respect to said classification (LISA_α and LISA_s). SANA [ACI+16] is the first instantiation of a swarm attestation protocol based on public key cryptography. The underlying scheme of the swarm attestation protocol is an aggregate digital signature scheme that allows the efficient aggregation of attestations based on the multi-signature scheme by Boldyreva [Bol03] and the aggregate signature scheme proposed by Boneh et al. in [BGLS03]. SANA leverages the structure of aggregate and multisignature signature schemes, in order to obtain a highly scalable attestation protocol. The solution provides public verifiability, and limits physical attacks by authenticating reports within secure hardware. Kohnhauser et al. [KBGK17] proposed a swarm attestation scheme secure against physical attacks. Prior to their scheme, the only model secure against physical attacks was DARPA [ISTZ16]. DARPA however requires each device to send a heartbeat to other devices in the network at specific time intervals. Failing to do so results in a time-out that suggests that the device has been taken off the network in order to tamper with it and extract keys. Denial of service attacks are mitigated

in SeED [ISZ17] through the use of a secure clock, which enables provers to periodically forward their attestation reports in a non-interactive protocol. SALAD [KBK18] is a swarm attestation protocol suitable for dynamic and disruptive networks. The solution is based on a distributed aggregation approach, allowing all device to gradually compute the aggregate attestation response using reports from devices in their communication range.

3.4 Privacy in Trusted Computing

In the first deployments of hardware-based attestation, TPMs' public keys had to be certified by a privacy Certification Authority CA (see Figure 3.2). This induced some privacy concerns, as the privacy CA could collude with the verifier (see Section 3.4.1) in order to determine which platform generated a given attestation. In addition, given that the privacy CA must take part in every transaction, it induces scalability issues. In this section, we present a cryptographic protocol which enables decentralized and privacy-preserving attestation.

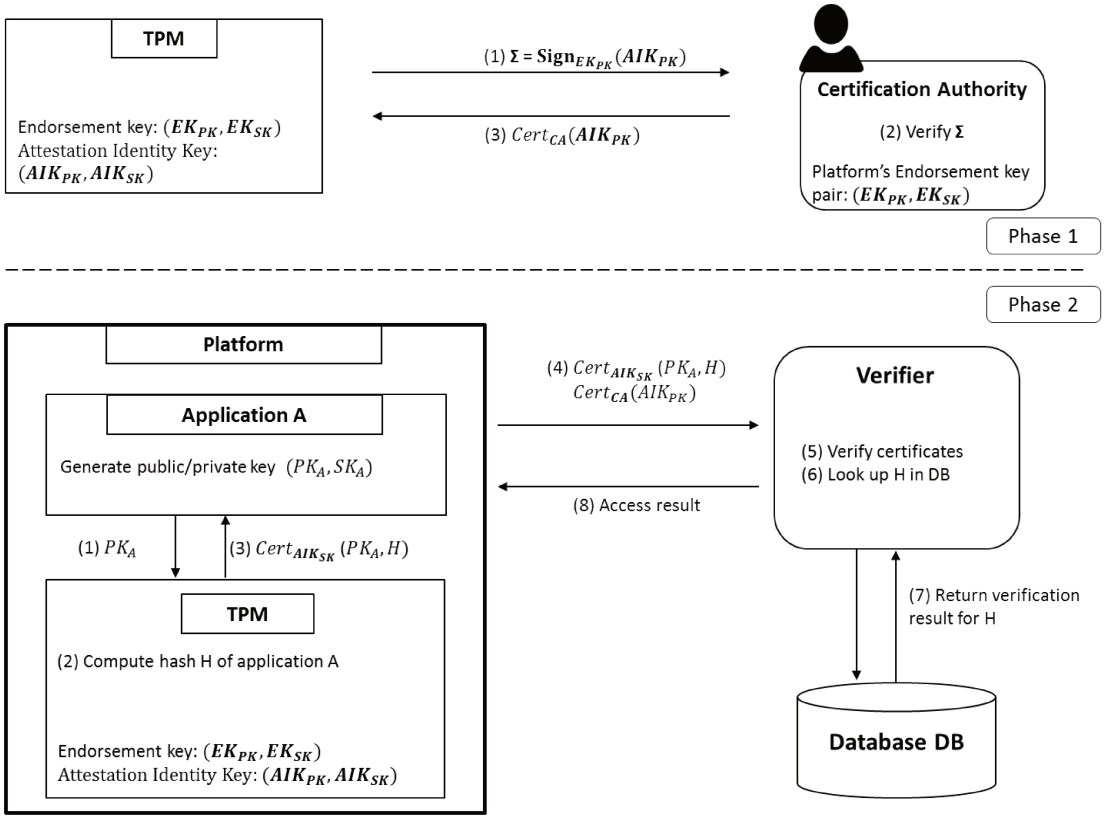


Figure 3.2: Attestation protocol with a privacy CA.

3.4.1 Direct Anonymous Attestation

Direct Anonymous Attestation (DAA) is an anonymous digital group signature scheme that provides privacy-preserving authentication for devices. A DAA signature allows a TPM (see Section 3.2.1) that is embedded in a host computer, to generate attestations about the state of the host system. Direct Anonymous Attestation (DAA) was introduced as a privacy-preserving

alternative to the traditional attestation process, which required that a privacy CA must authenticate each platform prior to the attestation generation process. The CA had thus the ability to identify which platform generated which attestation, and could trace transactions to a given computer. DAA signatures are not traceable, but signatures generated by the same platform can be linked using an auxiliary value called the *basename*. The first RSA-based scheme introduced by Brickell, Camenisch and Chen [BCC04] was standardized in the TPM 1.2 specification and the ISO 20008-2 norm [ISO13a]. Subsequently, more efficient DAA schemes based on elliptic curves and bilinear groups, namely ECC-DAA schemes, were then proposed [BCL08, BCL09, Che09, CPS10, BFG+13, CDL, CDL16, CDL17, KLL+18, WNT+19].
A DAA scheme comprises the following entities: an *issuer*, a *host*, a TPM, and a *verifier*. The host and the TPM together form a *trusted platform*. The issuer is a trusted third party responsible for attesting and authorizing platforms to join the network. A verifier is any other system entity or trusted third-party that can verify a platform's credentials in a privacy-preserving manner, without the need of knowing a platform's identity. The DAA scheme comprise two phases. The first phase is an interactive JOIN/ISSUE phase that runs between the platform and the issuer, and at the end of which the platform obtains its DAA credentials. The second phase is the SIGN/VERIFY phase that runs between the platform and the verifier. It is the attestation phase during which the platform uses its DAA group signing key to anonymously sign a message sent by the verifier. The verifier initiates the SIGN/VERIFY phase by sending a message m comprising a random *challenge* Ch and the basename bsn. Depending on the application, the basename is used to link to signatures when needed. At the end of the SIGN/VERIFY phase, the verifier is convinced that it is interacting with a platform in possession of a valid group key generated by the issuer.

DAA schemes are complex and designing a security model for the scheme has generated a large body of work over the years. In 2013, Bernhard et al. [BFG+13] have notably proved that in previous models, insecure schemes could be proven secure. Bernhard et al. therefore proposed a new game-based security model, which encapsulates the security notions that a DAA scheme should satisfy. They develop the model for a DAA variant called pre-DAA.

3.4.1.1 Pre-Direct Anonymous Attestation

Pre-DAA schemes simplify the model of traditional DAA schemes by considering the host and the TPM to be a single party in the system. The resulting model, introduced by Bernhard et al. [BFG+13], is a security model where the TPM performs all the DAA computation without delegation to the host. In a DAA scheme, a user is comprised of the Host and the TPM. In the pre-DAA model, we consider the Host and TPM to be a single entity. The user joins a group maintained by an issuer by executing a Join protocol, and said user may later authenticate itself anonymously with a verifier by executing a Sign protocol. The Join protocol is assumed to take place over an authenticated channel, as each user possesses a secret key but no public key that can be used to authenticate him as it is the case for group signatures. The result of Join is a credential associated with the user's secret key, that will be used later as the signing key. Once the user is part

of the group, he may sign messages on behalf of the group. The signatures should be untraceable in order to preserve the anonymity of the signer. The security of the previous instantiations of the pre-DAA model [BFG+13, BDGT17] is based on the interactive LRSW assumption (described in Chapter 2).

3.5 Conclusion

In this chapter, we describe the concept of attestation and its application to solving security challenges in various IoT applications. We showed that developing attestation protocols for low-end embedded devices requires designing efficient and highly scalable protocols. The mechanism of swarm attestation was introduced, to prove the integrity of device swarms in a secure and scalable manner. We provide an overview of attestation in trusted computing, and notably how it leverages a trusted execution environment to securely authenticate devices. Finally, we presented a privacy-preserving attestation protocol called Direct Anonymous Attestation (DAA), which is one of the most deployed cryptographic schemes today. A DAA scheme allows devices to generate anonymous attestations, and can be used to construct privacy-preserving authentication protocols. We notably presented a DAA variant called pre-DAA, which will use in Part II of this thesis to construct privacy-preserving authentication and physical access control protocols, that are efficient and suitable for resource-constrained environments.

Part II

Privacy-Preserving Authentication and Physical Access Control

CHAPTER 4

OUR PRE-DIRECT ANONYMOUS ATTESTATION SCHEME

In Part II of this thesis, we design secure and privacy-preserving protocols for message authentication and physical access control. The underlying cryptographic primitive in the design of our protocols is a pre-DAA scheme that is efficient and suitable for resource-constrained environments. In this chapter, we introduce our pre-DAA scheme, and we analyze its security in the random oracle model. We notably prove the security of our scheme based on the non-interactive and much preferred q-Strong Diffie-Hellman (q—SDH) assumption.

Contents

4.1 Two classes of Direct Anonymous Attestation Schemes

We distinguish two families of DAA constructions based on the underlying hardness assumption their security relies on. The first family comprises protocols based on the Lysyanskaya-Rivest-Sahai-Wolf (LRSW) assumption [LRSW99]. The LRSW assumption is an interactive assumption presented in Section 2.2.3.3. The complexity of the problem is conditional on the number of requests to $\mathcal{O}$. The security of members of the second family is based on a non-interactive q-type assumption, namely the q-Strong-Diffie-Hellman (q—SDH) (see Section 2.2.3.1). The security of cryptographic schemes relies upon the guarantees provided by the underlying hardness assumption. It is therefore a critical step to define assumptions that provide the best security/efficiency trade-off. Despite schemes based on LRSW assumption being relatively efficient (bounded by the number of queries to oracle $\mathcal{O}$), the complexity of interactive assumptions as mentioned earlier depends on the number of queries to the oracle which is only polynomially bounded by the adversary's power. Similarly, q-type assumptions have raised questions in the cryptographic community as the complexity is bounded by the value q, which is polynomially bounded by the adversary (it could be assumed to have relatively large constants making it sometimes as inefficient as an exponentially bounded q). There are a number of arguments against LRSW, the most important of which being that there is an effort to reduce q-type assumptions to standard assumptions that can lead to more robust security guarantees. A second one is the fact that interactive assumptions are "*non-falsifiable*".

4.1.1 Interactive vs non-interactive security assumptions

In this section, we provide a justification as to why it is preferable to prove the security of our cryptographic schemes based on non-interactive assumptions.

Concept of falsification.

Evaluating the robustness of a cryptographic assumption may present technical challenges. A straightforward method for such evaluation is simply assessing its falsifiability. Indeed, an assumption is said to be falsifiable if there is an efficient constructive way to prove that it is false. The first introduction to the classification of hardness assumptions based on their falsifiability level was introduced by Naor [Nao03]. More explicitly, an assumption is said to be falsifiable if it possesses the following properties:

- If a hardness assumption does not hold, there is a *constructive* way of proving so;
- The complexity of verifying that the assumption is false should be evaluated with respect to acceptable parameters, particularly it should be polynomial in the size of one instance of the problem.

Falsification by challenge. As with many constructions in cryptography, it is always more suitable to have publicly verifiable schemes for more robust security guarantees. Therefore in order to evaluate the falsification level of a given assumption, we introduce a public challenge. If the assumption is false, the challenge can efficiently be solved and the solution can be verified.The complexity of falsifying an assumption therefore relies on the complexity of generating a random challenge, and the complexity of verifying the solution proposed for said challenge. Ideally, the verification procedure will be public and run in constant time. We then introduce our *falsifier* algorithm in charge of solving the challenge generated by the party testing the falsification of the assumption. To conclude, in our evaluation protocol a protocol designer generates a public challenge of size n which is sent to the *falsifier*. The *falsifier* proposes a solution y to the challenge that is publicly made available. The verifier V in turn verifies the validity of y. If assumption A is false, then the falsifier is able to generate a solution in time polynomial, with respect to the security parameter. A cryptographic scheme is said to be $(n,t,\epsilon)-$secure based on assumption A for any adversary $\mathscr{A}$ if the probability of $\mathscr{A}$ breaking the assumption is less than or equal to ϵ. n is the instance size and t is the running time of adversary $\mathscr{A}$. We consider an additional parameter δ which is the upper bound on the probability of failure of the falsification algorithm in the case that A is indeed false. There are three categories of falsification defined as follows:

1. **Efficiently falsifiable.** An (n,t,ϵ) assumption is efficiently falsifiable if there is a distribution D_n on challenges and a verification procedure V such that sampling an input challenge from D_n can be done in polynomial time t. The verification algorithm should also run in polynomial time t. Additionally, if assumption A is false, then there exists a falsifier $\mathscr{B}$ such that for a challenge $d \in D_n$ outputs a solution y for which the verification algorithm outputs accept with probability at least $1-\delta$.

2. **Falsifiable.** An (n,t,ϵ) assumption is falsifiable if the running time of sampling D_n and V is polynomial in $1/\epsilon$.

3. **Somewhat falsifiable.** An (n,t,ϵ) assumption is somewhat falsifiable if the running time of V is polynomial in $1/\epsilon$ and the running time of $\mathscr{B}$ (V can simulate $\mathscr{B}$ and thus evaluate the probability of success of the adversary).

The appeal for q—SDH -based schemes stems from its property of being self-reducible. Indeed given a q—SDH problem instance, it is possible to generate another instance (or instances) whose solution would derives a solution for the original instance. In other words, if there exists an adversary $\mathscr{A}$ capable of solving the q—SDH problem for an instance, we can build a different algorithm $\mathscr{B}$ that uses $\mathscr{A}$ to generate a solution for the initial q—SDH instance. This is demonstrated in [BB04] by proving that a q—SDH -based signature is existentially unforgeable under weak chosen message attack based on the q-SDH assumption. The q—SDH assumption and all families of self-reducible assumptions are efficiently falsifiable. Conversely, LRSW and the family of interactive assumptions are not even somewhat falsifiable. Indeed, in case the LRSW assumption doesn't hold, it is not clearly defined how hard it is to generate a triplet for a new message m^* (i.e. it is not clear how a second reduction $\mathscr{B}$ can use adversary $\mathscr{A}$'s LRSW instance for message m^* to output another LRSW

triplet). To date, the q—SDH assumption provides better security guarantees for constructions based on signatures[1]. The goal is to ultimately have an efficient DAA scheme based on efficiently falsifiable assumption.

4.1.2 The construction of DAA schemes based on LRSW and q—SDH

DAA schemes based on the LRSW assumption are built using the Camenisch-Lysyanskaya (CL) signature scheme [CL04]. In 2017, Barki, Desmoulins, Gharout, and Traoré [BDGT17] introduced the first pre-DAA scheme constructed using the Pointcheval, Sanders (PS) signature scheme [PS16]. The PS scheme is a randomized version of the CL scheme, where signatures can be randomized by the signer. By using a randomized signature to sign the TPM's secret, DAA signatures generated from the resulting DAA group signing key become unlinkable.
DAA schemes based on the q—SDH assumption are built using the Boneh, Boyen, Shacham+ (BBS+) signature scheme [BBS04].

4.2 Pre-DAA Definition and Security Model

Bernhard et al. [BFG+13] provided the first formal definition and security model for a pre-DAA scheme, using the *game-based approach*, whereby each property is individually defined through an experiment (or game) between a challenger $\mathscr{C}$ and an adversary $\mathscr{A}$.
In this section, we present the formal definition of a pre-DAA scheme, as well as the security model as introduced in [BFG+13].

4.2.1 Definition

Let $\mathscr{T}$ be the TPM (group member) with identifier i, $\mathscr{I}$ be the issuer (group manager), and $\mathscr{V}$ be the verifier. In the remaining of this chapter, we will refer to the TPM T_i and the user U interchangeably to designate the signing platform (given that in a pre-DAA scheme, all computations are undertaken by the TPM). A pre-DAA scheme comprises the following algorithms:

- Setup(1^λ). A probabilistic algorithm that takes a security parameter 1^λ as input, and outputs the description of the public parameters pp;

- Keygen(pp). A probabilistic algorithm that takes the public parameters as input and outputs the issuer's public/private key pair ($gmpk$/$gmsk$). We assume from this point that $pp \subset gmpk$;

- UKeygen(pp, i). A probabilistic algorithm that on input the public parameters pp and a TPM's identifier i, outputs the TPM's private key sk_i;

[1] At the time we are writing this thesis, it appears that a paper will be published at the Crypto 2020 conference demonstrating that q—SDH and LRSW are equivalent to the q-Discrete Logarithm assumption in the algebraic group model.

- Join(St_U, M_I)-Issue(St_I, M_U). An interactive protocol between a TPM $\mathcal{T}_i$ and the issuer $\mathcal{I}$. $\mathcal{T}_i$ initiates the protocol with a Join execution. During each pass of the protocol, each algorithm takes as input a state St_x and a message M_x, where $x \in \{U, I\}$, and produces a new state, a new message and a decision $\in$ {accept, reject, cont}.The final state of Join is the group signing key gsk_i for $\mathcal{T}_i$, and Issue outputs accept or reject;

- Sign(sk_i, gsk_i, m, bsn). A probabilistic algorithm that takes as input a message m, a basename bsn and a group signing key gsk_i and returns a signature σ;

- Verify($gmpk, \sigma, m, bsn$). A deterministic algorithm that takes as input a signature σ, a message m and a basename bsn and returns 1 if σ is a valid signature on m with respect to bsn and 0 otherwise;

- Identify$_{\mathcal{T}}$($\mathcal{T}, sk_i$). A deterministic algorithm that takes as input a transcript $\mathcal{T}$ and a TPM secret key sk_i and returns 1 if $\mathcal{T}$ corresponds to the transcript generated from a Join/Issue protocol between the issuer and $\mathcal{T}_i$, and 0 otherwise;

- Identify$_{\mathcal{I}}$(σ, m, bsn, sk_i). A deterministic algorithm that outputs 1 if σ was produced on m with secret key sk_i with respect to bsn, and 0 otherwise;

- Link($gmpk, \sigma, m, \sigma', m', bsn$). A deterministic algorithm that outputs 1 if σ and σ' are two signatures generated on m, m' respectively by the same TPM, with respect to the basename bsn.

4.2.2 Security properties

In the security model for a pre-DAA as defined by Bernhard et al. [BFG+13], we consider a probabilistic polynomial time adversary $\mathcal{A}$ who attempts to break the security of the scheme. We distinguish to sets of users and store their identifiers in two different lists: a list $\mathcal{HU}$ storing the identifiers of honest users, and a list $\mathcal{CU}$ storing the identifiers of corrupted users (whose secret key sk_i and group signing key gsk_i are known to the adversary). In addition, we maintain a list $\mathbb{S}$ of queries to the signing oracle, a list $\mathbb{C}$ of queries to the challenge oracles, and a list $\mathbb{T}$ of transcripts resulting from executions of a Join-Issue protocol. Adversary $\mathcal{A}$ is able to corrupt users (therefore has complete access to their TPM's secret keys and group signing keys) via a set of oracles defined as follows:

- $\mathcal{O}_{Add(i)}$: $\mathcal{A}$ uses this oracle to create a new honest user with identifier i;

- $\mathcal{O}_{AddCorrupt_U(i)}$: $\mathcal{A}$ uses this oracle to create a new corrupt user with identifier i;

- $\mathcal{O}_{Init_U}$: $\mathcal{A}$ can use this oracle to create a group signing key for an honest user with identifier i;

- $\mathcal{O}_{Join_I(i)}$: $\mathcal{A}$ uses this oracle to execute the issuer's side of the Join-Issue protocol. This oracle will be used to simulate the execution of the Join protocol between an honest or corrupted user, and an *honest* issuer;

- $\mathcal{O}_{Join_U(i)}$: $\mathcal{A}$ uses this oracle to execute the user's side of the Join-Issue protocol. This oracle will be used by $\mathcal{A}$ acting as a malicious issuer. The adversary provides that oracle with an honest user's identifier i. If the latter accepts, $\mathcal{A}$ gets a transcript $\mathcal{T}$ of the protocol execution, which is then saved in $\mathbb{T}$;

- $\mathcal{O}_{Corrupt(i)}$: $\mathcal{A}$ uses this algorithm to corrupt user i. He obtains both the user's secret key sk_i and his group signing key gsk_i. User i is therefore moved from the list of honest users $\mathcal{HU}$ to the list of corrupted users $\mathcal{CU}$;

- $\mathcal{O}_{Sign(i,m,bsn)}$: $\mathcal{A}$ uses this oracle to obtain a signature on message m with respect to the basename bsn produced by user i. The triple (i,m,bsn) is saved in $\mathbb{S}$;

- $\mathcal{O}_{Ch_b(i_0,i_1,m,bsn)}$: $\mathcal{A}$ uses this oracle to obtain a signature σ on m generated by user i_b, where b is either equal to 0 or 1.

A secure pre-DAA scheme must satisfy four security properties, namely *correctness*, *anonymity*, *traceability*, and *non-frameability*. Each property is formally described as follows:

Correctness. This property ensures that the system works as expected. Notably, the following four conditions must be met: (1) the group signing key gsk_i is valid; (2) a valid signature is accepted by the verifier $\mathcal{V}$; (3) a valid signature can be traced back to the correct sk_i;and (4) two signatures produced by the same user (i.e. using the same sk_i) and with respect to the same basename bsn are linkable. The *correctness* experiment is depicted in Figure 4.1.

$\mathsf{Exp}^{\mathrm{corr}}_{\mathcal{A}}(1^\lambda)$:

1. $pp \leftarrow \mathsf{Setup}(1^\lambda)$;
2. $(gmpk, gmsk) \leftarrow \mathsf{Keygen(pp)}$.
3. $\mathcal{HU} \leftarrow \emptyset$.
4. $(i, m_0, m_1, bsn) \leftarrow \mathcal{A}^{\mathcal{O}_{Add}, \mathcal{O}_{Init_U}(gmpk)}$.
5. If $gsk_i = \perp$ then return 0.
6. $\sigma_0 \leftarrow \mathsf{Sign}(gsk_i, sk_i, m_0, bsn)$.
7. $\sigma_1 \leftarrow \mathsf{Sign}(gsk_i, sk_i, m_1, bsn)$.
8. If $\mathsf{Verify}(gmpk, \sigma_0, m_0, bsn) = 0$ then return 1.
9. If $\mathsf{Verify}(gmpk, \sigma_1, m_1, bsn) = 0$ then return 1.
10. If $bsn \neq \perp$, if $\mathsf{Link}(gmpk, \sigma_0, m_0, \sigma_1, m_1, bsn) = 0$, then return 1.
11. $\forall b \in \{0,1\}$, if $\mathsf{Identify}_{\mathcal{J}}(\sigma_b, m_b, bsn, sk_i) = 0$ then return 1.
12. Let $\mathcal{T}_i$ be the transcript from the Join-Issue protocol for user i;
13. If $\mathsf{Identify}_{\mathcal{T}}(\mathcal{T}_i, sk_i) = 0$ then return 1.
14. Return 0.

Figure 4.1: Correctness security experiment.

Adversary $\mathcal{A}$'s advantage in the correctness game is defined as $\mathsf{Adv}^{\mathrm{corr}}_{\mathcal{A}}(1^\lambda) = \mathsf{Pr}[\mathsf{Exp}^{\mathrm{corr}}_{\mathcal{A}}(1^\lambda) = 1]$. A pre-DAA scheme is *correct* if any probabilistic polynomial time adversary $\mathcal{A}$'s advantage is equal to 0.

Anonymity. The *anonymity* property stipulates that, given a signature σ and to identities i_0 and i_1, no polynomial time adversary should be able to determine if σ was generated by i_0 or i_0 with a probability significantly bigger than the probability of guessing. The *anonymity* experiment is depicted in Figure 4.2

$\text{Exp}^{anon-b}_{\mathscr{A}}(1^\lambda)$:

1. $pp \leftarrow \mathsf{Setup}(1^\lambda)$;
2. $(gmpk, gmsk) \leftarrow \mathsf{Keygen(pp)}$.
3. $\mathscr{HU} \leftarrow \emptyset, \mathscr{CU} \leftarrow \emptyset, \mathbb{S} \leftarrow \emptyset, \mathbb{C} \leftarrow \emptyset$.
4. $b' \leftarrow \mathscr{A}^{\mathscr{O}}(gmpk, gmsk)$ where $\mathscr{O} = \{\mathscr{O}_{Add}, \mathscr{O}_{AddCorrupt_U}, \mathscr{O}_{Corrupt}, \mathscr{O}_{Join_U}, \mathscr{O}_{Sign}, \mathscr{O}_{Ch_b}\}$
5. If $\exists i, m, bsn$ such that $bsn \neq \perp$ and $(i, bsn) \in \mathbb{C}$ and $(i, m, bsn) \in \mathbb{S}$ then abort the game.
6. Return b'.

Figure 4.2: Anonymity security experiment.

Adversary $\mathscr{A}$'s advantage in the anonymity game is defined as $\mathsf{Adv}^{\text{anon-b}}_{\mathscr{A}}(1^\lambda) = 2 \cdot |\mathsf{Pr}[\text{Exp}^{\text{anon-b}}_{\mathscr{A}}(1^\lambda) = 1] - 1|$. A pre-DAA scheme is *anonymous* if any probabilistic polynomial time adversary $\mathscr{A}$'s advantage is negligible.

Traceability. A pre-DAA scheme satisfies the *traceability* property if no two-time adversary $\mathscr{A} = (\mathscr{A}_1, \mathscr{A}_2)$ can produce the following elements: (1) a valid signature which cannot be traced back to a secret key that has been committed to during the execution of a Join-Issue protocol; (2) two signatures that were generated using the same secret key with respect to te same basename, but that are unlinkable. The *traceability* experiment is depicted in Figure 4.3

$\text{Exp}^{trace}_{\mathscr{A}}(1^\lambda)$:

1. $pp \leftarrow \mathsf{Setup}(1^\lambda)$;
2. $(gmpk, gmsk) \leftarrow \mathsf{Keygen(pp)}$.
3. $\mathscr{HU} \leftarrow \emptyset, \mathscr{CU} \leftarrow \emptyset, \mathbb{T} \leftarrow \emptyset$.
4. $(\sigma, m, bsn, sk_1, ..., sk_l) \leftarrow \mathscr{A}_1^{\mathscr{O}}(gmpk)$ where $\mathscr{O} = \{\mathscr{O}_{Add}, \mathscr{O}_{Corrupt}, \mathscr{O}_{Join_I}, \mathscr{O}_{AddCorrupt_U}, \mathscr{O}_{Sign}\}$
5. If the following conditions hold then return 1
   (a) $\mathsf{Verify}(gmpk, \sigma, m, bsn) = 1$;
   (b) $\forall T \in \mathbb{T}, \exists i \in [1, l] : \mathsf{Identify}_T(T, sk_i) = 1$;
   (c) $\forall i \in [1, l] : \mathsf{Identify}_{\mathscr{S}}(\sigma, m, bsn, sk_i) = 0$.
6. $(\sigma_0, m_0, \sigma_1, m_1, bsn, sk) \leftarrow \mathscr{A}_2(gmpk, gmsk)$.
7. If $bsn = \perp$ return 0.
8. If the following conditions hold then return 1
   (a) $\forall b \in \{0,1\}$, $\mathsf{Verify}(gmpk, \sigma_b, m, bsn) = 1$;
   (b) $\forall b \in \{0,1\}$ $\mathsf{Identify}_{\mathscr{S}}(\sigma_b, m_b, bsn, sk) = 1$;
   (c) $\mathsf{Link}(gmpk, \sigma_0, m_0, \sigma_1, m_1, bsn) = 0$.
9. Return 0.

Figure 4.3: Traceability security experiment.

Adversary $\mathscr{A}$'s advantage in the traceability game is defined as $\mathsf{Adv}^{\text{trace}}_{\mathscr{A}}(1^\lambda) = \mathsf{Pr}[\mathrm{Exp}^{\text{trace}}_{\mathscr{A}}(1^\lambda) = 1]$. A pre-DAA scheme satisfies the *traceability* property if any probabilistic polynomial time adversary $\mathscr{A}$'s advantage is negligible.

Non-frameability. A pre-DAA scheme satisfies the *non-frameability* property if no adversary can produce the following: (1) a signature that can be traced to a given user i who has not generated a signature on the corresponding message/basename pair; (2) two signatures that are linkable even though they should not following a set of conditions detailed in steps (8)-(9) of the *non-frameability* security experiment depicted in Figure 4.4.

$\mathrm{Exp}^{non-frame}_{\mathscr{A}}(1^\lambda)$:

1. $pp \leftarrow \mathsf{Setup}(1^\lambda)$;
2. $(gmpk, gmsk) \leftarrow \mathsf{Keygen}(\mathsf{pp})$.
3. $\mathscr{CU} \leftarrow \emptyset, \mathscr{HU} \leftarrow \emptyset, \mathbb{S} \leftarrow \emptyset$.
4. $(\sigma, i, m, bsn) \leftarrow \mathscr{A}_1^{\mathscr{O}}(gmpk)$ where $\mathscr{O} = \{\mathscr{O}_{Add}, \mathscr{O}_{Join_U}, \mathscr{O}_{AddCorrupt_U}, \mathscr{O}_{Corrupt}, \mathscr{O}_{Sign}\}$
5. If the following conditions hold then return 1
   (a) $\mathsf{Verify}(gmpk, \sigma, m, bsn) = 1$;
   (b) $i \in \mathscr{HU}$;
   (c) $(i, m, bsn) \notin \mathbb{S}$;
   (d) $\mathsf{Identify}_{\mathscr{S}}(\sigma, m, bsn, sk_i) = 1$.
6. $(\sigma_0, m_0, bsn_0, \sigma_1, m_1, bsn_1, sk) \leftarrow \mathscr{A}_2(gmpk, gmsk)$.
7. If $\exists b \in \{0,1\} : \mathsf{Verify}(gmpk, \sigma_b, m_b, bsn_b) = 0$ then return 0.
8. If $\forall b \in \{0,1\} : \mathsf{Link}(gmpk, \sigma_0, m_0, \sigma_1, m_1, bsn_b) = 0$ then return 0.
9. For $b \in \{0,1\}$, if $\mathsf{Identify}_{\mathscr{S}}(\sigma_b, m_b, bsn_b, sk) = 1$ and $\mathsf{Identify}_{\mathscr{S}}(\sigma_{1-b}, m_{1-b}, bsn_{1-b}, sk) = 0$, then return 1.
10. If $bsn_0 \neq bsn_1$ or $bsn_0 = \perp$ or $bsn_1 = \perp$, then return 1.
11. Return 0.

Figure 4.4: Non-frameability security experiment.

$\mathscr{A}$'s advantage in the non-frameability game is defined as $\mathsf{Adv}^{\text{non-frame}}_{\mathscr{A}}(1^\lambda) = \mathsf{Pr}[\mathrm{Exp}^{\text{non-frame}}_{\mathscr{A}}(1^\lambda) = 1]$. A pre-DAA scheme satisfies the *non-frameability* property if any probabilistic polynomial time adversary $\mathscr{A}$'s advantage is negligible.

4.3 Presentation of Our Scheme

In this Section, we introduce a new efficient and secure pre-DAA scheme. Our pre-DAA scheme is based on the Pointcheval, Sanders (PS) signature scheme presented in Section 2.5.2.1.

4.3.1 Setup

Let $\mathbb{G}_1, \mathbb{G}_2$, and $\mathbb{G}_T$ be three bilinear groups of prime order p. This algorithm selects $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ a type 3 bilinear map. It then randomly selects three generators $g, h \xleftarrow{\$} \mathbb{G}_1$ and $\widetilde{h} \xleftarrow{\$} \mathbb{G}_2$ in a "verifiable manner". It selects a hash function $\mathscr{H} : \{0,1\}^* \rightarrow \mathbb{G}_1$, modeled as a random oracle in the security analysis. It then outputs the public parameters $pp = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g, h, \widetilde{h}, \mathscr{H}, e)$.

4.3.2 Keygen

The issuer $\mathscr{I}$ selects $x_0, x_1 \xleftarrow{\$} \mathbb{Z}_p^*$, and sets his secret key $gmsk = (x_0, x_1)$. He also defines the corresponding public key $gmpk = (C_{x_0} = g^{x_0} h^{\widetilde{x}_0}, X_1 = h^{x_1}, \widetilde{X}_0 = \widetilde{h}^{x_0}, \widetilde{X}_1 = \widetilde{h}^{x_1})$ where $\widetilde{x}_0 \xleftarrow{\$} \mathbb{Z}_p^*$. He generates a zero-knowledge signature of knowledge π on his private key (x_0, x_1), defined as $\pi = SoK\{\alpha, \beta, \gamma : C_{x_0} = g^{\alpha} h^{\beta}, X_1 = h^{\gamma}, \widetilde{X}_0 = \widetilde{h}^{\alpha}, \widetilde{X}_1 = \widetilde{h}^{\gamma}\}[m_0]$ where m_0 is the empty string. The proof of knowledge on $\mathscr{I}$'s secret keys prevents impersonation attacks. $\mathscr{I}$ generates an additional ECDSA signature [Alg09] on C_{x_0}, X_1, π, which is verified by the TPM i as soon as $gmpk$ is deployed. i is deployed with an endorsement key pair (esk_i / epk_i) of an EUF-CMA signature scheme $Sign$, which is used in the authentication step with the issuer. Finally, i selects a private key $s_1 \xleftarrow{\$} \mathbb{Z}_p^*$ as its secret key.

4.3.3 Join-Issue

This interactive protocol runs between the TPM with unique identifier i and the issuer $\mathscr{I}$. During the Join execution, i first computes a hiding commitment of its secret key s_1 as follows: $C_{s_1} = X_1^{s_1}$. It then builds a zero-knowledge signature of knowledge π of s_1 defined as follows: $\pi_1 = SoK\{\alpha : C_{s_1} = X_1^{\alpha}\}[m_0]$. It generates an EUF-CMA-secure signature $Sign$ with public/private key (epk_i, esk_i) defined as $\mathscr{S}_C = Sign_{esk_i}(C_{s_1})$ on the commitment, and sends $(C_{s_1}, \pi_1, \mathscr{S}_C)$ to the issuer. Upon receiving $(C_{s_1}, \pi_1, \mathscr{S}_C)$, $\mathscr{I}$ checks that $C_{s_1} \neq 1$, checks the validity of $\mathscr{S}_C$, and the validity of π_1. He then selects $b, s_2 \xleftarrow{\$} \mathbb{Z}_p^*$, and generates i's group signing key (u, u') as follows: $u = h^b, u' = u^{x_0} [C_{s_1} X_1^{s_2}]^b = u^{x_0 + x_1(s_1 + s_2)}$. $\mathscr{I}$ builds a zero-knowledge signature of knowledge π_2 of $b, x_0, \widetilde{x}_0$ defined as follows: $\pi_2 = SoK\{\alpha, \beta, \gamma, \mu : u = h^{\alpha} \wedge u' = u^{\beta} [C_{s_1} \cdot C_{s_2}]^{\alpha} \wedge C_{x_0} = g^{\beta} h^{\gamma} \wedge C_{s_2} = X_1^{\mu}\}[m_0]$, where $C_{s_2} = X_1^{s_2}$. It sends i its group signing key (u, u'), as well as C_{s_2} and π_2. Upon receiving $((u, u'), C_{s_2}, \pi_2)$, i checks the validity of π_2. If $C_{s_1} \cdot C_{s_2} = 1$, it aborts. Otherwise, it computes a signature $\sigma_0 = Sign_{esk_i}(C_{s_1}, C_{s_2}, u, u', \pi_2)$ and sends σ_0 to the issuer. The issuer verifies σ_0 and finally sends back s_2 if the verification was successful. The TPM checks that $C_{s_2} = X_1^{s_2}$, and sets $sk_i = s_1 + s_2$ (mod p) and $gsk = (u, u')$. The issuer stores the values (i, C_{sk_i}, σ_0) in a register REG, where $C_{sk_i} = C_{s_1} \cdot C_{s_2}$.

4.3.4 Sign

Upon receiving the challenge Ch and the basename bsn from the verifier, i selects $l \xleftarrow{\$} \mathbb{Z}_p^*$ and computes a randomized version (w, w') of the credentials (u, u') where $w = u^l$ and $w' = (u')^l$. It then computes $c = w^{sk_i}$. Finally, it computes a tag $T = \mathscr{H}(bsn)^{sk_i}$ on the basename, and build

a zero-knowledge signature of knowledge $\pi_3 = SoK\{\alpha : c = w^\alpha \wedge T = \mathscr{H}(bsn)^\alpha\}[Ch]$ of a valid PS signature (w, w'), generated on the secret key sk_i. It defines the signature $\sigma = (w, w', \pi_3, c, T)$.

4.3.5 Verify

Upon receiving σ, the verifier $\mathscr{V}$ first checks that $w \neq 1$ and $T \neq 1$. It then verifies the following equality: $e(w, \widetilde{X}_0) \cdot e(c, \widetilde{X}_1) = e(w', \widetilde{h})$. $\mathscr{V}$ accepts if π_3 is valid and all the previous checks succeed by returning 1. This last verification step completes the verification of i's credentials and its signature.

4.3.6 Identify$_{\mathscr{S}}$

Return 1 if $T = \mathscr{H}(bsn)^{sk_i}$ and 0 otherwise.

4.3.7 Identify$_{\mathscr{T}}$

Return 1 if $C_{s_1} \cdot C_{s_2} = X_1^{sk_i}$, where C_{s_1} and C_{s_2} are the two commitments produced during the *Join* protocol associated with the transcript $\mathscr{T}$, and 0 otherwise.

4.3.8 Link

If both σ and σ' verify for m, and m' respectively using the same basename $bsn \neq \perp$, and both σ and σ' were produced by the same user (i.e. the tags $T = T'$), return 1. Otherwise return 0.

4.4 Security Proof

In this section, we prove that our pre-DAA scheme satisfies the security properties defined in Section 4.2.2, using Shoup's game hopping technique [Sho04], in the random oracle model. The proof of *correctness* follows by inspection. Indeed, a signature generated using a valid gsk_i is accepted, and two signatures generated using the same gsk_i can be linked using the Link function described in Section 4.3.

4.4.1 Anonymity

We prove that our pre-DAA scheme satisfies the anonymity property under the XDH assumption. We show that an adversary with non-negligible advantage in the anonymity game can be used to break the XDH assumption. For $b = 0$ and $b = 1$, we will define a sequence of games where Game 0 is $\text{Exp}_{\mathscr{A}}^{anon-b}(\lambda)$ where the adversary $\mathscr{A}$ tries to correctly guess the challenge user i_b.

Game 0. The challenger $\mathscr{C}$ randomly selects the public parameters $pp = (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g, h, \tilde{h}, \mathscr{H}, e)$, and picks two random values $x_0, x_1 \in \mathbb{Z}_p^*$ as the issuer's private key. It then computes the associates public key $gmpk = (C_{x_0} = g^{x_0} h^{\tilde{x}_0}, X_1 = h^{x_1}, \tilde{X}_0 = \tilde{h}^{x_0}, \tilde{X}_1 = \tilde{h}^{x_1})$. It then sends $\mathscr{A}$ the corresponding private key $gmsk = (x_0, x_1)$, and answers $\mathscr{A}$'s queries as follows:

- $\mathcal{O}_{Add}(i)$: $\mathscr{C}$ creates a new user i, picks a random value $s_1^i \xleftarrow{\$} \mathbb{Z}_p^*$, and computes $C_{s_1^i} = X_1^{s_1^i}$. $\mathscr{C}$ also generates i's endorsement key pair (esk_i, epk_i).
- $\mathcal{O}_{AddCorrupt_U}(i)$: $\mathscr{C}$ does nothing.
- $\mathcal{O}_{Join_U}(i)$: $\mathscr{C}$ computes the commitment $C_{s_1^i}$ as well as the proof π_1. If the protocol does not abort, $\mathscr{C}$ obtains the group signing key $gsk_i = (u, u')$ associated with the secret key $sk_i = s_1^j + s_2^j$, where s_2^j was chosen by $\mathscr{A}$ and transmitted to $\mathscr{C}$ along with a valid proof π_2.
- $\mathcal{O}_{Corrupt}(i)$: $\mathscr{C}$ provides $\mathscr{A}$ with the secret key sk_i and the group signing key gsk_i of TPM i.
- $\mathcal{O}_{Sign}(i, m, bsn)$: $\mathscr{C}$ uses sk_i and gsk_i to generate a signature σ on m with respect to bsn, and sends σ to $\mathscr{A}$.

Eventually, $\mathscr{A}$ queries $\mathcal{O}_{Ch_b}$ with the input (i_0, i_1, m, bsn). The oracle outputs a signature σ_b produced by i_b. $\mathscr{A}$'s goal is to guess the value of b. Upon receiving the challenge, $\mathscr{A}$ may still query the $\mathcal{O}_{Add}$, $\mathcal{O}_{AddCorrupt_U}$, $\mathcal{O}_{Join_U}$, $\mathcal{O}_{Corrupt}$ and $\mathcal{O}_{Sign}$, but with some restrictions. $\mathscr{A}$ cannot query $\mathcal{O}_{Corrupt}$ with one of the identities provided to the challenge oracle, nor is he allowed to query $\mathcal{O}_{Sign}$ with either (i_0, bsn) or (i_1, bsn) otherwise he could use the Link algorithm to trivially win the game. Eventually, $\mathscr{A}$ outputs its guess b'. Let S_0 define the event that $b = b'$ in Game 0 (i.e. the event that $\mathscr{A}$ wins Game 0), and S_i the event defining $b = b'$ in Game i. We have $Adv_{\mathscr{A}}^{anon-b}(1^\lambda) = |Pr[\mathsf{Exp}_{\mathscr{A}}^{anon-b}(\lambda) = b] - \frac{1}{2}| = |Pr[S_0] - \frac{1}{2}|$. In the next game, we slightly modify the signature output by the challenger $\mathscr{C}$ which operates $\mathcal{O}_{Ch_b}$, so that $\mathscr{A}$ can detect a change only with negligible probability.

Game 1. This is the same game as Game 0, except that the signature output by $\mathcal{O}_{Ch_b}$ is generated differently.
$\mathscr{C}$ picks a random secret $\hat{s} \xleftarrow{\$} \mathbb{Z}_p^*$ that he will use to compute the signature σ. It chooses a random element $w \xleftarrow{\$} \mathbb{G}_1$, then computes $c = w^{\hat{s}}, w' = w^{x_0 + \hat{s} \cdot x_1}$ (recall that he knows x_0 and x_1), and computes $T = \mathscr{H}(bsn)^{\hat{s}}$. $\mathscr{C}$ simulates the proof π_3 in the Random Oracle Model (ROM) using standard techniques. $\mathscr{C}$ returns $\sigma = (w, w', c, T)$ along with π_3 as the signature on (m, bsn) produced by i_b. Under the XDH assumption, $\mathscr{A}$ cannot detect this change (i.e. that the discrete logarithm of T in base $\mathscr{H}(bsn)$ is not equal to the discrete logarithm of $C_{s_1^{i_b}} \cdot C_{s_2^{i_b}}$ in base X_1). Indeed, one can easily construct an XDH distinguisher D_1 with advantage satisfying $|Pr[S_0] - Pr[S_1]| \leq Adv_{D_1}^{XDH}(1^\lambda)$. Note that in Game 1, $\mathscr{C}$ reveals no information to $\mathscr{A}$ about the bit b since the signature was generated using a random key $\hat{s}$, different from s_{i_0} and s_{i_1}, and the tag $T = \mathscr{H}(bsn)^{\hat{s}}$ was computed using $\hat{s}$. Therefore, $Pr[S_1] = \frac{1}{2}$, and we have:

$$
\begin{aligned}
Adv_{\mathscr{A}}^{anon-b}(1^\lambda) &= |Pr[\mathsf{Exp}_{\mathscr{A}}^{anon-b}(\lambda) = b] - \frac{1}{2}| \\
&= |Pr[S_0] - \frac{1}{2}| \\
&= |Pr[S_0] - Pr[S_1]| \\
&\leq Adv_{XDH}^{D_1}(1^\lambda)
\end{aligned}
\tag{4.1}
$$

$Adv_{D_1}^{XDH}(1^\lambda)$ is negligible under the XDH assumption, therefore $Adv_{\mathscr{A}}^{anon-b}(1^\lambda)$ is also negligible. Our pre-DAA scheme thus satisfies the anonymity property under the XDH assumption.

4.4.2 Traceability

Let $\mathscr{A}$ be an adversary who breaks the traceability property of our pre-DAA scheme with non-negligible probability. We distinguish two ways an adversary can break this property:

- Type-1 forger: An adversary that manages to output a valid signature σ that cannot be traced back to a secret key that was previously queried to $\mathscr{O}_{Join_I}$.
- Type-2 forger: An adversary that outputs two valid signatures σ_0 and σ_1 generated by the same TPM i using her secret key sk_i, and for the same basename bsn, and yet are unlinkable.

We show that a Type-1 forger can be used as a subroutine to build a forger $\mathscr{B}$ against the basic Pointcheval-Sanders signature scheme [PS18] under a weak chosen message attack (wCMA), whereas Type-2 forgery cannot happen. Initially, $\mathscr{B}$ chooses a random bit $c_{mode} \in \{1,2\}$ that indicates which type of forgery it guesses $\mathscr{A}$ will attempt.

If $c_{mode} = 1$: Forger $\mathscr{B}$ starts by requesting to its challenger $\mathscr{C}$ (for the EUF-wCMA security of the PS signature) signatures on random messages $m_1,...,m_q \in \mathbb{Z}_p^*$. $\mathscr{C}$ replies by first generating the public parameters $pp = (p,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,e,h,\tilde{h})$ of the PS signature [2] scheme, as well as $X_1 = h^{x_1}, \tilde{X}_0 = \tilde{h}^{x_0}$, and $\tilde{X}_1 = \tilde{h}^{x_1}$. $\mathscr{B}$ chooses the generator g and the value C_{x_0} at random, and simulates the proof π in the ROM using the simulation technique of standard proofs of knowledge of discrete logarithms. Thereby $\mathscr{B}$ can provide $\mathscr{A}$ with the public parameters of our pre-DAA scheme and answers oracle queries as follows:

- $\mathscr{O}_{Add}(i)$: $\mathscr{B}$ creates a new TPM i, and defines its secret key sk_i as m_i. $\mathscr{B}$ also generates TPM i's endorsement key pair (esk_i, epk_i).
- $\mathscr{O}_{AddCorrupt_U}(i)$: $\mathscr{B}$ stores the endorsement key pair (esk_i, epk_i) of this corrupted TPM.
- $\mathscr{O}_{Join_I}(i)$: Upon receiving $C_{s_1^i}$ and the proof π_1, $\mathscr{B}$ checks the validity of π_1. It then uses the soundness property of π_1 to extract the secret s_1. $\mathscr{B}$ then chooses a value $s_2 \in \mathbb{Z}_p^*$ such that $s_1 + s_2$ maps to m_i, and queries $\mathscr{C}$ to obtain a valid PS signature on m_i. $\mathscr{B}$ therefore obtains from $\mathscr{C}$ a pair $(u, u' = u^{x_0 + m_i \cdot x_1} = u^{x_0 + (s_1+s_2)x_1})$. $\mathscr{B}$ then simulates proof π_2 and answers $\mathscr{A}$'s queries by providing him with the commitment $C_{s_2} = X_1^{s_2}$, the pair (u,u'), as well as the proof π_2. If $\mathscr{A}$ sends $\mathscr{B}$ a valid signature S on $(u,u',\pi_2,C_{s_1},C_{s_2})$, then $\mathscr{B}$ sends back s_2. From $\mathscr{A}$'s viewpoint, the simulation provided by $\mathscr{B}$ is indistinguishable from a real attack scenario.
- $\mathscr{O}_{Corrupt}(i)$: $\mathscr{B}$ provides $\mathscr{A}$ with the secret key sk_i and the group signing key gsk_i of user i.

[2] We use in fact a variant of their scheme which requires to add a tuple $(g, Y = g^y)$, for a random generator $g \in \mathbb{G}_1$, in the public key pk. Pointcheval and Sanders showed that this modified version also achieves EUF-wCMA under the q—MSDH assumption (see [PS18] Remark 13).

- $\mathcal{O}_{Sign}(i, m, bsn)$: As he holds both sk_i and gsk_i, $\mathcal{B}$ can generate a signature σ on m with respect to bsn, and sends σ to $\mathcal{A}$.

Eventually, $\mathcal{A}$ outputs with non-negligible probability a valid tuple (σ, m, bsn), such that the signature σ was produced using an sk_i that is not associated with any of the calls to $\mathcal{O}_{Join_I}$. Using the soundness property of π_3, $\mathcal{B}$ extracts the secret $\hat{s}$ from the commitment c of the signature σ. Also, since σ is a valid pre-DAA signature on (m, bsn), (w, w') is a valid PS signature on $\hat{s}$ (i.e. $w \neq 1$ and $w' = w^{x_0 + \hat{s} \cdot x_1}$). Since $\hat{s}$ is not associated with any of the calls to O_{Join_I}, $\hat{s} \notin \{m_1, ..., m_q\}$, the triple $(w, w', \hat{s})$ is therefore a valid forgery of the basic PS signature scheme. Consequently using $\mathcal{A}$, $\mathcal{B}$ can break the Pointcheval-Sanders signature scheme under weak chosen message attack, hence breaking the q—MSDH-1 assumption ([PS18], Theorem 10).

If $c_{mode} = 2$: We prove that such an adversary $\mathcal{A}_2$ cannot exist. $\mathcal{A}_2$ outputs $(\sigma_0, m_0, \sigma_1, m_1, bsn, sk_i)$ such that σ_1 and σ_2 are valid signatures on respectively m_0, m_1 with respect to the same bsn. By definition in the traceability experiment, both signatures were also produced using the same key sk_i (since both $\mathsf{Identify}_{\mathcal{S}}(gmpk, \sigma_0, m_0, bsn, sk_i)$ and $\mathsf{Identify}_{\mathcal{S}}(gmpk, \sigma_1, m_1, bsn, sk_i)$ output 1). By completeness of the proof π_3, the tags T_0 and T_1 associated with σ_0 and σ_1 respectively are necessarily defined as $T_0 = \mathcal{H}(bsn)^{sk_i}$ and $T_1 = \mathcal{H}(bsn)^{sk_i}$. Therefore we have $T_0 = T_1$. By definition, $\mathcal{A}$ wins if the output of $\mathsf{Link}(gmpk, \sigma_0, m_0, \sigma_1, m_1, bsn)$ is 0 (i.e. $T_0 \neq T_1$). Such a forger therefore cannot exist.
In conclusion, if $\mathcal{A}$ can break the traceability property of our pre-DAA scheme, then $\mathcal{B}$ can break the q—MSDH-1 assumption with the same probability. Therefore under the non-interactive q—MSDH-1 assumption, our pre-DAA scheme satisfies the traceability property in the ROM.

4.4.3 Non-frameability

Let $\mathcal{A}$ be an adversary against the non-frameability property of our pre-DAA scheme. We will distinguish two types of forgers:

- Type-1 forger: An adversary that manages to output a signature which can be traced back to a user i for a message/basename pair that i has never signed.
- Type-2 forger: An adversary that outputs two signatures that are linkable even though they should not (i.e. they were either produced using two different secret keys, or with respect to two different basenames).

In the following, we will show that a Type-1 forger can be used to construct a reduction $\mathcal{B}$ against the One-More Discrete Logarithm (OMDL) assumption[3], while a Type-2 forger cannot occur.

If $c_{mode} = 1$: We observe that a corruption of the public register REG would imply that $\mathcal{A}$ has successfully produced a forgery for signature S. Given that $Sign$ is EUF-CMA, we consider that

[3]The OMDL assumption is used to simplify the proof and to get a tighter reduction. We would however like to emphasize that the scheme can also be proven to satisfy the *non-frameability* property under the discrete logarithm assumption.

such event will not occur. Therefore our reduction R will only be described using $\mathscr{A}$ against the OMDL challenge. $\mathscr{B}$ receives an input from the OMDL challenger $\mathscr{C}$ which is a random instance $(h^{u_1},...,h^{u_r})$, where h is a random generator of $\mathbb{G}_1$. The adversary $\mathscr{A}$ picks two random values $x_0, x_1 \xleftarrow{\$} \mathbb{Z}_p^*$ as the issuer's private key, and publishes the corresponding public key $gmpk = (C_{x_0} = g^{x_0}h^{\tilde{x}_0}, X_1 = h^{x_1}, \tilde{X}_0 = \tilde{h}^{x_0}, \tilde{X}_1 = \tilde{h}^{x_1})$. $\mathscr{B}$ uses the soundness property of π to recover $x_0, \tilde{x}_0$ and x_1, and answers $\mathscr{A}$'s oracle queries as follows:

- $\mathscr{O}_{Add}(i)$: $\mathscr{B}$ creates a new user i (using the input of the OMDL challenge and its knowledge of x_1), and sets $C_{s_1^i} = (h^{u_i})^{x_1} = X_1^{u_i}$ and $s_1^i = u_i$. Obviously s_1^i is unknown to both $\mathscr{B}$ and $\mathscr{A}$. $\mathscr{B}$ also generates u_i's public/private endorsement key pair (esk_i, epk_i).

- $\mathscr{O}_{AddCorrupt_U}(i)$: $\mathscr{B}$ does nothing.

- $\mathscr{O}_{Join_U}(i, M)$: $\mathscr{B}$ computes $C_{s_1^i} = (h^{u_i})^{x_1}$ and simulates the proof π_1 in the ROM. Since $\mathscr{B}$ holds esk_i, it can compute the signature S. If the protocol does not abort $\mathscr{B}$ will obtain a valid group signing key $gsk_i = (u, u')$ associated with $sk_i = u_i + s_2^i$, where s_2^i was chosen by $\mathscr{A}$ and transmitted to $\mathscr{B}$ along with a valid proof π_2. Using the soundness property of π_2, $\mathscr{B}$ retrieves the value b that it will use to simulate $\mathscr{O}_{Sign}$.

- $\mathscr{O}_{Corrupt}(i)$: $\mathscr{B}$ calls on the Discrete Logarithm oracle $\mathscr{O}_2$ of the OMDL challenge with h^{u_i} as input. Thereby, it obtains u_i from $\mathscr{O}_2$ and is able to compute $sk_i = u_i + s_2^i$ (where s_2^i has been transmitted by $\mathscr{A}$ to $\mathscr{B}$). It can therefore provide $\mathscr{A}$ with the secret key sk_i along with the group signing key $gsk_i = (u, u')$ of user U_i.

- $\mathscr{O}_{Sign}(i, m, bsn)$: As he holds both sk_i, gsk_i and b, $\mathscr{B}$ proceeds as follows to generate a valid signature (w, w', c, T, π_3) on message/basename pair (m, bsn): it randomly selects $r, l \xleftarrow{\$} \mathbb{Z}_p^*$ and sets $w = u^l, w' = (u')^l, c = (h^{u_i})^{b \cdot l} \cdot w^{s_2^i} = ((h^b)^l)^{u_i} \cdot w^{s_2^i} = w^{u_i} \cdot w^{s_2^i} = w^{sk_i}, H = \mathscr{H}(bsn) = h^r$, and $T = H^{s_2^i} \cdot h^{u_i r} = H^{s_2^i + u_i}$ (which is possible in the ROM). As for π_3, it can be easily simulated in the ROM. Hence $\mathscr{B}$ can perfectly simulate the $\mathscr{O}_{Sign}$ oracle in the ROM.

Eventually, after d calls to the $O_{Corrupt}$ oracle, $\mathscr{A}$ outputs with non-negligible probability a valid signature σ on message/basename pair (m, bsn) such that $\mathsf{Identify}_{\mathscr{S}}(\sigma, m, bsn, sk_i)$ outputs 1, whereas the user holding sk_i has never produced such signature. By definition, we know that the corresponding user is honest. Therefore the value u_i associated with the user's unknown secret sk_i was never queried to the Discrete Logarithm oracle $\mathscr{O}_2$. Using the Forking Lemma [PS00] and the soundness property of π_3, $\mathscr{B}$ retrieves the secret key sk_i associated with the signature σ, and therefore $u_i = sk_i - s_2^i$ (mod p), the discrete logarithm of the challenge h^{u_i} in the base h. By outputting u_i along with the secrets $\{u_j\}_{j=1}^d$ that it has obtained by querying the DL oracle $\mathscr{O}_2$, $\mathscr{B}$ therefore breaks the OMDL assumption.

If $c_{mode} = 2$: $\mathscr{A}$ eventually outputs two valid signatures σ_0 and σ_1 on m_0 and m_1 respectively, which are linkable although they should not (which means that they were either generated with respect to two different basenames bsn_0 and bsn_1, or using different secret keys s_0 and s_1). Let s_b denote the key used to generate σ_b and s the key output by $\mathscr{A}$ at the end of the experiment.

Given that $\mathscr{A}$ is a successful Type-2 forger, this means that condition 7 of the non-frameability experiment is true. In particular, the completeness of proof π_3 implies that we have:

$$T_0 = \mathscr{H}(bsn_0)^{s_0} \text{ and } T_1 = \mathscr{H}(bsn_1)^{s_1} \tag{4.2}$$

The condition 8 should also be true. Thus $\exists b \in \{0,1\}$ such that
$\mathsf{Link}(gmpk, \sigma_0, m_0, \sigma_1, m_1, bsn_b) = 1$. Without loss of generality, we assume that $b = 0$. Therefore condition 8 implies that:

$$T_0 = \mathscr{H}(bsn_0)^{s_0} = T_1 = \mathscr{H}(bsn_0)^{s_1} \tag{4.3}$$

As $T_0 \neq 1$ and $T_1 \neq 1$ (otherwise the signatures σ_0 and σ_1 would have been invalid), this implies that

$$s_0 = s_1 \text{ (mod p)} \tag{4.4}$$

For $\mathscr{A}$'s forgery to be successful, condition 9 or 10 should be true. Let us suppose condition 9 to be true (for $b = 0$, without loss of generality). This implies that:

$$T_0 = \mathscr{H}(bsn_0)^{s} \tag{4.5}$$

$$T_1 \neq \mathscr{H}(bsn_1)^{s} \tag{4.6}$$

This is impossible. Indeed, equations (4.2) and (4.5) imply that $s_0 = s$ and (4.3) implies that $s_0 = s_1 = s$ (mod p). From (4.2), we know that $T_1 = \mathscr{H}(bsn_1)^{s_1}$ which contradicts (4.6). Let us now assume that condition 10 is true, i.e. that $bsn_0 \neq bsn_1$. Equations (4.2) and (4.3) imply that $T_1 = \mathscr{H}(bsn_1)^{s_1} = \mathscr{H}(bsn_0)^{s_1}$. Since $T_1 \neq 1$ (otherwise σ_1 would have been invalid), the last equality implies that $\mathscr{H}(bsn_0) = \mathscr{H}(bsn_1)$ where $bsn_0 \neq bsn_1$. This would imply that $\mathscr{A}$ breaks the second pre-image resistance property of the hash function $\mathscr{H}$, which is infeasible in the ROM. Therefore Type-2 forgery can never occur. In conclusion, under the OMDL assumption and the second pre-image resistance of the hash function $\mathscr{H}$, our pre-DAA scheme satisfies the non-frameability property.

4.5 Conclusion

In this chapter, we introduce a novel pre-DAA scheme based on the Pointcheval-Sanders signature scheme. We prove the security of our scheme under a variant of the q—SDH assumption, namely the q—MSDH assumption introduced by Pointcheval and Sanders in [PS18]. Our pre-DAA scheme is suitable for resource-constrained environments such as SIM cards, as we will demonstrate in the remaining chapters in Part II. We will leverage the strong security and privacy properties of our pre-DAA scheme to design efficient and privacy-preserving physical access control and authentication protocols for two specific use cases. In Chapter 5, our pre-DAA scheme is used to construct a decentralized and privacy preserving pseudonym scheme, for message authentication in vehicular ad hoc networks. In Chapter 6, our pre-DAA scheme is used to construct an efficient and privacy-preserving mobile transit pass service, which allows commuters to anonymously authenticate on the public transport network without being traced.

CHAPTER 5

A Practical and Privacy-Preserving Pseudonym Scheme for V2X Communications

In this chapter, we design a new decentralized pseudonym scheme, which allows vehicles to autonomously generate and update their own pseudonyms in a secure and privacy-preserving manner. This is achieved by designing a pseudonym scheme based on our pre-DAA scheme introduced in Chapter 4. All secure computations in the pre-DAA pseudonym lifecycle are executed by the secure element, thus creating a secure enclave for pseudonym generation, update, and revocation. In addition, the pre-DAA-based construction transfers accountability from the vehicle to the user, thus complying with the many-to-many driver/vehicle relation. A test-bed implementation on a standard Java card shows that messages can be anonymously signed and verified in less than 50 milliseconds, which complies with the delay constraints of Vehicle-to-Everything (V2X) communications.

The results of this chapter have been published in [DDR+19].

Contents

5.1 Introduction

The new ecosystem in the automotive industry is characterized by vehicles communicating between themselves and with roadside infrastructures. Indeed, as one of the key manifestations of the Internet of Things revolution, vehicle-to-everything (V2X) communication will impact society by improving human safety and experience on the road. V2X technologies will enable the deployment of Intelligent Transportation Systems (ITS), characterized by connected, and semi-autonomous vehicles, and enabling the smart management of traffic information, collision detection and prevention, and the real-time regulation of traffic. V2X communication comprises different data communication channels, namely Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-Device (V2D), Vehicle-to-Grid (V2G), and vehicle-to-Pedestrian (V2P). The communication system enables vehicles to form a Vehicular Ad-Hoc Network (VANET), which shares the same challenges as other wireless sensor networks forming the Internet of Things.

5.1.1 Use cases

The widespread deployment of VANETs will positively impact a number of safety-related issues. A secure V2X communication system contributes to improving road safety and avoiding collisions, notably by allowing each vehicle to broadcast its position, speed, and direction to the other entities in the network. The communication network can also be used to provide vehicles with more visibility in traffic, notably by notifying them of queues, road works, or road hazards. The latter can especially be useful whenever visibility is reduced by meteorological conditions for example. In addition, V2X can increase capabilities of autonomous driving, by way of sensors and a secure communication system. Cooperative driving is also enabled, allowing vehicles to work together to minimize disruption caused by lane switching, hence optimizing the road space.

5.1.2 Architecture

As illustrated by Figure 5.1, the VANET architecture comprises vehicles communicating among themselves, and with roadside units (RSU). Each vehicle is equipped with interconnected Electronic Control Units (ECU), which comprise Hardware Security Modules (HSM) as tamper-proof

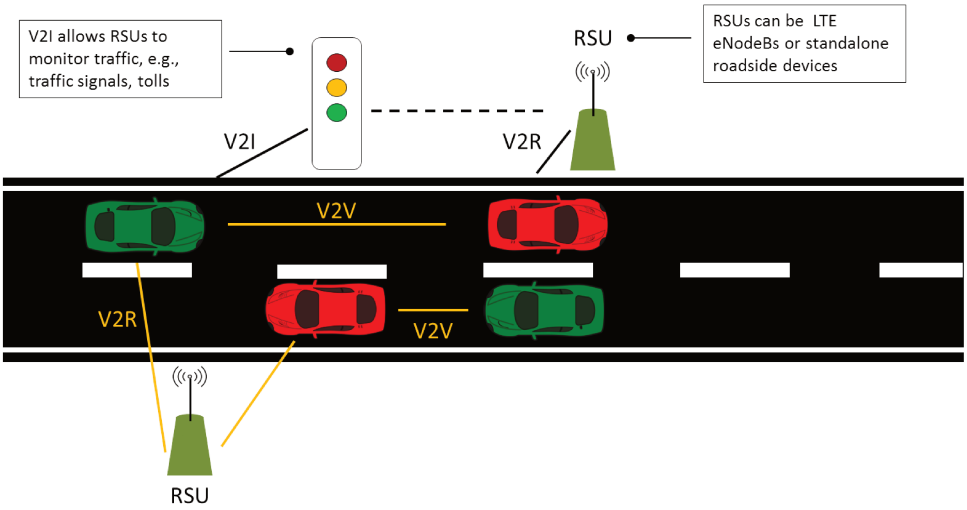


Figure 5.1: Vehicle-to-Everything Communication.

embedded systems. HSMs store cryptographic keys, as well as auxiliary security modules such as pseudo-random number generators (PRNG), in addition to securely executing encryption and authentication operations. The vehicle's on-board unit (OBU) is the central system in the vehicle, in charge of forwarding communication as well as security commands to the secure embedded systems. The E-safety Vehicle Intrusion Protected Applications (EVITA) project [EVI08, RWI+09] developed an HSM-based secure on-board vehicle architecture for the security requirements of V2X communication. For cost-reducing purposes, the EVITA architecture incorporates a HSM as well as more cost effective and lightweight security modules. An example of such alternative solutions is a SIM card embedded into an ECU chip. The crucial role of the aforementioned tamper-resistant hardware is to provide a root of trust, as well as secure key storage, and a secure enclave for the execution of cryptographic operations. The privacy-preserving communication protocol developed in this chapter leverages the security properties of a lightweight tamper-resistant hardware, namely a Java card SIM card, hence meeting the requirements of the EVITA architecture.

5.2 Vehicle-to-Everything Communication

Vehicular-to-Everything communications increase traffic efficiency and safety by allowing vehicles to share safety-related information and location-based services.

5.2.1 V2X communication frameworks

The first standardized and deployed V2X communication technology is the 802.11p-based Dedicated Short Range Communication (DSRC) wireless communication technology [JD08], which allows direct communication between vehicles, and between vehicles and road-side infrastructures without any cellular infrastructure.
The new technology for V2X communication is the Cellular-based V2X (C-V2X), standardized in 2017 in 3GPP Release 14 [3GP17]. C-V2X is based on the standardized 4G LTE and 5G mobile

connectivity to send and receive data from one vehicle to another, and from a vehicle to road-side infrastructures and pedestrians. In addition, and unlike WLAN-based V2X, C-V2X enables wide range communication over a cellular network (V2N).
In 2019, the European Commission has announced the adoption of technology-neutral Cooperative Intelligent Transportation System (C-ITS) [Com17], which in prevention of the widespread deployment of the 5G technology, allows the adoption of cellular-based V2X. In terms of message and vehicle authentication and communication security, cellular-based V2X, or the technology neutral C-ITS approaches offer more flexible security guarantees. Indeed, in their IoT security assessment report, the GSM Associations encourages network operators to use SIM-based mechanisms for the secure identification of IoT devices [GSM19]. Such recommendations also apply for the secure and privacy-preserving authentication of V2X messages, as we will demonstrate in the remaining of this chapter.

5.2.2 Safety Messaging Protocol.

In VANET vehicles send safety-related messages known as Cooperative Awareness Message (CAM) [ETS14], in order to indicate their status on the road, namely their identifier, position, speed, acceleration, and direction. Safety messages may contain information on traffic conditions, warnings regarding the position of another vehicle in its vicinity thus alerting on a potential collision situation, as well as liability-related messaged which are strongly bound to the message originator in case of liability. The information relayed in a CAM, namely a vehicle's location and identifier, requires the establishment of a secure and privacy-preserving communication protocol which protects vehicles' and their users' privacy against other individuals, as well as against authorities. Indeed, CAMs are relayed via wireless broadcast to other vehicles in the network, hence becoming subject to the threats and attacks operated over a wireless networks. In addition, a CAM is sent by a vehicle at a high frequency of one message every 200-300 milliseconds in a range of 10 seconds travel time over a single hop broadcast [JTM+06]. Moreover, the ETSI technical specification for vehicular communications stipulates that a CAM generation time should not exceed 50 milliseconds [ETS14]. The real-time constraints of the safety messaging protocol render the reduction of computational overhead a safety issue, they are therefore additional conditions to factor in the design of a communication protocol.

5.2.3 Threats and attacks in VANET

A VANET presents the same drawbacks as traditional networks based on wireless communication, namely limited data rates, low latency requirements, and the threat of external attackers tampering on the communication. Given the impact on safety and privacy, the security and privacy issues in V2X communications are at the forefront of the standardization concerns when considering the widespread deployment of VANETs. In this section, we highlight the various attacks VANETs are exposed to.

We provide in this section a general classification of security threats against VANETs. A typical attacker model includes an *insider* attacker and an *outsider* attacker, whereby an insider attacker is an authenticated member of the network with a certified public key who is able to communicate with other network members. An outsider attacker on the other hand attempts to mount attacks that do not require a certified public key, notably by exploiting network protocols. In the remaining, we consider general attacks on messages exchanged over V2X communication networks, as opposed to physical or network-specific attacks.

- *Bogus Messages*: Attackers may diffuse erroneous or bogus information on the network to affect the behavior of other drivers (e.g. divert traffic from the road they are taking by falsely announcing traffic on said road);

- *False location information*: Attackers may modify the information relayed by their traffic sensors regarding their position, speed, or direction, in order to avoid liability in case of an accident for example;

- *Tracking*: Attackers may attempt to track other vehicles by recovering their identifier ID, thus tracking other vehicles trajectories , and potentially recovering the driver's identity;

- *Denial of Service*: Attackers may perform DoS attacks on the VANET, in an attempt to bring it down or cause an accident. Examples of DoS attacks include channel *jamming*, as well as the continuous injection of bogus messages;

- *Impersonation*: An attacker may attempt to use another vehicle's identifier to send messages over the network;

- *Sybil Attack*: In an attempt to protect user privacy by employing anonymous messaging schemes, the consequence can be that an attacker pretends to be multiple vehicles. This attack, known as the *Sybil* attack [GD07], can have potentially harmful consequences in VANETs. Indeed, since it is difficult to determine whether two messages are from the same vehicle, a malicious vehicle may pretend to be other vehicles and distribute false information on the network;

- *Wormhole Attack*: In a wormhole attack [HPJ03], an attacker controls two or more nodes in the network. Upon receiving messages from one location, he *tunnels* them to the other location to replay them in the network as originating the second node.

5.3 Requirements

In this section, we present the security, privacy, and functional requirements for the secure deployment of VANETs.

5.3.1 Security and privacy requirements

A security system for safety messaging in a VANET should satisfy the following requirements:

Security requirements

- *Authentication*: Vehicles share safety-related message, which must be authenticated in order to avoid some trivial attacks. Indeed, an attacker may *replay* old messages sent over the network, as well as perform *GPS spoofing* in an attempt to tamper with other vehicles' locations. Another form of attack is *public key certificate replication*, whereby an attacker forges the valid certificate of honest vehicles. Finally, other vehicles' reaction should be based on legitimate messages, i.e. messages sent by legitimate senders. Strong authentication is therefore a critical security requirement;
- *Integrity*: Some attacks may target the integrity of messages sent over the network. Indeed, a potential insider attacker may perform message suppression, or tamper with messages sent by other vehicles. He may also perform *replay* attacks of messages of his own, or other vehicles'. In addition, malicious users may attempt to block traffic-related updates from other vehicles, in an attempt to favor their own journeys, notably by sending erroneous updates on the traffic situation. The integrity of messages must therefore be ensured in addition to the authentication of its sender;
- *Availability*: Ensuring the availability of the network is both a security and safety related issue. Indeed, vehicles not receiving real-time updates of other vehicles positions and notifications in their vicinity may result in collisions. Adversaries may mount *Denial of Service (DoS)* attacks on the communication channel, via compromised roadside units for example, thus overloading the communication channel;
- *Non-repudiation*: Driver responsibility must be ensured at all times. Indeed, drivers causing an accident should be reliably identified for liability purposes. In addition, a driver should not be able to deny sending a message after sending;
- *Non-frameability*: No vehicle or roadside unit should be able to link a given message to an honest vehicle that has not generated said message, even when they collude with the registration authority.

Privacy requirements

One of the most important features of VANET development is safety. Vehicular communication systems allow vehicles to anticipate risks of accidents and collisions by broadcasting their position, speed, acceleration in authenticated messages on vehicular networks. The authenticity and integrity of broadcast messages introduces privacy concerns, notably by revealing the position of drivers to any third party eavesdropping on the network. Communications in VANETs based on 802.11 wireless technologies notably facilitate eavesdropping. An attacker or eavesdropper may be able to trace a specific driver/vehicle, and infer mobility patterns from such data. The regulation

of privacy requirements in vehicular communication systems has led to the development of V2X communication security standards by the European Telecommunications Standards Institute (ETSI) TS 102941 [ETS09], and the IEEE Wireless Access in Vehicular Environments (WAVE) [IEE16b], which determined the following privacy requirements for communications in VANETs:

- *Anonymity*: a vehicle should be able to use a resource or service without disclosing its identifier;
- *Pseudonymity*: a vehicle should be able to use a resource or service without disclosing its identifier, while still being accountable for each action;
- *Unlinkability*: a vehicle should be able to access a resource or service multiple times without other parties being able to link such actions;
- *Unobservability*: a vehicle should be able to use a resource or service without third parties being able to observe that the resource is being used.

Functional requirements

In addition to the security and privacy requirements listed above, communications in a VANET must satisfy the following constraint:

- *Real-time constraints*: The communication rate in a VANET is typically high, and delays in relaying messages may have potential harmful consequences. Strict time constraints must therefore be respected.

5.3.2 Tamper-proof hardware

As described in Section 5.1.2, vehicles in VANET are equipped with an on-board unit (OBU), with a tamper-proof device which can be as lightweight as a SIM card. As recommended for applications where *trust* in devices is a critical aspect in guaranteeing security, it is important to leverage a tamper-proof hardware in order to provide secure authentication and identification in the safety messaging protocol. A tamper-proof embedded circuit is used to guarantee all the security properties mentioned above. It is also used for the secure storage of cryptographic keys, as well as to provide the guarantee that cryptographic operations are executed in a secure enclave.

5.4 Related Work

5.4.1 PKI-based solutions

The challenge in the deployment of VANETs is the secure and privacy-preserving transmission of safety messages, without generating communication overhead. Secure V2X communications are based on signed safety-related messages using Public Key Certificates. In order to guarantee vehicle and user privacy, PKI-based pseudonym schemes were introduced. State-of-the-art PKI-based pseudonym schemes however present scalability issues, notably due to the centralized architecture

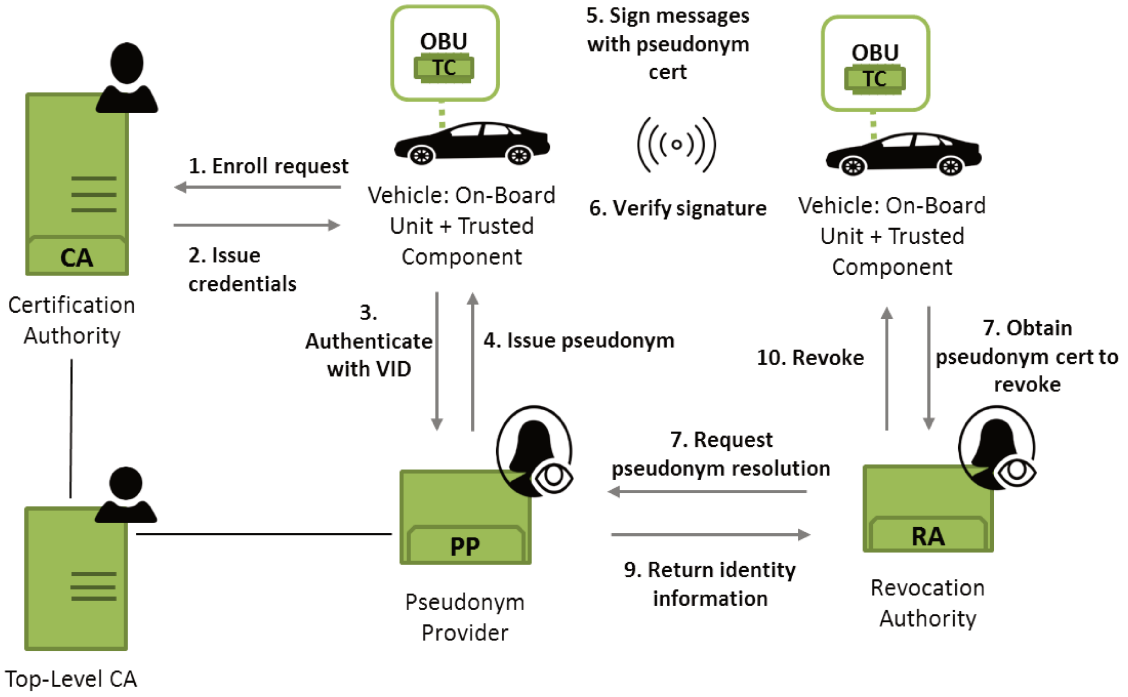


Figure 5.2: The centralized pseudonym scheme lifecycle.

of the pseudonym issuance and update phases.

The authenticity and integrity of broadcast messages induces privacy concerns, notably by revealing the position of drivers to any third party eavesdropping on the network. Communications in VANET based on 802.11 wireless technologies facilitate eavesdropping. Indeed, an attacker or eavesdropper may be able to trace a specific driver/vehicle, and infer mobility patterns from such data. Conversely, a totally anonymous communication system does not allow the system to enforce the accountability of vehicles in case of liability issues. Pseudonym schemes have been extensively developed to address the privacy, and responsibility requirements in vehicular networks. Existing solutions include asymmetric pseudonym schemes based on Public Key Infrastructures (PKI) [Coo08], Group Signatures [BBS04], Identity-based Signatures [Sha84b], and Direct Anonymous Attestation (DAA) [BCC04, BFG+13, CCD+17, BDGT17].

5.4.1.1 PKI-based pseudonym schemes

PKI architectures have been standardized by ETSI [ETS18] and IEEE [IEE16a] to secure V2X communications. Pseudonym-based protocols for V2X communications are traditionally based on public key infrastructures (PKIs), ensuring that messages exchanged on the network are authenticated, and ensuring the liability of each sender at all times.
As illustrated by Figure 5.2, PKI-based pseudonym schemes include the following entities: vehicles with a unique vehicle identifier (VID), a *Certification Authority* (CA), who is a trusted third party operated by governmental agencies, a *Pseudonym Provider* (PP) in charge of periodically

provisioning each vehicle with pseudonyms, and a *Revocation Authority* (RA) enforcing vehicle credential revocation. Initially, each vehicle in the ITS registers with the CA in order to obtain a certificate $Cert_i(VID)$ on their VID. Vehicles then authenticate with the PP using their certificate $Cert_i(VID)$. During this pseudonym provision phase, PP generates a unique pair of public/secret key (pk_{ps}, sk_{ps}) for each vehicle, and a pseudonym certificate $Cert_i(pk_{ps})$. Vehicles then periodically broadcast safety messages signed with the key certified by $Cert_i(pk_{ps})$, indicating their position, speed, acceleration, and the detection of potential road hazards to other members of the ITS. Each node periodically engages in the pseudonym update phase with PP, ensuring their anonymity across services and locations [FRF+07]. In case of liability issues or detection of a compromised or stolen vehicle, the RA engages in a pseudonym resolution phase with the PP in order to retrieve the identifier related to the compromised credential. Revoked pseudonyms and certificates are subsequently placed on a public register such as a Certificate Revocation List (CRL).

5.4.2 Existing solutions

Asymmetric pseudonym schemes for vehicular networks are divided in PKI certificate-based solutions, group signatures-based solutions, and identity-based signatures-solutions. Certificate-based and identity-based pseudonym schemes imply a centralized approach to the pseudonym lifecycle. Indeed, all solutions involve a periodic pseudonym update phase with the pseudonym provider, where all vehicles obtain new and randomized credentials. In addition, vehicles in VANET generate messages every 200-300 milliseconds [PKHK06, FRF+07], indicating relatively high communication rates. Given the nature of VANET radio communications, the centralized infrastructure is hardly scalable, creating potential bottlenecks on the network. In 2017, Whitefield et al. [WCG+17] introduced the first Direct Anonymous Attestation (DAA)-based pseudonym scheme. Their solution provides a decentralized approach for the pseudonym generation and update phases. The DAA-based construction leverages the properties of a trusted hardware component, thus allowing vehicles to autonomously generate their own pseudonyms. This proposition however requires a Trusted Component (TC) to delegate part of the pseudonym generation operations to the vehicle's On-Board Unit (OBU), introducing privacy issues in case the OBU becomes compromised. Their solution is the first decentralized pseudonym scheme to provide anonymity, user-controlled linkability, and accountability in vehicle-to-everything (V2X) communications. In the DAA-based approach, each vehicle obtains a long term credential, from which he is able to autonomously derive unlinkable pseudonyms using a trusted hardware component (TC). The vehicle signs messages with its publicly verifiable pseudonym certificates. Pseudonyms generated by the same vehicle are unlinkable, thus preserving the anonymity of drivers and vehicles in the network. Additionally, the use of trusted computing allows the revocation authority to revoke compromised credentials without the need for pseudonym resolution, or the management of costly CRLs, hence preserving the privacy of each node while reducing communication overhead. Their solution however introduces security and privacy issues, given that expensive computations are delegated to the more powerful but potentially compromised OBU. A compromised OBU might

hinder the privacy of the vehicle by introducing element in the signature computation which allow tracing signatures back to a signer.

5.5 A pre-DAA-based Pseudonym Scheme

We introduce in this chapter a new pseudonym scheme based on our pre-DAA scheme. For our pre-DAA-based pseudonym scheme, all computations during the pseudonym lifecycle (pseudonym generation, update and revocation) are executed by the trusted module, in a trusted enclave. In addition to providing a decentralized architecture for pseudonym generation, our protocol does not delegate any secure computation to the OBU which can become potentially compromised. The relation between drivers and vehicles is many-to-many, and requires a pseudonym scheme which can easily adapt to multiple drivers. Indeed, many drivers can use the same vehicle, and conversely the same vehicle might have many potential users. By introducing the first standalone pseudonym scheme, our construction offers flexibility by binding the TC to the user rather than the vehicle, hence being the first solution tailored for the many-to-many driver/vehicle relation.

5.5.1 Related work

DAA-based privacy-preserving solutions for VANET have been introduced by Chen et al. [CNW11] in 2011, to provide anonymous communication schemes which also enforce vehicle accountability. The DAA-based pseudonym scheme proposition by Whitefield et al. [WCG+17, WCS+19], is based on elliptic-curve cryptography (ECC). The DAA-based pseudonym scheme improves asymmetric pseudonym schemes in terms of *security*, *user-controlled privacy*, and *scalability*. Indeed, it allows vehicles to autonomously generate their pseudonyms, and relies on the trusted hardware to enforce revocation without compromising user privacy. The initial solution by Whitefield et al. however raises security and privacy concerns, notably due to the delegation of part of the pseudonym generation step to the host for efficiency reasons. Indeed, a compromised vehicle (therefore host) is able to include information in each pseudonym certificate in order to track every signature generated by the TC. In addition, the driver-vehicle relation is many-to-many [PKHK06] as opposed to the one-to-one relation enforced by the existing DAA scheme. A practical pseudonym scheme should therefore bind pseudonyms to users rather than vehicles. Finally, the pseudonym generation phase should ideally be executed in a secure enclave, and should be efficient for a constrained TC (typically a tamper-resistant integrated circuit (IC) card) to generate and store pseudonym certificates.

Our (pre-DAA)-based pseudonym scheme is the first decentralized pseudonym protocol where pseudonym generation, storage, update, and revocation are effectively executed by the TC. The TC can therefore be a removable trusted hardware module (e.g. a tamper-resistant IC), which binds pseudonyms to drivers rather than vehicles. This model provides flexibility between the driver-vehicle entities, while providing stronger security and privacy guarantees by using a trusted hardware to generate all authenticated broadcast messages.

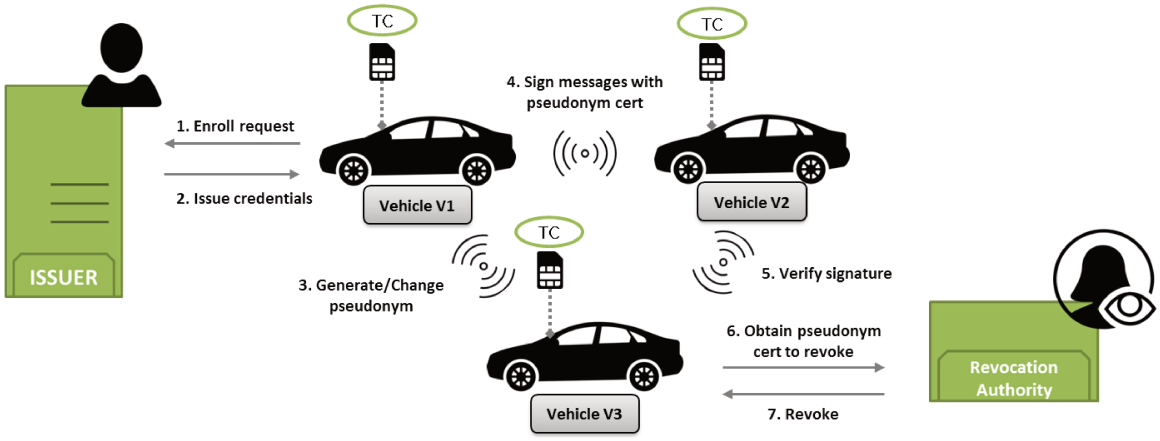


Figure 5.3: Pre-DAA-based pseudonym scheme lifecycle.

5.5.2 Our novel Protocol Construction

System entity	*Parameters*	*Description*
Issuer	pk_I/sk_I	Long-term key pair
	$K_I \leftarrow \mathcal{H}(n_I)$	Security parameter generated from a one-time nonce n_I.
TC	$pk_{ek_{tc}}/sk_{ek_{tc}}$	TC Endorsement key pair
	$s \xleftarrow{\$} \mathbb{Z}_p^*$	TC secret DAA key
	$C_s := pk(s)$	TC public DAA key
	pk_{ps}	Pseudonym public key
	cre	Blind credential
	$ps_{Cert_{tc}}$	Pseudonym
Verifier	pk_{ps}	Vehicle's pseudonym public key.
RA	pk_{ra}/sk_{ra}	RA key pair
	pk_{ps}	Pseudonym public key of the vehicle to revoke.

Figure 5.4: pre-DAA Pseudonym Scheme system parameters

Our pre-DAA-based pseudonym scheme, also illustrated by Figure 5.3 is based on our pre-DAA scheme introduced in Chapter 4.

The pre-DAA pseudonym scheme involves the following entities: a vehicle equipped with a TC executing all pseudonym generation and management operations, and an Issuer $\mathcal{I}$ who is a trusted party responsible for authenticating vehicles during the JOIN protocol. In our architecture, vehicles comprise their on-board units (OBUs) and a removable TC for the trusted computations. Given that pseudonym generation and revocation phases only require the TC, we will refer to the TC as the vehicle interchangeably in the remaining of this chapter. The infrastructure also involves additional verifier entities, who may be other users or road-side units in the ITS. We refer the reader to Figure 5.4 for a detailed description of all system parameters.

Trusted Component (TC)		**Issuer**
Public input: $pk_I, pk_{ek_{tc}}$		**Public input:** pk_I
Private input: $sk_{ek_{tc}}$		**Private input:** sk_I
	$\xrightarrow{pk_{ek_{tc}}}$	**Select** nonce $n_I \xleftarrow{\$} \{0,1\}^t$
		Compute $C \leftarrow ENC_{pk_{ek_{tc}}}(n_I \| K_I)$
	$\xleftarrow{C}$	
Decrypt C to obtain: $n_I \| K_I \leftarrow DEC_{sk_{ek_{tc}}}(C)$		
Compute C_s :=Commit(s)		
Build π_1	$\xrightarrow{C_s, \pi_1}$	**Verify** π_1
		Set pre-DAA credentials:
		Compute cre := DAAissue(C_s)
		Build π_2
	$\xleftarrow{cre, \pi_2}$	
Check the validity of π_2		
Store cre		

Figure 5.5: Overview of the JOIN protocol

5.5.2.1 Vehicle Initialization

The initial step for each vehicle is to run the SETUP protocol. To prove the authenticity of each TC, an endorsement key pair $(pk_{ek_{tc}}, sk_{ek_{tc}})$ is embedded in every TC at manufacture. During the SETUP phase, the issuer generates the system public parameters, denoted by $pp = (\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, p, P_1, Q_1, P_2, \mathscr{H}, e)$. $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ denote three cyclic groups of order p, while P_1 and Q_1 are random generator of $\mathbb{G}_1$, and P_2 is a random generator of $\mathbb{G}_2$. $\mathscr{H}$ denotes a hash function $\mathscr{H} : \{0,1\}^* \rightarrow \mathbb{Z}_p$, and $e : \mathbb{G}_1 \text{ x } \mathbb{G}_2 \rightarrow \mathbb{G}_T$ is a bilinear map (or pairing). Public parameters are published and known to every ITS entity.

During the KEY GENERATION phase, the issuer chooses two random values x_0 and x_1 in Z_p^* and sets its long term secret key $sk_I := (x_0, x_1, \tilde{x}_0)$ for a random $\tilde{x}_0 \in \mathbb{Z}_p^*$. The issuer then computes the public values $C_{x_0} = P_1^{x_0} \cdot Q_1^{\tilde{x}_0}$, $X_1 = Q_1^{x_1}, \tilde{X}_0 = P_2^{x_0}, \tilde{X}_1 = P_2^{x_1}$. The issuer's long term public key is set as $pk_I \leftarrow (C_{x_0}, X_1, \tilde{X}_0, \tilde{X}_1)$. Finally, as previously mentioned, every TC is endorsed by the manufacturer who initializes the TC with a long term endorsement key pair $pk_{ek_{tc}}/sk_{ek_{tc}}$.

5.5.2.2 Vehicle Registration

At the end of this registration phase, the vehicle will obtain the credentials *cre* certifying that it is equipped with a valid TC, which has been registered by the issuer.

The vehicle initiates the registration protocol (Figure 5.5) by sending its public endorsement key $pk_{ek_{tc}}$ to the issuer, indicating that the driver wants to join the network. The issuer generates a challenge $n_I \| K_I$, where n_I is a t-bit fresh nonce with t a security parameter. He then encrypts

TC	Passive host (OBU)
Private input: s	
Compute $\hat{cre} :=$ DAArandomize(cre)	
Compute $pk_{ps} = w^s$	
Set the pseudonym certificate:	
$ps_{Cert_{tc}} := (\hat{cre}, pk_{ps})$	

Figure 5.6: Overview of the CREATE protocol

$n_I||K_I$ with the public endorsement key $pk_{ek_{tc}}$ using an asymmetric public key encryption scheme ENC/DEC, generating a challenge C which he sends to the TC. Given that only the TC is able to decrypt the challenge using the secret endorsement key $sk_{ek_{tc}}$, the previous step establishes an authenticated channel between the TC and the issuer. The TC chooses a random secret s in $\mathbb{Z}_p^*$ as his private DAA key, which it stores in memory. It then generates its DAA public key C_s by executing the Commit function defined as follows:

> Commit: compute $C_s = X_1^s$ associated with the secret s.

The TC then builds a signature of knowledge of the secret DAA key on the challenge as follows: $\pi_1 = SoK\{\alpha : C_s = X_1^\alpha\}[n_I||K_I]$. The TC sends the public value C_s, and the signature of knowledge π_1 to the issuer. Upon receiving the message, the issuer first verifies that the TC has correctly decrypted the challenge. He then verifies that $C_s \neq 1$, before finally checking the validity of the proof π_1. He then begins computing the credential *cre* on the blinded TC secret C_s by executing the DAAissue function defined as follows:

> DAAissue: select a fresh value $b \xleftarrow{\$} \mathbb{Z}_p^*$. Computes the credential $cre = (u, u')$ associated with s, where $u = Q_1^b$ and $u' = u^{x_0} \cdot C_s^b = u^{x_0 + s \cdot x_1}$. The credential $cre = (u, u')$ corresponds in fact to a (blind) Pointcheval-Sanders signature[PS16] on the secret s.

In order to prove that the credential (u, u') is well-formed, the issuer builds a signature of knowledge π_2. The proof π_2 is defined as $\pi_2 = PoK\{\alpha, \beta, \gamma : u = Q_1^\alpha \wedge u' = u^\beta \cdot C_s^\alpha \wedge C_{x_0} = P_1^\beta \cdot Q_1^\gamma\}[n_I||K_I]$. Finally, the issuer sends *cre*, along with π_2 to the TC. During the last verification step, the TC ensures that $u \neq 1$ and that the proof π_2 is valid.

5.5.2.3 Pseudonym Generation

Authenticated and anonymous communication is established in V2X through the creation, use, and update of pseudonym certificates. The creation of pseudonyms (Figure 5.6) is executed within the TC, implying a decentralized and trusted pseudonym generation phase.
The CREATE protocol is initiated autonomously by the TC, without any external communication. The TC executes the protocol by first randomizing its credential *cre*. This randomization step is done by running the DAArandomize function which is defined as follows:

TC	Passive host (OBU)	Verifier
Private input: $s, ps_{Cert_{tc}}$	**Private input:** $ps_{Cert_{tc}}$	**Public input:** pk_I, pk_{ps}
	Generate CAM	
	$\xleftarrow{\text{CAM}}$	
Generate		
$\pi_{sign} := SoK\{\alpha : pk_{ps} = w^{\alpha}\}$[CAM]		
Set $m_{sign} := \pi_{sign}$		
Set the message:		
$msg := (\text{CAM}, m_{sign}, ps_{Cert_{tc}})$	$\xrightarrow{msg}$	$\xrightarrow{msg}$
		Compute
		$b :=$ DAAcredVerify$(ps_{Cert_{tc}})$
		If $b = 1$:
		DAAsignVerify(m_{sign})

Figure 5.7: Overview of the SIGN/VERIFY protocol

DAArandomize: Select a fresh value $l \xleftarrow{\$} \mathbb{Z}_p^*$, and compute $w := u^l$ and $w' := u'^l$. Set the randomized credential $\hat{cre} := (w, w')$, which cannot be linked to any other credential generated by the same TC.

It then computes its pseudonym public key from $\hat{cre} = (w, w')$ as follows: $pk_{ps} = w^s$. Finally, the TC generates the pseudonym certificate $ps_{Cert_{tc}}$ as follows: $ps_{Cert_{tc}} := (\hat{cre}, pk_{ps})$.

5.5.2.4 V2X Communication

A vehicle in the ITS frequently broadcasts messages in the network, indicating its position, speed, potential traffic hazards, and other safety and infotainment-related messages. Messages must contain a valid pseudonym certificate, authenticating the message as originating from a valid TC in the ITS, whilst preserving the anonymity of vehicles and drivers. Authenticated messages are generated using the Sign protocol as described in Figure 5.7.
The SIGN/VERIFY protocol is initiated by the OBU which forwards a Cooperative Awareness Message CAM containing the vehicle's position, speed and other road notifications to the TC. The TC then signs the message by generating a signature of knowledge π_{sign} on CAM as follows:
$\pi_{sign} = SoK\{\alpha : pk_{ps} = w^{\alpha}\}$[CAM]: select $r \xleftarrow{\$} \mathbb{Z}_p^*$ and compute $R = w^r$. Compute the challenge $c = \mathcal{H}(R||w||pk_{ps}||\text{CAM})$, and the response $t = r - c \cdot s$ (mod p). The resulting signature π_{sign} is set to be the challenge and response pair (c, t), and m_{sign} denotes the message π_{sign}. The TC then constructs the message $msg := (\text{CAM}, m_{sign}, ps_{Cert_{tc}})$ and broadcasts msg to the vehicular network. Upon receiving a message msg, the verifier (other vehicles and roadside units in the network) runs the DAAcredVerify function defined as follows:

DAAcredVerify: check that $w \neq 1$. If it is the case, check the validity of the blinded credentials

TC	Passive host (OBU)	RA
Public input: pk_{ra}		**Public input:** $pk_I, pk_{ps}, ps_{Cert_{tc}}$
Private input: $s, ps_{Cert_{tc}}$		**Private input:** sk_{ra}
		Generate revocation message:
		$msg := \{\text{revoke}\|\|pk_{ps}\}$
		Select $n_{ra} \xleftarrow{\$} \mathbb{Z}_p^*$
		Compute
		$\sigma_{req} := \mathsf{Sign}_{sk_{ra}}(msg, pk_{ps}, w, n_{ra})$
Run $\mathsf{Verify}_{pk_{ra}}(\sigma_{req})$	$\xleftarrow{\sigma_{req}}$ $\xleftarrow{\sigma_{req}}$	
Compute $pk'_{ps} = w^s$		
If $pk'_{ps} = pk_{ps}$:		
Compute $\pi_{rvk} = SoK\{\alpha : pk_{ps} = w^{\alpha}\}[n_{ra}]$		
Set $\sigma_{rvk} := \pi_{rvk}$		
Delete *cre* from memory	$\xrightarrow{\sigma_{rvk}}$ $\xrightarrow{\sigma_{rvk}}$	**Verify the revocation signature:** $\mathsf{DAAsignVerify}(\sigma_{rvk})$

Figure 5.8: Overview of the REVOKE protocol

by verifying that the equation $e(w, \tilde{X}_0) \cdot e(pk_{ps}, \tilde{X}_1) = e(w', P_2)$ holds. If all checks succeed, set $b = 1$, otherwise set $b = 0$.

If $b = 1$, the certificate $ps_{Cert_{tc}}$ is considered valid, and the verifier subsequently checks the validity of π_{sign} by running DAAsignVerify described as follows:

DAAsignVerify: compute $R' = w^t \cdot pk_{ps}^c$. Compute $c' = \mathcal{H}(R'||w||pk_{ps}||\text{CAM})$. If $c = c'$ accept, otherwise reject.

This final verification step, if successful, validates the authenticity of CAM. The verifier in the ITS infrastructure is the OBU of other vehicles, as the VERIFY step does not require knowing the secret key of their respective TC.

5.5.2.5 Revocation

When a compromised pseudonym is detected, the revocation authority (RA) initiates the revocation protocol with the concerned vehicle. The RA is a trusted third party in the ITS, and as such we assume he has access to the issuer's public key pk_I, as well as the pseudonym public key pk_{ps} and certificate $ps_{Cert_{tc}}$ of the vehicle to revoke.
The RA initiates the process as described by the REVOKE protocol in Figure 5.8. The RA begins by signing a revocation request message $msg := \{\text{revoke}||pk_{ps}\}$ with the secret key sk_{ra}, containing revocation instructions. He then broadcasts the signature $\sigma_{req} := Sign_{sk_{ra}}(msg, pk_{ps}, w, n_{ra})$ generated on the message, the pseudonym public key to be revoked pk_{ps}, the randomized credential w, and a freshly generated nonce $n_{ra} \xleftarrow{\$} \{0,1\}^t$ to the vehicular network. Upon receiving σ_{req},

the OBU forwards the signature to its TC. The TC then verifies the signature using pk_{ra}. If the verification is successful, the TC computes $pk'_{ps} = w^s$. If the equality $pk'_{ps} = pk_{ps}$ holds, the TC has the confirmation that its credential needs to be revoked. It then computes the signature of knowledge $\pi_{rvk} = SoK\{\alpha : pk_{ps} = w^{\alpha}\}[n_{ra}]$. The TC subsequently deletes its credential $cre = (u, u')$. Lastly, the TC forwards $\sigma_{rvk} := \pi_{rvk}$ to the RA as revocation confirmation via the OBU. The RA runs the DAAsignVerify function as described in Section 5.5.2.4, taking as input σ_{rvk}. This last step completes the REVOKE protocol, and the RA has received confirmation that the TC has deleted its keys. As in the DAA-based pseudonym construction by Whitefield et al. [WCG+17], there is a potential risk for a compromised OBU to not forward the revocation request to the TC. Such cases are addressed by the introduction of a heartbeat mechanism, such that the TC periodically expects either a revocation message, or a heartbeat including a fresh timestamp signed by the RA. After a prolonged period without receiving either, the TC presumes potential malicious behavior, and is able to autonomously initiate the revocation phase.

5.5.3 Efficiency analysis

This section provides a comparative analysis between the computational efficiency of the DAA-based scheme [WCG+17, WCS+19] and our pre-DAA-based scheme.
In Figure 5.9, $k\mathbb{G}_i$ denotes k exponentiation (in fact scalar multiplication in elliptic curve groups) in the group $\mathbb{G}_i$, and $k\mathbb{G}_i^j$ represents k j-multi exponentiation in $\mathbb{G}_i$. kP denotes k pairings computations, while kE and kD represent k asymmetric encryption and decryption respectively (corresponding to the (ENC/DEC) algorithms in the protocol description in Figure 5.5). kS and kV denote k signature generation and verification respectively (corresponding to the (Sign/Verify) algorithms in the protocol description). $k\mathcal{H}$ represents k hashes and k MAC indicates k MAC computations (e.g. HMAC).
As presented in Figure 5.9, the JOIN phase in the DAA scheme requires the delegation of pairing computations to the host. Unfortunately, pairings are prohibitively expensive and cannot be undertaken by a constraint TC. Our solution prevents this delegation step by providing a much more efficient JOIN phase, which does not require any pairing computation. The JOIN phase in our protocol can therefore be undertaken by the TC alone, thus preventing security and privacy breaches. In addition, the CREATE, VERIFY, and REVOKE protocols are more efficient in our pre-DAA-based solution. Given the high communication rate in vehicular networks (at least one message transmitted every 200-300 ms), our solution generates minimal overhead during message authentication and transmission in the network.

5.6 Security Analysis

The nature of VN communication radios makes vehicles vulnerable to eavesdropping attacks. Vehicles can easily be remotely tracked by adversaries, which can result in unlawful surveillance, or the generation of mobility patterns of different users. Attackers may for example exploit such information to track the whereabouts of drivers from their home to their workplace. Adversaries in

DAA	TC	Host	Issuer	Verifier	RA
SETUP			$2\mathbb{G}_2$		
JOIN	$2D$,$3\mathbb{G}_1$, $1\mathcal{H}$, $1MAC$	$4P$	$2E$, $2\mathbb{G}_1^2$, $2\mathbb{G}_1$, $1\mathcal{H}$, $1MAC$		
CREATE	$2\mathbb{G}_1$, $1\mathcal{H}$	$4\mathbb{G}_1$, $1\mathcal{H}$			
SIGN	$1\mathbb{G}_1$, $1\mathcal{H}$				
VERIFY				$4P$, $2\mathbb{G}_1^2$,$3\mathcal{H}$	
REVOKE	$1\mathbb{G}_1$, $1V$, $1\mathcal{H}$	$4\mathbb{G}_1$			$1S$, $4P$, $2\mathbb{G}_1^2$, $2\mathcal{H}$

pre-DAA	TC	Host	Issuer	Verifier	RA
SETUP			$1\mathbb{G}_1^2$, $1\mathbb{G}_1$, $2\mathbb{G}_2$		
JOIN	$1D$, $2\mathbb{G}_1^3$, $1\mathbb{G}_1^2$, $2\mathbb{G}_1$, $2\mathcal{H}$		$1E$, $4\mathbb{G}_1^2$, $2\mathbb{G}_1$, $2\mathcal{H}$		
CREATE	$3\mathbb{G}_1$				
SIGN	$1\mathbb{G}_1$, $1\mathcal{H}$				
VERIFY				$3P$, $1\mathbb{G}_1^2$, $1\mathcal{H}$	
REVOKE	$2\mathbb{G}_1$, $1V$,$1\mathcal{H}$				$1S$, $1\mathbb{G}_1^2$, $1\mathcal{H}$

Figure 5.9: Computational efficiency comparison of DAA-based Pseudonym Schemes
Gray cells: N.A.

VANETs are either internal, i.e. legitimate members of the ITS, or external. External adversaries can have the additional motive of disrupting the network by forging valid pseudonyms, thus compromising the authenticity of communications. We assume cryptographic protocols to be secure, and the TC within each vehicle to be a tamper-resistant hardware module such as a SIM card or a TPM (see Section 3.2.1). Therefore the TC creates a secure enclave for cryptographic key storage and secure computation. The endorsement key provides the first security guarantee that deployed TCs are validated by the manufacturer. The subsequent operations in the pseudonym lifecycle are all executed by the TC, ensuring that no security breaches can occur. Therefore, we only consider adversaries who attempt to forge vehicles' credentials, in order to perform malicious attacks. Such forgeries are prevented based on the security properties of our pre-DAA scheme introduced in Chapter 4. In the following, we analyze the key security properties of our pre-DAA-based pseudonym scheme:

- **Anonymity.** In the signature generation process, a vehicle uses its randomized pseudonym obtain from running the CREATE protocol. Given a signature msg generated on a CAM, an adversary cannot determine whether the signature was generated using the randomized pseudonym certificate $ps_{Cert_{tc}} := (\hat{cre}, pk_{ps})$, or another pseudonym based on the eXternal Diffie-Hellman (XDH) assumption. Indeed, the anonymity of the signer is based on the *anonymity* property of the underlying pre-DAA scheme, which, as described in Section 4.4, holds under the XDH assumption. This property is also guaranteed by the fact that all signature generation steps are undertaken by the TC without any delegation to the OBU. In case it becomes compromised, the OBU is thus not able to attach any information that could

identify the vehicle to the signature without being detected.

- **Non-repudiation.** Vehicle liability is guaranteed by the presence of a TC which stores a private key s used in the interactive registration protocol between the vehicle and the issuer. Indeed, the verification of a signature $m_{\text{sign}} = \pi_{\text{sign}}$ where $\pi_{\text{sign}} = SoK\{\alpha : pk_{ps} = w^{\alpha}\}$[CAM] generated by a vehicle proves that the signer possesses a secret s which has been certified by the issuer. Notably, the *traceability* property of the underlying scheme ensures that no vehicle can generate a message that cannot be traced back to a valid pseudonym key that was generated during the interactive registration protocol between the vehicle and the issuer.

- **Non-frameability.** In order to frame other vehicles, an attacker must be able to link a given signature to an honest vehicle that has not generated said signature. The *non-frameability* property of the pre-DAA scheme ensures that attackers cannot generate signatures which can be linked back to the pseudonym of an honest vehicle. The property is proven secure under the One-More Discrete Logarithm (OMDL) assumption in Section 4.4.

5.7 Implementation and Evaluation

We present in Table 5.1 the performances obtained from our protocol implementation on a GlobalPlatform-compliant Javacard SIM card, embedded in a Samsung Galaxy S5 NFC smartphone. The SIM card, which acts as a TC, randomizes its credentials, generates its pseudonyms and anonymously signs messages. It then transmits the resulting anonymous signatures to the OBU of other vehicles (in our prototype the processor of a Samsung Galaxy S5) via the application protocol data unit (APDU). As presented in Table 6.1, the transmission time generates most of the overhead, whereas signature computation has timings comparable to symmetric key algorithms. These optimal timings are due to the fact that almost all the operations carried out to randomize credentials or generate pseudonyms can be precomputed by the SIM card. Only the pair (c, t) has to be computed on-the-fly during signature generation.

Signature generation (Card)	**Signature verification (OBU)**
(36-48) 38.01 ms	(4-16) 11 ms
Transmission time (card to OBU)	
(33-43) 34.52 ms	
Total signature generation time	
(3-5) 3.49 ms	

Table 5.1: Timing (min-max) average in milliseconds of the pre-DAA protocol.

5.8 Conclusion

In this chapter, we study the limitations of a centralized pseudonym scheme as an authentication protocol for Vehicular Ad-Hoc Networks. The centralized model, in addition to generating communication overhead for a time-sensitive use case, introduces privacy shortcomings. Indeed, multiple signatures generated with the same pseudonym can be traced back to a given user. We introduced a pre-DAA-based pseudonym scheme which leverages the presence of a trusted hardware module to provide a decentralized, privacy-preserving pseudonym scheme for V2X communications. The pseudonym generation and update steps can be executed solely by the secure element, thus achieving the security and accountability properties of a pseudonym scheme without compromising the vehicle's privacy. In applying the standalone pseudonym generation and update model, vehicles are able to securely generate their own pseudonyms, while remaining accountable in case of liability issues. Indeed, the pre-DAA signature strongly binds each signed safety message to the driver, as opposed to linking them to the vehicle owner. In addition, potential threats arising from delegating part of the secure operations to the vehicle's OBU are leviated, hence ensuring that no tracing of the vehicle can take place. In the next chapter, we apply the same trusted component-assisted authentication paradigm to the mobile ticketing in public transport use case.

CHAPTER 6

A Privacy-Friendly Mobile NFC Transit Pass Service

Mobile TEE-based and more precisely Universal Integrated Circuit Card (UICC)'s have led to the widespread deployment of secure applications, notably those requiring secure access control. Indeed, such applications have strong security requirements, which have recently been topped by privacy requirements. Indeed, the strong identification and authentication of users on platforms implementing said applications must implement the data minimization property of only collecting the data required to use the service.
In this chapter, we introduce Pass-As-You-Go (PAYGO), a provably secure and privacy-preserving mobile transit pass service for public transport systems. PAYGO is a practical and secure mobile transit pass protocol, which allows the secure and anonymous authentication of commuters on a public transport network, as well as the unlinkability of their trips. The service implements a subscription-based authentication protocol, which leverages the security and privacy properties of our pre-DAA scheme. We also evaluate the efficiency of PAYGO, from a testbed implementation on a Global Platform-compliant Java card smart card. PAYGO is the first step towards the widespread deployment of transport pass services that implement privacy-by-design rather than relying on user consent in order to comply with the increasingly stringent user data privacy rules and regulations.

The contributions detailed in this chapter were presented at the *Real World Crypto 2020* Symposium, and are currently in submission [DDT].

Contents

6.1 Mobile NFC for Transport

6.1.1 Introduction

In 2019, the city of Rio de Janeiro has deployed the first NFC mobile transit pass, in collaboration with the French smart card group Gemalto and the Brazilian transport operator RioCard Tecnologia da Informação [Gem18b]. Commuters in Hong-Kong have been using the technology for the secure digitization of their Octopus smart card, commonly used for contactless payments, notably on public transports [Gem18a].

The explosion of mobile services in the past two decades was initiated by mobile network operators, who were highly interested in diversifying the usage of mobile services for ubiquitous applications in users' daily lives. The Near Field Communication (NFC) capability of modern smartphones, has notably induced the widespread deployment of applications such as electronic banking, smart payment, smart ticketing, and mobile transit passes. Commuters with an NFC-enabled smartphone are able to store their subscription credentials or transport tickets converted into smart tokens directly into their smartphones. They tap their handsets on NFC-enabled turnstiles, which are equipped with an NFC-enabled reader. The technology presents a number of advantages for transport operators, mobile network operators, and users. Indeed, the service allows fast and secure authentication, as well as usability advantages. The pervasive nature of public transport systems coupled with the invasive impact it may have, notably on identifying and tracing commuters, led to specifically center security and privacy as a key research and development domain for mobile transport passes [GSM12].

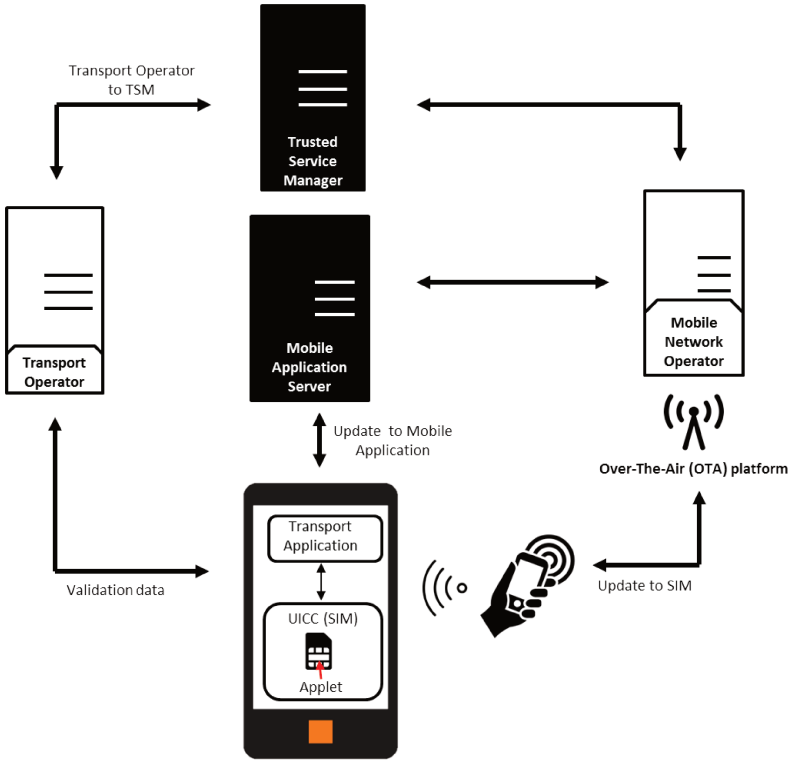


Figure 6.1: Overview of the NFC-enabled mobile transit pass framework.

6.1.2 Technology and Architecture

6.1.2.1 Near Field Communication Technology

Near Field Communication (NFC) is a standardized set of protocols for short distance radio communication. The technical standards for NFC communication include the physical layer radio-frequency identification (RFID) communication standards ISO/IEC 18092 [ISO13b] and ISO/IEC 14443 [ISO18], as well as the international standard ISO/IEC 7816 [ISO19], specifying smart cards' interfaces, physical components, and the communication protocols. ISO/IEC 14443 compliant cards operate at 13.56 MHz and have an operational range of up to 10 centimeters. The reader initiates the communication by sending out a signal to the NFC tag. The tag is powered by the reader's signal, which allows it to be able to function without being battery-powered. The technology is therefore especially useful for implementing secure authentication and communication in constrained environments.

6.1.2.2 Architecture of an NFC-enabled mobile transit pass service

Contactless ticketing provides two different options for users, namely physical cards or mobile NFC passes. The first solution is the most widely deployed solutions, whereby a users transport credentials are stored in the smart card embedded in a physical pass. The second option offers the possibility to store transit pass credentials directly into the embedded Universal Integrated Circuit Card (UICC), commonly known as Subscriber Identity Module (SIM) card, present in each mobile phone. Indeed, as explained in Section 3.2.1, the NFC capabilities of modern smartphones

enables the NFC-enabled reader to directly communicate with the transport applet in the Java Card SIM card. The latter option offers a number of benefits for users, transport operators, and mobile network operators. The advantages include the fast and secure authentication of user's tickets or transit passes, the reduction of queuing in front of pass purchase or recharge counters and tickets booths, efficient wallet management, and the potential for inter-operator transport services. Figure 6.1 displays the overall architecture of a mobile ticketing service. The service comprises a Transport Operator (TO) and a Mobile Network Operator (MNO), who communicate with a Trusted Service Manager (TSM) in order to deploy a secure system. The MNO manages, provisions, and communicates with the SIM card Over-The-Air, and retrieves data via the mobile application server. The TO manages the transport service, notably using the data forwarded by the local server, which in turn retrieves transit pass validation data from the NFC-enabled reader.
The architecture for a mobile NFC solution for the transport application comprises the following stakeholders:

- A Transport Operator (TO), who manages the transport service and is responsible for the secure deployment and management of credentials assigned to each registered user on the transport network;

- A commuter or User (U), who subscribes for a periodical transit pass. The transit pass is linked to the user's identity in an initial registration phase, and should be deployed by a transport operator cognizant of each user's responsibility, security, and privacy. U's transit pass is stored in a dedicated applet with application ID AID. As described in Section 3.2.1, Java Card SIM cards implements tamper-proof enclaves which store cryptographic keys, and allow the secure execution of cryptographic protocols;

- An NFC-enabled reader (V), which authenticates and validates commuters' passes in front of turnstiles or when boarding buses or trains. The reader communicates the validation data and metadata to a remote server managed by the transport operator.

- A third party revocation authority (E), who safeguards user privacy by sharing the anonymity lifting keys and revocation keys with the transport operator. This distributed scenario prevents the transport operator from single-handedly retrieving a user's identity and mobility data in clear. The revocation capabilities are split between the transport operator and the revocation entity, and the revocation process can only take place when both entities jointly collaborate. The revocation entity, whose impartiality regarding the privacy of users becomes a key aspect of his role, can in some cases be the union representative for commuters for example.

6.1.3 Requirements

6.1.3.1 Privacy concerns

The new data regulations have defined more stringent privacy requirements for transport operators over the past few years. Notably, subscription-based transit passes (e.g. the Navigo pass in the

Île-de-France region) should strive to preserve the anonymity of commuters by default, rather than imposing implicit consent from users to use a service on which they can be authenticated. The notion of privacy on transport network therefore includes *user anonymity*, whereby it is impossible to trace a pass validation at a given station back to the user's identity. This notion is however insufficient, as the ability to trace multiple pass validations to a single user may lead to recovering the user's identity from his daily itinerary. The notion of privacy must therefore also include *untraceability*, where it is impossible to link two or more validations to the same identifier. No solution has been deployed yet to satisfy the latter requirement.

6.1.3.2 Security requirements

Considering the privacy concerns introduced above, an anonymous mobile transit pass service must satisfy the following security properties:

- *Consistency:* a valid pass (which obtained valid credentials from the transport operator in a previous registration process) must be granted access by the reader;
- *Unforgeability*: it must not be possible for a defrauder to forge the pass credentials, or to modify them in the card;
- *Anonymity:* it must not possible to distinguish a signature generated by a given user ID_0 from a signature generated by a user ID_1;
- *Unlinkability:* it must not be possible for the transport operator to (1) link two validations generated by the same pass, or (2) recover the identifier ID of a given user from a validation signature σ;
- *Anti-passback*: whilst the service must enforce *anonymity* and *unlinkability*, it must not be possible for defrauders to validate the same pass consecutively for different users.

6.1.3.3 Functional requirements

A privacy-preserving mobile transit pass service must satisfy the following functional requirements:

- *Efficiency:* the authentication process (signature generation and verification) must satisfy the same stringent timing requirements as a non-anonymous pass authentication protocol;
- *No pairing*: given that all signature generation computations are undertaken by the SIM card, which cannot handle pairing computations, the signature generation step must not require any pairing computations on the pass side.

6.2 Related Work and Motivation

In this section, we present the existing work on mobile transit passes and discuss their limitations. We then discuss the motivations for our contribution.

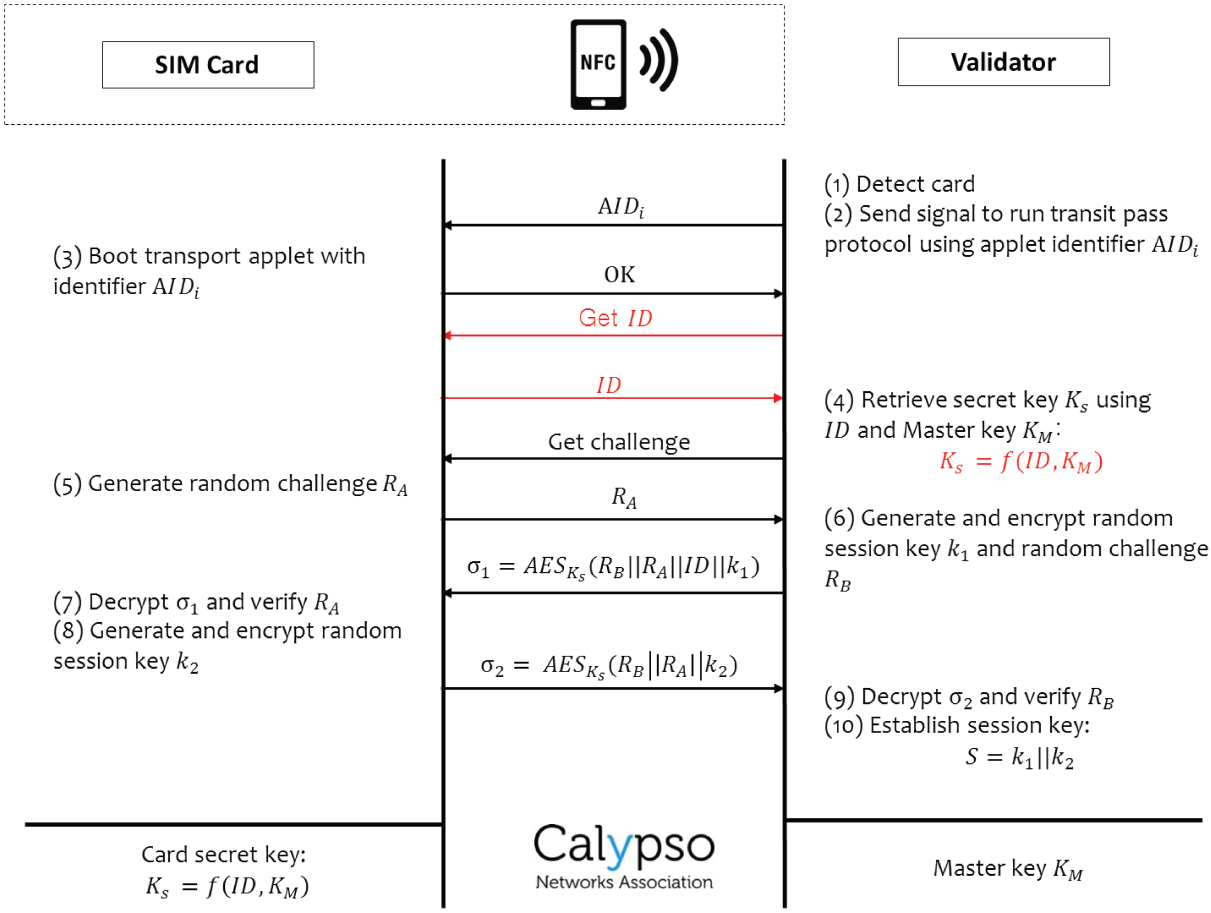


Figure 6.2: Overview of the Calypso Secure Authentication step.

6.2.1 The Calypso Standard

A consortium of European transport operators from Belgium, France, Germany, Italy, and Portugal, initiated a standardization project with the aim to propose an open technology for portable devices, in order to deploy a secure and efficient public transport ticketing system known as Calypso. The Calypso standard [Ass14] is now deployed all over the world, as the standard for mobile NFC ticketing systems. Notably, the examples mentioned in Section 6.1.1 are based on the Calypso standard, as well as other worldwide deployed commercial solutions [Gem18a, Gem18b]. The widespread deployment and adoption of the standard stems from a desire for interoperable transport pass systems, as well as the robust security specifications in the Calypso norm. Indeed, as presented in Figure 6.2, the secure authentication and validation of a transit pass works as follows:

- The reader detects a user's transit pass, and establishes an NFC communication field with the NFC tag (embedded in the mobile phone);

- The reader retrieves the unique user ID of the transit pass, which is stored in the SIM card. The unique user ID is used in the generation of the shared secret key K_S;

- Authentication schemes based on symmetric key encryption schemes require each party to generate a shared session key using a common master key. The secret key K_S is therefore used to encrypt the shared session key S using a symmetric key encryption scheme such as AES;

- The reader having authenticated the card, is now able to check the validity of the user's subscription details.

6.2.2 Solutions based on Public-Key Cryptography

The cryptographic framework for a secure transport service was first discussed by Heydt-Benjamin et al. [HCDF06]. They establish the functional, security, and privacy properties inherent to such services. A number of mobile transit pass solutions have since been proposed [DPWD11, ET11, RHBP13, SVW08, TE13], each presenting limitations that we briefly describe in this section.
Initially, solutions employing cryptographic protocols for secure smart ticketing defined an incomplete privacy model, whereby the user remains anonymous with respect to any outside entity, but not the transport operator [ET11, SVW08, TE13]. A more complete model must however include the unlinkability of trips in order to avoid tracing users across the transport network, whilst preserving the ability to link validations when users attempt to use their transit pass for multiple validations (anti-passback property). Privacy-preserving solutions addressing both anonymity and unlinkability have since been proposed, based public-key cryptographic schemes such as set-membership proofs [ALT+15], and Direct Anonymous Attestation [DLST14]. Arfaoui et al. [ALT+15] proposed an m-ticketing system where users store a set of tickets directly into their smartphones. It does not however address the case of transit passes, where users obtain long term credentials that are valid for an unlimited number of trips, depending on the subscription plan. The unlinkability property for such cases is therefore not addressed. Desmoulins et al. [DLST14] proposed a transit pass solution based on Direct Anonymous Attestation [BCC04], which meets the required anonymity and unlinkability properties. Their validation step however requires delegating part of the validation computation to the mobile phone processor, due to the limited computational capabilities of the secure element. This constraint introduces privacy concerns, as the user's smartphone is the user's phone affects the privacy guarantee of the transport application. In addition, a key functional requirement for NFC mobile transport services is the ability to use the transit pass even in the case where the phone is switched off, or if its battery is empty. This property can only be achieved by using a stand-alone authentication scheme, which can be solely undertaken by the NFC SIM card which, in the event that the smartphone battery is empty, will be indirectly powered by the NFC reader. The efficiency requirement specifying that the pass validation time must not exceed 300 ms [GSM12] introduces further challenges when designing said stand-alone solution.
PAYGO provides a solution which meets both these stringent privacy and efficiency requirements, by leveraging the properties of our new pre-DAA construction.

6.2.3 Motivation

The Calypso standard presents privacy shortcomings which, in light of the new directives enforced by the European Union regarding user data privacy [EU], are to be addressed in a formal setting. Indeed, the notion of privacy for services which bound a user's identity to his responsible use of the system include two notions, namely *anonymity* and *untraceability*. In the public transport setting, the first notion stipulates that it should not be possible to trace a user's pass validation back to its unique identifier. In the case where a user's trajectories are linked and repetitive, the sole notion of *anonymity* does not prevent from tracing back a given transit pass to the concerned user's identity. The metadata generated from transit pass validations is enough to trace users, and obtain information on their daily habits. Indeed, the Calypso norm for a secure transit pass requires the pass to provide its identifier ID in clear in the first steps of the authentication phase with the reader. Upon detecting the same validation patterns from the same pass ID, the transport operator is able to derive accurate trajectories for the same ID, thus lifting the anonymity of each pass validation. The *untraceability* notion enables to overcome this shortcoming. Implementing an *untraceable* transit pass protocol however requires to factor a number of functional issues. Indeed, a completely untraceable pass might induce a fair evasion problem, whereby it would be impossible to detect when two users use the same pass consecutively (also known as *passback*), or to detect clones. Provably secure cryptographic schemes, notably Direct Anonymous Attestation schemes, enable the construction of transit pass protocols with *controlled-linkability* (thus conveying the *anti-passback* property), whilst ensuring that honest users remain anonymous and untraceable at all times.

In addition to the privacy requirements, stringent efficiency requirements for the transport case are to be factored in when designing secure access control protocols. Notably, in order to avoid congestion in front of turnstiles, the pass validation step should take less than 300 milliseconds [GSM12]. Privacy-preserving solutions for transit passes have so far addressed the anonymity issue [DPWD11, ET11, TE13], while neglecting the trip unlinkability property. Solutions which have addressed the latter issue, still suffer from efficiency issues, leading to the delegation of computationally expensive secure operations to the mobile phone [DLST14]. The latter solution however hinders the privacy of users, as a compromised mobile phone might leak information about the user's identity. In addition, a state of the art mobile transit pass should still provide validation functionalities, albeit limited, when the phone runs out of battery. By delegating part of the authentication computation to the mobile phone, the latter issue remains.

We introduce Pass-As-You-Go (PAYGO), an NFC-enabled transport service, which enables the anonymous validation of transit passes, while preserving the unlinkability of each validation. Our pre-DAA scheme is efficient enough to be executed by an element as constrained as a SIM card. As the SIM card performs all the computations required during the validation phase (in a stand-alone manner without delegation to the mobile phone), this avoids any tracking of the user's journeys through the potentially compromised mobile phone. In addition, the stand-alone authentication by the SIM card implies that the pass validation still works when the phone is switched off or even when the phone runs out of battery (in both cases the NFC SIM card will be powered by the reader

via NFC). Our DAA scheme implies that a user is able to validate a transit pass whilst remaining anonymous, even when faced with a malicious transport operator. In addition, pass validations by the same user are unlinkable, provided that they are not performed during a short (predetermined) time span (e.g. 10 minutes). Otherwise, both validations would be linkable in order to detect fair evasion (i.e. to detect a malicious user passing back his transit pass to a second person who wants to gain access to the public transport system).

6.3 Pass-As-You-Go: Protocol Description

In this section, we describe our PAYGO protocol which implements a privacy-preserving mobile transit pass service. Our protocol is the first to implement trip unlinkability, whilst ensuring user responsibility through the anti-pass-back property.

6.3.1 Overview

The PAYGO architecture is defined as follows: the transport operator (TO) who manages the transport service acts as the issuer ($\mathscr{I}$). Each user (U) with a smartphone registers for a valid transport pass stored in their SIM card $\mathscr{SE}$, thus becoming part of the group of commuters with a valid pass. Turnstiles at stations are equipped with an NFC-enabled reader, which act as the verifier ($\mathscr{V}$) in the pre-DAA architecture. In PAYGO, the pre-DAA scheme is extended to include a third entity, namely an extractor (E). E represents a third party whose role is to safeguard the privacy of users on public transport networks, notably regarding the transport operator. This allows sharing the revocation capabilities between entities with diverging interests. Indeed, the extractor and the transport operator must collaborate in order to lift the anonymity and revoke the credentials of a fraudulent user.

PAYGO comprises three main phases:

- *Setup*. In this phase, TO generates the public parameters of the group, as well as the public and private keys for TO and $\mathscr{SE}$. The group in the public transport network is the set of users with a valid transit pass (i.e. users who have *registered* with the transport operator).

- *Registration*. A users obtains the credentials for his transport pass by engaging in an interactive *Registration* protocol with TO, credentials which are in turn stored by $\mathscr{SE}$. During the *Registration* phase, TO and $\mathscr{SE}$ execute the *Join*/*Issue* protocol of the pre-DAA scheme, which results in $\mathscr{SE}$ obtaining its authentication credentials (i.e. the pre-DAA group signing key).

- *Validation*. During each validation phase, the reader ($\mathscr{V}$) sends a random challenge and a basename to the smartphone, detected via NFC. The basename is used to prevent users from swiping their pass twice, also known as the anti-passback property (see Section 6.3.4.3). The SIM card uses the credentials to generate a valid pre-DAA signature, thus anonymously

SIM card ($\mathscr{SE}$)		**Transport Operator (TO)**
Public input:		**Public input**:
$pp, gmpk, epk, pk_{Pai}, ID_U$		$pp, gmpk, epk, pk_{Pai}$
Private input: esk, s_1		**Private input**: $gmsk$, REG, TO's share of sk_{Pai}
Compute $C_{s_1} = X_1^{s_1}$, $\mathscr{S} = Sign_{esk}(C_{s_1})$, and $C_{Pai} = \mathrm{Enc}_{pk_{Pai}}(s_1) = g_P^{s_1} r^n \pmod{n^2}$		
Build $\pi_1 = SoK\{\alpha : C_{s_1} = X_1^{\alpha} \wedge C_{Pai} = g_P^{\alpha} r^n\}[ID_U]$	$\xrightarrow{(C_{s_1}, C_{Pai}, \pi_1, \mathscr{S})}$	**Check** $C_{s_1} \neq 1$ and **Verify** $\pi_1, \mathscr{S}$
		Select $b, s_2 \in \mathbb{Z}_p^*$
		Compute the group signing key $gsk = (u, u')$
		where $u = h^b, u' = u^{x_0}[C_{s_1} X_1^{s_2}]^b = u^{x_0 + x_1(s_1 + s_2)}$
		Build $\pi_2 = SoK\{\alpha, \beta, \gamma, \mu : u = h^{\alpha} \wedge$
		$u' = u^{\beta}[C_{s_1} \cdot C_{s_2}]^{\alpha} \wedge C_{x_0} = g^{\beta} h^{\gamma} \wedge C_{s_2} = X_1^{\mu}\}[m_0]$
		where $C_{s_2} = X_1^{s_2}$. **Compute** $C_s = C_{s_1} \cdot C_{s_2}$
Verify π_2. **If** $C_{s_1} \cdot C_{s_2} = 1$ **abort**	$\xleftarrow{((u,u'), C_{s_2}, \pi_2)}$	
Compute $\sigma_0 = Sign_{esk}(C_{s_2}, u, u', \pi_2)$	$\xrightarrow{\sigma_0}$	**Verify** σ_0
Check that $C_{s_2} = X_1^{s_2}$ and **Set** $s = s_1 + s_2$ (mod p)	$\xleftarrow{s_2}$	**Store** $(ID_U, C_s, C_{Pai}, \sigma_0)$ in REG

Figure 6.3: PAYGO registration protocol.

authenticating itself. If the authentication process succeeds, $\mathscr{V}$ grants access to the user with pass $\mathscr{SE}$. Access is otherwise denied.

An additional *Revocation* phase addresses cases where a user's credentials are to be revoked, notably upon fraud or clone detection.

6.3.2 Setup

The *Setup* phase consists in running the *Setup* and *Keygen* algorithms of the pre-DAA scheme as defined in Sections 4.3.1 and 4.3.2. Essentially, this phase allows the transport operator to generate the public parameters, and its public/private key pair $(gmpk, gmsk)$. It is also during this phase that $\mathscr{SE}$ generates its secret key s_1. In parallel, TO and E generate the public and private keys $(pkPai, skPai)$ for a threshold Paillier cryptosystem, which is used to extract a secret key from a given commitment during the revocation phase (as detailed in Section 6.3.5).

6.3.3 Registration

A user obtains a weekly, monthly, or yearly transport pass subscription by registering with the transport operator. We denote by ID_U a user's identifier, and REG the database where TO stores the unique identifiers of registered user. The technical description of the registration protocol is detailed in Figure 6.3. Let us provide the intuition behind our construction. The *registration* phase is an extension of the *Join-Issue* protocol of our pre-DAA scheme. A user (U) (defined here by the

corresponding SIM card $\mathscr{SE}$) obtains a blind PS signature [PS18] (u, u') on his secret key s, where s is jointly computed by $\mathscr{SE}$ and TO: $s = s_1 + s_2$, where s_1 is chosen by $\mathscr{SE}$ and is unknown to TO, while s_2 is chosen by TO and is sent to $\mathscr{SE}$ at the end of the *Registration* protocol. In addition, $\mathscr{SE}$ computes C_{Pai}, a Paillier encryption of s_1, which can be retrieved and decrypted by revocation entities. Indeed, using the threshold version of the Paillier cryptosystem (see Section 2.5.1.3) allows to reduce trust in the decryption entity, by sharing the decryption capabilities (therefore secret keys) between l entities. Threshold decryption allows any subset t out of l entities to decrypt a ciphertext, but disallows the decryption if less than t entities participate. During registration, $\mathscr{SE}$ generates a threshold Paillier encryption on his secret s_1 using the Paillier encryption key generated during the *Setup* phase, thus allowing a set of revocation authorities (comprised of TO and E) to jointly decrypt the ciphertext during the *revocation* process.
At the end of the registration phase, TO knows the PS signature (u, u') generated on s (which will be randomized, when used by $\mathscr{SE}$ as his group signing key), while the secret s remains hidden. TO stores the credential (u, u'), as well as $\mathscr{SE}$'s commitments C_{s_1} and C_{Pai} in a private register REG. The registration phase corresponds to the *Join*/*Issue* protocol of a pre-DAA scheme, at the end of which $\mathscr{SE}$ obtains credentials associated with its secret key s.

6.3.4 Validation

A user is able to use his pass to anonymously authenticate at access control points. During this pass validation phase, $\mathscr{SE}$ makes use of the *Sign* algorithm of the pre-DAA scheme. Each turnstile is equipped with an NFC-enabled reader ($\mathscr{V}$), which detects the user's mobile via NFC connectivity. Once the connection is established with the transport applet, $\mathscr{V}$ (who corresponds to the verifier in the pre-DAA architecture) generates a 256-bit random challenge Ch. The random challenge reinforces security by preventing a user from preparing the responses in advance. $\mathscr{V}$ also sets the basename bsn corresponding to the validation time period P_j. Specifically, the list of basenames are generated in advance by TO and a second revocation authority (E), which we assume will not collude with TO. The basename generation process, as detailed in Section 6.3.4.1, enables anonymity revocation in the specific case of fraud detection. Said anonymity revocation process requires the joint collaboration of both TO and E, in order to prevent illegitimate or abusive revocations.

6.3.4.1 Basename generation

TO and E jointly generate the basenames in advance, for pre-determined time periods P_j. Indeed, this enables the anonymity revocation authorities to initiate the *revocation* process in the case of fraud detection 6.3.5, as well as to enforce the anti-passback property during *validation* phase 6.3.4.3. The basename generation process is described as follows:

1. Depending on the system policy, a time period P_j for $1 \leq j \leq n$ ranges from a few seconds to a few minutes.

SIM card ($\mathscr{SE}$)		**Reader (V)**
Public input:		**Public input**:
$pp, gmpk$		$pp, gmpk$
Private input: s, gsk		**Private input**: N/A
		At time period P_j, **Choose** $Ch \xleftarrow{\$} \mathbb{Z}_p$
Select $l \xleftarrow{\$} \mathbb{Z}_p^*$	$\xleftarrow{Ch, bsn}$	**Set** $bsn = X_{P_j}$
Compute a randomized version (w, w') of the credentials (u, u') where $w = u^l$ and $w' = (u')^l$		
Compute $c = w^s$, **generate** the tag $T = bsn^s$		
Build a zero-knowledge signature of knowledge $\pi_3 = SoK\{\alpha : c = w^\alpha \wedge T = bsn^\alpha\}[Ch]$ of a valid PS signature (w, w') generated on the message s		
Set the signature $\sigma = (w, w', \pi_3, c, T)$	$\xrightarrow{\sigma}$	**Check** that $w \neq 1$ and $T \neq 1$
		Verify that $e(w, \tilde{X}_0) \cdot e(c, \tilde{X}_1) = e(w', \tilde{h})$
		For all signatures σ_1 stored for the same basename:
		If $Link(gmpk, \sigma, Ch, \sigma_1, Ch, bsn)$ returns 0:
		Verify the validity of π_3 then **accept**
		Otherwise **reject**

Figure 6.4: PAYGO validation protocol.

2. TO and E respectively generate the following sets of keys $\{sk_{P_j}^{TO}\}_{j=1}^n$ and $\{sk_{P_j}^{E}\}_{j=1}^n$, where n denotes the maximum number of tags to be generated in a specific period of time (the sets of keys are generated monthly for example). They also generates the corresponding public keys $\{pk_{P_j}^{TO}\}_{j=1}^n = \{X_1^{sk_{P_j}^{TO}}\}_{j=1}^n$, and $\{pk_{P_j}^{E}\}_{j=1}^n = \{X_1^{sk_{P_j}^{E}}\}_{j=1}^n$.

3. To compute the basename for time period P_j, TO sends E his share of the basename computation $X_{TO} = X_1^{sk_{P_j}^{TO}}$, as well as the signature of knowledge $\pi_j^{TO} = SoK\{\alpha : X_{TO} = X_1^\alpha\}$. Upon receiving X_{TO}, E appends its own secret key as follows: $X_{P_j} = X_{TO}^{sk_{P_j}^{E}}$. The final basename bsn is defined as $bsn = X_{P_j}$. E also generates the signature of knowledge π_j^E, where $\pi_j^E = SoK\{\alpha : X_{P_j} = X_{TO}^\alpha\}$. The signatures of knowledge bind each entity to their part of the basename computation, proving that the values were computed as intended.

Generating the basename in this manner ensures that during the *revocation* process, each entity will be able to retrieve the public commitment of the concerned $\mathscr{SE}$, by using the multiplicative inverse of their respective keys (see Section 6.3.5).

6.3.4.2 Validation

As depicted in Figure 6.5, a user validates his pass by generating an anonymous signature using the credentials obtained in Section 6.3.3. Upon receiving the challenge and basename from the

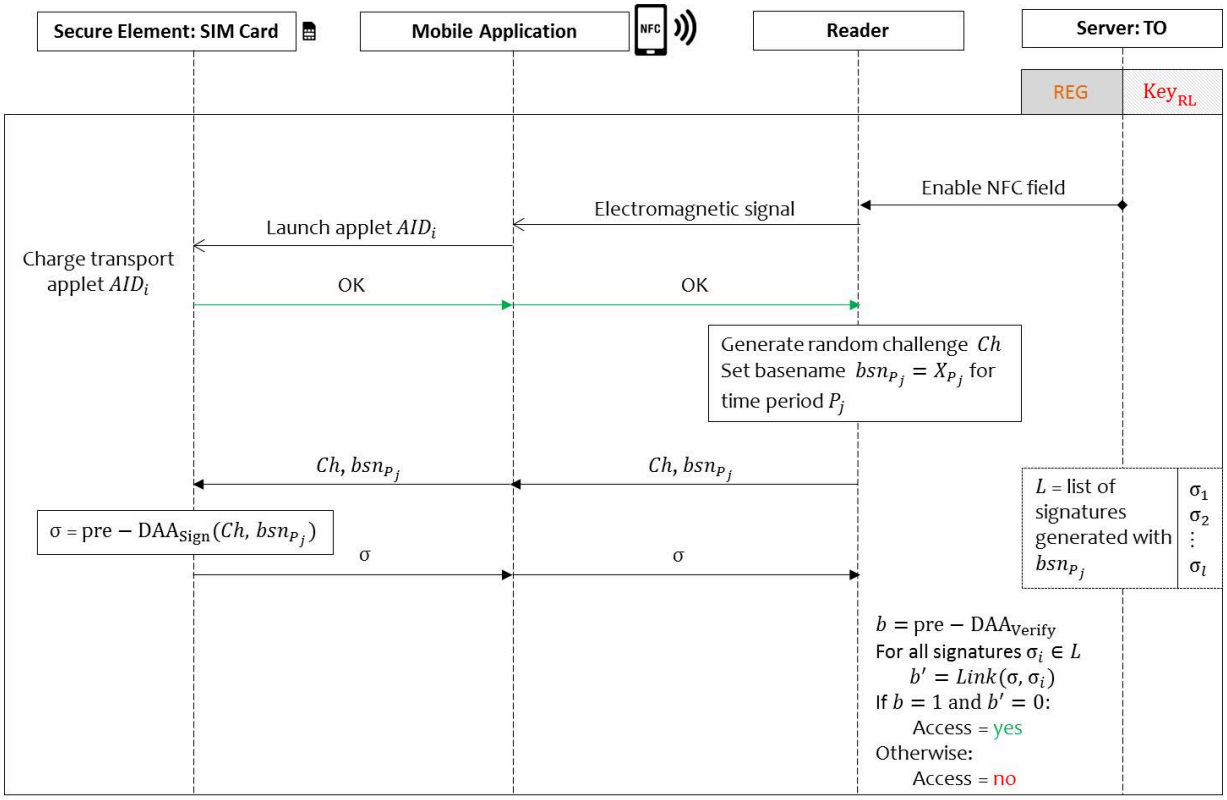


Figure 6.5: Overview of the PAYGO validation phase without revocation.

reader $\mathcal{V}$, $\mathcal{SE}$ generates a signature of knowledge on message Ch with respect to bsn. The result is an anonymous and basename-dependent linkable signature, which $\mathcal{V}$ is able to verify without lifting the anonymity of U. The detailed description of the *Validation* protocol is presented in Figure 6.4. The *validation* phase includes the additional *Link* function of the pre-DAA scheme (defined in Section 4.3.8), which allows to link two signatures to the same user with respect to the same basename. The *Link* function is used to enforce the anti-passback property.

6.3.4.3 Anti-Passback property

The anti-passback property of PAYGO denies access to a user who validates his pass twice during a short time period determined by the basename. For example, the reader should not grant access if two users attempt to use the same transport pass consecutively. After each validation, the reader ($\mathcal{V}$) stores the signature σ_1 for basename bsn (for time period P_j). If a user generates a signature σ_2 for the same bsn (during time period P_j), $\mathcal{V}$ detects passback by running $Link$ on σ_1, σ_2 and bsn. Indeed, the resulting tags T and T' will be the same for both signatures. For time period P_{j+1}, the basename is renewed, and the signatures generated by the same user are no longer linkable. The frequent and timely update of bsn is crucial to the overall untraceability of honest users on the transport network. Their specific generation and management should therefore be optimized accordingly by transport authorities.

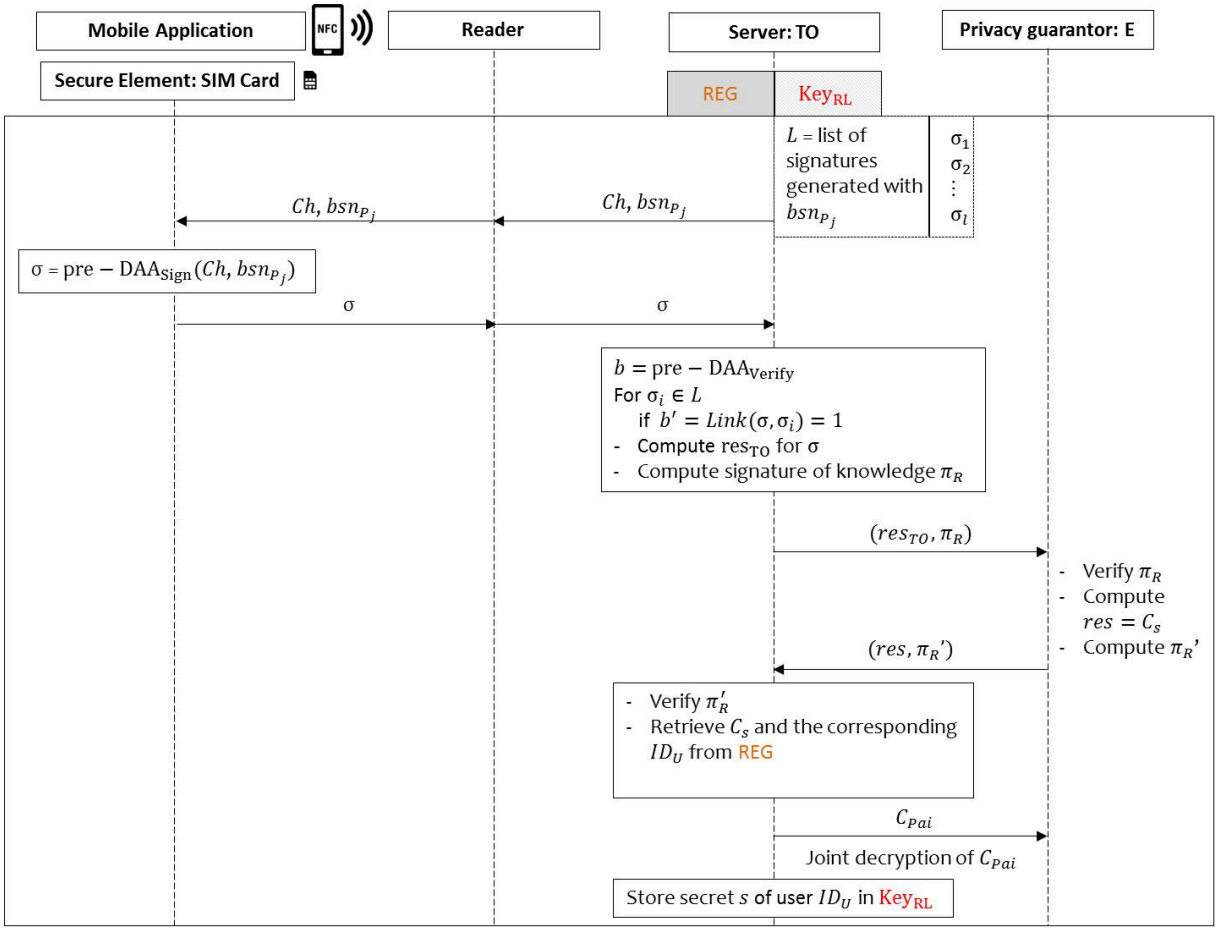


Figure 6.6: Overview of the PAYGO validation phase with revocation.

6.3.5 Revocation

The *revocation* phase allows the transport authority to revoke a user's credentials, upon fraud or clone detection. The revocation process is managed by the transport operator (TO) and the external privacy guarantor (E). Both maintain a secret key-based revocation list Key_{RL}, which stores the secret keys of revoked users. This phase is divided into two sub-phases, namely *Identify* and *Revoke*.

6.3.5.1 Identify.

The first step in the *revocation* procedure allows the transport authority to retrieve the identifier ID_U of the user whose pass is to be revoked. The basename generation procedure described in Section 6.3.4.1 ensures that revocation authorities, namely TO and E, must collaborate in order to jointly retrieve the identifier of a pass which has issued a given signature. They proceed as follows:

- TO detects two validations which are linked to the same user at different stations in the same time period P_j. This indicates that a clone of the user's pass exists, and triggers the identification process.

- For the concerned signature $\sigma_R = (w_R, w_R', \pi_{3R}, c_R, T_R)$, TO computes $res_{TO} = T_R^{(sk_{P_j}^{TO})^{-1}}$, as well as a signature of knowledge $\pi_R = SoK\{\alpha : res_{TO} = T_R^{\alpha} \wedge (pk_{P_j}^{TO})^{\alpha} = X_1\}$.
- Upon receiving (res_{TO}, π_R) E verifies π_R, and computes $res = res_{TO}^{(sk_{P_j}^{E})^{-1}}$, as well a signature of knowledge $\pi_R' = SoK\{\alpha : res = res_{TO}^{\alpha} \wedge (pk_{P_j}^{E})^{\alpha} = X_1\}$.
- The final result res corresponds to the value $X_1^s = C_s$.
- TO retrieves C_s and the corresponding ID_U from REG.

6.3.5.2 Revoke.

Upon recovering ID_U, TO and E jointly decrypt the associated Paillier ciphertext C_{Pai} using a joint Paillier decryption method, such as the one presented in [HMR+19]. They retrieve the corresponding secret s. TO stores s in Key_{RL}. Henceforth, the verification step of the validation procedure described in Figure 6.4 is modified to include revocation as follows:

1. **Check** that $w \neq 1$ and $T \neq 1$
2. **Verify** that $e(w, \tilde{X}_0) \cdot e(c, \tilde{X}_1) = e(w', \tilde{h})$
3. **For all** $s^i \in Key_{RL}$, **compute** $T_{temp} = X_{P_j}^{s^i}$:
   If $T = T_{temp}$, **reject**
4. **For all** signatures σ_1 stored for the same basename:
   If $Link(gmpk, \sigma, Ch, \sigma_1, Ch, bsn)$ returns 0:

      Verify the validity of π_3 then **accept**

5. Otherwise **reject**

An overview of the validation phase with revocation is depicted in Figure 6.6.

6.4 Security Analysis

The security of PAYGO mainly relies on the security of the underlying pre-DAA scheme. In this section, we demonstrate how PAYGO satisfies the required security properties defined in Section 6.1.3.2.

- **Consistency.** This property can be verified by inspection. Indeed, a valid credential is a randomized version (w, w') of the group signing key (u, u') obtained during the *registration* process. In the validation process, the reader grants access if the following conditions are met: (1) (w, w') is a valid Pointcheval, Sanders (PS) signature (see Section 4.3) on a secret s known to the SIM card (i.e. the group signing key is valid and was obtained during a previous registration process with the transport operator); (2) The same secret is used to generate the tag T. By verifying the previous two conditions, the reader grants access with probability 1.

- **Unforgeability.** In order to forge a signature, a defrauder $\mathscr{A}$ must be able to generate a signature σ that cannot be traced back to the secret key that was queried in a previous *registration* protocol. In Section 4.4.2, we prove that such a forger can be used to construct a second forger against the security of the PS signature. The PS signature being proven secure under the q-MSDH assumption ([PS18], Theorem 10), such a forger succeeds only with negligible probability.

- **Anonymity.** PAYGO satisfies this property based on the *anonymity* property of the pre-DAA scheme. Indeed, given a pass validation and two identifiers ID_0 and ID_1, an adversary $\mathscr{A}$ (which can be the transport operator attempting to break the privacy of users) that is able to distinguish whether the validation was generated by ID_0 and ID_1, can be leveraged to build a forger which solves the external Diffie-Hellman (XDH) problem with non-negligible probability. The anonymity property is perfectly modeled by the *anonymity* property of the underlying pre-DAA scheme, which was proven to hold in Section 4.4.1.

- **Unlinkability.** The first requirement of this property states that it should not be possible for an adversary $\mathscr{A}$ to link two validations generated by the same pass. In PAYGO, validations generated by the same pass are linked using the basename bsn. As described in Section 6.3.4.1, the basename generation process ensures that two passes generated at different time slots $P_i \neq P_j$, the signatures cannot be linked. Indeed, the basename is changed for each time slot, and the tag T_i generated on bsn_{P_i} is different than the tag T_j generated on bsn_{P_j}. This ensures that two validations by an honest user cannot be linked.
  The second requirement states that it should not be possible for an adversary $\mathscr{A}$ to retrieve the identifier from a valid signature. By randomizing its group signing key (u, u') prior to generating a signature, we ensure that no entity, even when they collude with the transport operator, can decide whether a given pass has generated the signature or not.

- **Anti-passback.** A defrauder $\mathscr{A}$ validating the same pass consecutively will do so in the same time slot P_j; As described in Section 6.3.4.3, this results in $\mathscr{A}$ generating two signatures using the same key sk_i with respect t the same basename bsn. The Link function will therefore determine with probability 1 that the two signatures are linked. This property is therefore ensured with overwhelming probability.

6.5 Implementation and Evaluation

In this section, we present the specifications of the setup used in the implementation of PAYGO.

6.5.1 Implementation specifications

6.5.1.1 Setup and communication model

For our implementation, we use a Global Platform-compliant SIM card, which is embedded in a Samsung Galaxy S5 NFC-enabled smartphone. The communication between the reader and the

SIM card (via the smartphone) follows the request/response model, whereby the smart card plays the passive role. It remains passive waiting for a command request from the reader via the client (host) application, namely the PAYGO application on the smartphone. Upon receiving a command Application Protocol Data Unit (APDU) containing a command from the reader (for example to initiate the PAYGO process), the card executes the instructions specified in the command and replies with a response APDU. The smartphone is used as an NFC relay and to help trigger and store some precomputed values, but it does not participate in the signature generation process.
The SIM card is a Java Card 2.2.2 Oberthur smart card with a 44MHz ARM processor. The card has 10kB of volatile fast memory (RAM) and 450kB of persistent memory (EEPROM). The only specificity is that the SIM card supports mathematical Application Programming Interfaces (API) for modular arithmetic and arithmetic operations on elliptic curve. Through its own random number generator, the card can also generate its own secret key s_1. The reader used to emulate an NFC-enabled reader at a public transport station is an HID Omnikey contactless reader.

6.5.1.2 Precomputations

Commands that provide the instructions to generate some precomputed values are triggered by the PAYGO application on the smartphone, which sends a command APDU containing the instructions to initiate precomputation to the SIM whilst the mobile is turned on. The precomputations consists in the SIM generating multiple randomization values $l \in \mathbb{Z}_p$ and the corresponding (w, w), which are subsequently stored for latter validations.

6.5.2 Performances

Table 6.1 displays the performances obtained from the implementation of our PAYGO protocol. The timings are obtained on an average of 100 tests.

<table>
<tr><td colspan="3">Off-line computation</td></tr>
<tr><td colspan="2">Battery On
(238-264) 253 ms</td><td>Battery Off
(753-831) 798 ms</td></tr>
<tr><td colspan="3">On-line computation</td></tr>
<tr><td colspan="2">Signature generation (SIM card)</td><td>Signature verification (Reader)</td></tr>
<tr><td>Battery on
(153-167) 162 ms</td><td>Battery Off
(450-472) 462 ms</td><td>(4-16) 11 ms</td></tr>
<tr><td colspan="3">Total On-line computation</td></tr>
<tr><td colspan="2">Battery On</td><td>Battery Off</td></tr>
<tr><td colspan="2">(157-183) 173 ms</td><td>(455-487) 471 ms</td></tr>
</table>

Table 6.1: PAYGO *Validation* phase timing (min-max) average (ms).

6.5.3 Evaluation

The implementation is split between an *off-line*, phase which allows the SIM card to perform some precomputation (as explained in Section 6.5.1.2), and an *on-line* phase for the computations that rely on the values Ch and bsn sent by the terminal. In addition, we include the timings for the cases where a SIM has to perform pass validation without being powered by the smartphone ("Battery off"). In such cases, the SIM card performs in "downgraded" mode, due to it being powered by the NFC reader. In such cases where the smartphone is turned off, the precomputations are performed by the card upon receiving the command APDU from the reader, inducing longer computation timings. Such occurrences are however rare, and the timings are only included for completeness purposes. This however demonstrate that a SIM card can authenticate in a standalone manner (without delegating any signature computation to the mobile phone).
The timings obtained for our pre-DAA scheme, show that a computationally constrained SIM card can generate a signature in less than 200 milliseconds. The maximum timings for signature validation by mobile transit passes is 300 milliseconds [GSM12]. PAYGO is therefore complies with the efficiency requirements.

6.6 Conclusion

In this chapter, we introduce PAYGO, a new privacy-preserving transit pass protocol, that preserves the anonymity of users on public transport networks. The protocol is based on our pre-DAA scheme introduced in Chapter 4, and is efficient enough for a SIM card to solely execute the secure element side of the authentication protocol. PAYGO complies with the timing constraints of an NFC-enabled authentication protocol, that state that an NFC-enabled pass validation must be performed in under 300 milliseconds. The user-controlled traceability of the underlying pre-DAA scheme enforces user liability by preventing a user to use its pass twice consecutively (anti-passback property). We demonstrate with this new construction that we build privacy-preserving authentication schemes from Direct Anonymous Attestation, that are efficient and suitable for applications where devices are not equipped with high-end hardware modules.
This chapter concludes the first part of this thesis, where we study the security and privacy requirements for authentication and access control in environments where user and data privacy are critical. We develop privacy-protocols for such environments based on our pre-DAA constructions, whilst respecting the efficiency and scalability requirements of such use cases.

Part III

Collective Device Attestation

CHAPTER 7

A SWARM ATTESTATION PROTOCOL WITH SEQUENTIAL DETECTION

A swarm attestation protocol runs between the trusted verifier and a group of untrusted devices. Existing swarm attestation protocols are vulnerable to denial of service attacks on the verifier, who receives a single attestation response to validate, with no mechanism to detect rogue elements that might potentially aggregate an erroneous individual attestation.
In this chapter, we aim to extend the notion of swarm attestation to include a sequential detection mechanism, whereby a trusted verifier is able to not only validate the integrity of each device's internal software state in an efficient manner, he is also able to detect the identifier of devices that might forward an erroneous attestation response. Our swarm attestation protocol is the first to enable individual detection from an aggregate attestation response.

The results of this chapter have been published in [DLLT20].

Contents

7.1 Introduction

We presented the concept of swarm attestation in Section 3.3. The general architecture of a swarm attestation protocol, as illustrated in Figure 7.1, is comprised of the network operator, who deploys devices in the field, the group of devices performing a specific task, and the verifier. Nodes in the swarm generate their individual attestation responses, which are in turn accumulated into a single attestation response. The topology of the mesh network plays a crucial role in the effectiveness of the attestation protocol. Notably, during the attestation process, a spanning tree is generated over the network, allowing each node to propagate their response in the tree via their parent node. The root of the spanning tree is directly linked to the verifier, who collects a In-network aggregation schemes does not allow manipulation detection. The result only stating retrospectively that the aggregate message was manipulated. Indeed, existing aggregation schemes based on digital signatures (respectively message authentication codes) do not allow the extraction of an individual signature (respectively tag) from the aggregate result. Such attacks may therefore trigger the verification process indefinitely, inducing a denial of service attack on the verifier. The goal of this chapter is to introduce a new swarm attestation protocol CoRA, which enables to sequentially detect the origin of an erroneous aggregate response.

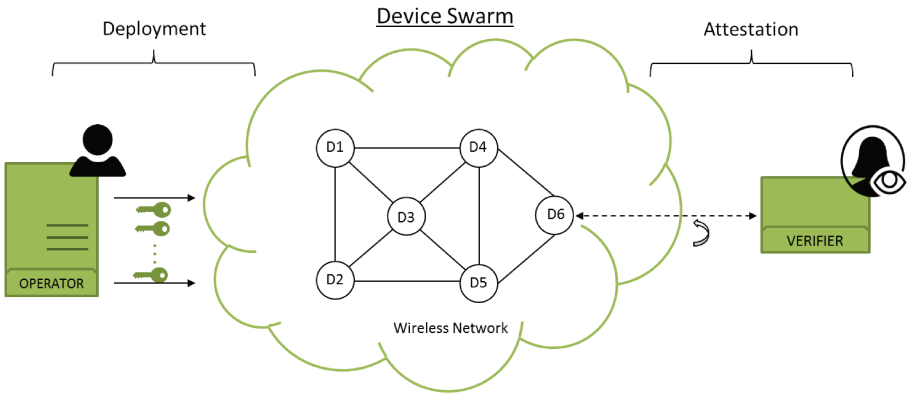


Figure 7.1: Swarm Attestation Overview

7.2 System Model and Assumptions

7.2.1 System model

In the architecture presented in Figure 7.1, a swarm attestation protocol is divided into two main phases, namely the deployment phase and the attestation phase. During the former, the operator $\mathcal{O}$ deploys devices in the field with a unique identifier ID_i for device D_i, and a secret key sk_i. The verifier $\mathcal{V}$ initiates the attestation phase by generating a challenge Ch, as part of the attestation request for a device A_s in the network. The attestation process will create a spanning tree rooted at A_s, thus enabling an efficient and scalable aggregation of the individual attestation reports. A_s then propagates the request down the spanning tree. Upon each node generating, authenticating, and aggregating their attestation reports, the final aggregated response reaches A_s, who aggregates its own response, before forwarding the result to $\mathcal{V}$. The verifier, who is in possession of the expected internal state of each device, is thus able to verify the integrity of the entire network. For simplicity, we assume in the remaining of the chapter that $\mathcal{V}$ and $\mathcal{O}$ are either the same entity, or are managed by the same entity.

7.2.2 Threat and attack model

We consider a network of low-end, heterogeneous, embedded devices, communicating over a wireless mesh network. Devices in the network have limited computational power and storage capacity. In the infrastructure of the swarm attestation protocol, we assume the operator $\mathcal{O}$ to be the network administrator, who initially deploys devices in the field. The deployment phase is only executed once, and consists in each device D_i being initialized with a secret symmetric key sk_i shared with $\mathcal{O}$ (and the verifier $\mathcal{V}$), as well as a unique identifier ID_i, which can be the IP address of wireless devices for example. In order to verify the integrity of the internal software state of every device in the network, the trusted verifier $\mathcal{V}$ periodically engages in the attestation protocol with the device swarm. At the beginning of the attestation phase, a spanning tree of all nodes is formed, thus facilitating the aggregation process. At the end of the attestation phase, $\mathcal{V}$ collects a single attestation report, which guarantees the integrity (or lack thereof) of the entire swarm. The verifier possesses the list of expected software state, hence allowing him to verify the validity of the final attestation response in correlation with the expected values.

7.2.2.1 Hardware Assumptions.

For the construction of our swarm attestation protocol, we use the model of Francillon et al. [FNRT12], whereby we assume that each device is equipped with the minimal hardware requirements for a swarm attestation protocol : a Read-Only Memory (ROM) to store the protocol codes, and a Memory Protection Unit (MPU) that stores cryptographic keys and controls access to them. Each device is deployed with a secret symmetric key sk_i shared with the owner/verifier, and a unique identifier ID_i. Each device has the capacity to compute a MAC tag in prime order groups. Each node is also capable of computing a collision-resistant hash function $\mathcal{H}$.

7.2.2.2 Adversary Model

We consider an adversary $\mathscr{A}$, who has full access to the communication channel (Dolev-Yao model [DY83]). As such, $\mathscr{A}$ can eavesdrop, modify, insert, and drop messages exchanged between devices. $\mathscr{A}$ can also capture nodes and access and modify their software state. Security against physical adversaries (who are able to extract cryptographic keys), cannot however be guaranteed [ISTZ16]. In contrast, we consider the following threat model: an adversary $\mathscr{A}$ who has compromised the networks, and potentially captured a number of devices (up to $n-1$ devices in a network comprising n devices), should still be unable to forge a valid aggregate attestation for the remaining nodes. With regards to Denial of Service (DoS) attacks, existing attestation protocols consider attacks where devices are rendered unavailable in the network, thus preventing the successful completion of the process. Such attacks cannot be prevented, as it is indeed impossible to guarantee availability for an adversary $\mathscr{A}$ capable of dropping all messages. In this work, we also consider DoS attacks against the verifier. Indeed, $\mathscr{A}$ is able to continuously aggregate an erroneous message in the final response, triggering a failed verification process each time. We mitigate against such attacks by providing an efficient detection mechanism in CoRA.

7.2.3 Security model and assumptions

An attestation protocol starts with $\mathscr{V}$ generating a challenge c for prover P. P computes an attestation response r on its internal state and the challenge c, before returning r to $\mathscr{V}$. Finally, $\mathscr{V}$ verifies that r is a valid response. During a swarm attestation process, $\mathscr{V}$ generates c for the entry node d_1. Node d_1 then propagates the attestation in the network, and finally, collects an attestation response r' from all remaining devices in the network to which it appends its own response before returning the final message to $\mathscr{V}$. In order to ensure the correctness of the verification step, it is assumed that $\mathscr{V}$ knows all the valid internal states for every prover. He is thus able to compare the received values with the expected values. In the remaining of this chapter, we assume that $\mathscr{V}$ and $\mathscr{O}$ share the secret key in order for $\mathscr{V}$ to effectively execute the verification function. The attestation process is bounded by t_{exp}, which is the expected maximum amount of time estimated to execute the attestation procedure. t_{exp} is a function of the total number of nodes in the group, the time required to compute a single attestation, the transmission time for an intermediary response, and the aggregation time.

7.2.3.1 Remote attestation security model

A remote attestation scheme is comprised of the following algorithms:

- $\mathsf{Setup}(1^\lambda)$: a probabilistic algorithm that takes as input a security parameter λ, and outputs a long-term attestation key sk;

- $\mathsf{Challenge}(N, t_{exp})$: a probabilistic algorithm that takes as input a random nonce N and the timeframe t_{exp}, and generates a challenge c;

- Attest(sk,S): a deterministic algorithm that takes as input the attestation key sk and the internal state S, and produces an attestation response r;

- Verify(sk,S,r): a deterministic algorithm that takes as input the attestation key sk, the timeframe t_{exp} and the response r. It returns 1 iff Attest(sk,S) = r, and 0 otherwise.

Prior to starting the attestation process, $\mathcal{O}$ runs Setup(1^λ) and generates the attestation key sk. For a swarm attestation protocol, $sk = \{sk_1,...,sk_n\}$ where all the devices $d_i \in \{d_1,...,d_n\}$ are deployed with their individual attestation key sk_i.

[FNRT12] introduced the first formalized definition of a remote attestation protocol. The security definition for a remote attestation protocol is, namely FORGERY, is formalized by a game between a challenger $\mathscr{C}$ and an adversary $\mathscr{A}$. So far, no formalized definition security exists for swarm attestation protocols. We slightly modify the model of Francillon et al. to adapt their definition to swarm attestation protocols.

A swarm attestation protocol is secure if it unforgeable against an adaptive adversary. Unforgeability here means that the adversary may adaptively corrupt up to $n-1$ provers and learn their secret keys, and still not be able to forge an attestation response for n provers. Let $\{S_1,...,S_n\}$ be the initial states of a set of honest provers $\{\mathscr{P}_1,...,\mathscr{P}_n\}$. We define the following game between a challenger $\mathscr{C}$, and an adversary $\mathscr{A}$ that attempts to output a forgery on a subset of $\{S_1,...,S_n\}$:

$\mathsf{Exp}^{\mathsf{Forgery}}_{\mathscr{A},\mathscr{SA}}(1^\lambda)$:

1. $\mathscr{C}$ runs $\{k_1,...,k_n\} \leftarrow$ Setup(1^λ) and sends $c \leftarrow$ Challenge(N, t_{exp}) to d_1;

2. $\mathscr{A}$ has oracle access to the attestation function **Attest** in the form of $\mathcal{O}_{\mathsf{Attest}}$, and adaptively queries $\mathcal{O}_{\mathsf{Attest}}$ on the states $\{S_1,...,S_{qA}\}$ where $qA < n$ is the maximum number of attestation queries. For each S_i, $\mathscr{A}$ receives $\alpha_i = \mathcal{O}_{\mathsf{Attest}}(k_i, S_i)$;

3. Eventually, $\mathscr{A}$ outputs a response α;

4. Output 1 iff Verify(k,$\{S_1,...,S_n\}$,α) = 1 and there exists a pair (k^*_j, s^*_j) such that $\mathscr{A}$ has never queried Attest(k^*_j, s^*_j)

This security game is analogue to the existential forgery game for an aggregate MAC scheme as described in Section 2.4.2.3. Adversary $\mathscr{A}$ winning probability is defined by his success probability in convincing $\mathscr{V}$ that α is a valid attestation response stating that the swarm $\{d_1,...,d_n\}$ is in a trusted state.

Definition 7.1. (Attestation Protocol Security). A swarm attestation protocol $\mathscr{SA}$ =(Setup, Challenge, Attest, Verify) is secure against adaptive forgery if there exists a negligible function ν such that for any probabilistic polynomial-time adversary $\mathscr{A}$, Pr[$\mathsf{Exp}^{\mathsf{Forgery}}_{\mathscr{A},\mathscr{SA}}(1^\lambda) = 1$]$\leq \nu(\lambda)$.

7.2.4 Efficient In-Network Aggregation

Aggregation methods allow a collection of devices to securely and collaboratively compute the aggregated response corresponding to their respective attestation reports. The result of said function is thus forwarded to the verifier, generating extremely low communication overhead in the process. This concept, known as *in-network aggregation* [CPS06], considerably reduces communication overhead, and provides a highly scalable data collection mechanism. Existing aggregation methods that are linear in the number of nodes, and that can be executed in a single round [ACI+16], are based on aggregate digital signatures [BGLS03]. However, such constructions are extremely costly due to the use of pairing-based cryptography.
In IoT networks, it is mostly the case that the network administrator plays the role of both the operator and the verifier (or both are operated by the same entity, i.e. the owner). Therefore, a more efficient alternative to the use of aggregate signatures, is the use of aggregate algebraic MACs.

7.3 Aggregate Algebraic $\mathsf{MAC}_{\mathsf{BLS}}$

The underlying scheme for CoRA is an aggregate algebraic MAC, named aggregate $\mathsf{MAC}_{\mathsf{BLS}}$, and derived from the Boneh, Lynn, Shacham (BLS) signature scheme [BLS01]. Aggregate MAC schemes, as defined in Chapter 2, allow a set of n users to generate n tags on n potentially different messages, and aggregate the result into a single tag of the same size as an individual tag. The verification process, which is linear in the number of users, attests of the validity of all tags in the final aggregate. We define a new aggregate algebraic MAC, where the corresponding tags are group elements, as opposed to block ciphers or hash functions. Tags are single group elements of size 256-bit (for example elliptic curve group elements). The output size of secure hash functions being comparable to the output size of an algebraic MAC, the use of algebraic MACs does not induce additional space complexity.

7.3.1 $\mathsf{MAC}_{\mathsf{BLS}}$ construction.

The $\mathsf{MAC}_{\mathsf{BLS}}$ scheme comprises the following algorithms:

- $\mathsf{Setup}(1^\lambda)$: creates the public parameters $pp = (\mathbb{G}, q, g, \mathscr{H})$ where $\mathbb{G}$ is a group of prime order q of size λ, where CDH is hard. $g \xleftarrow{\$} \mathbb{G}$ is a random generator of $\mathbb{G}$. $\mathscr{H}_1 : \{0,1\}^* \to \mathbb{G}$ is a hash function, modeled as a random oracle in the security analysis.
- $\mathsf{KeyGen}(1^\lambda)$: selects a random element $x_i \xleftarrow{\$} \mathbb{Z}_q$, and sets the secret key $sk_i = x_i$. The algorithm also outputs the public parameter $ipp = X_i$, where $X_i = g^{x_i}$.
- $\mathsf{Mac}(pp, sk_i, m_i)$: computes $h_i \leftarrow \mathscr{H}_1(m_i)$, and generates the tag $\tau_i = h_i^{x_i}$.
- $\mathsf{Verify}(pp, sk_i, pk_i, m_i, \tau_i)$ computes $h_i = \mathscr{H}_1(m_i)$, and accepts if $\tau_i = h_i^{x_i}$.

Theorem 7.1. *Our $\mathsf{MAC}_{\mathsf{BLS}}$ is UF-CMVA secure under the CDH assumption in the random oracle model.*

The proof of Theorem 7.1 is provided in Section 7.3.3.

7.3.2 Aggregate $\mathsf{MAC}_{\mathsf{BLS}}$ construction.

Based on the $\mathsf{MAC}_{\mathsf{BLS}}$ scheme construction introduced above, we define the aggregate MAC scheme in groups of prime order. The aggregation function AggMac is public and unkeyed, namely any node is able to aggregate its own tag without a secret key (as opposed to the MAC computation step). The aggregate $\mathsf{MAC}_{\mathsf{BLS}}$ scheme is the $\mathsf{MAC}_{\mathsf{BLS}}$ scheme with the following additional algorithms:

- AggMac($\{\tau_i\}_{i=1}^n$): Given a set of tags $\{\tau_i\}_{i=1}^n$, outputs the aggregate tag $\tau = \prod_{i=1}^n \tau_i$.

- AggVerify($pp, \{(sk_i, pk_i)\}_{i=1}^n, \{m_i\}_{i=1}^n, \tau$): for $1 \leq i \leq n$, computes $h_i = \mathscr{H}_1(m_i)$. Checks if $\tau = \prod_{i=1}^n h_i^{x_i}$. Returns accept if true, and reject otherwise.

Theorem 7.2. *Aggregate MAC_{BLS} is unforgeable provided that the underlying MAC_{BLS} scheme is unforgeable.*

As described in Section 2.4.2.3, he security definition of unforgeability for aggregate algebraic MACs is that of existential unforgeability under chosen message attack (EUF-CMA). This definition illustrates the fact that an adversary who has access to at most $n-1$ secret keys, is still unable to output a valid aggregate tag by n users [BGLS03].
The proof of Theorem 7.1 is provided in Section 7.3.4.

7.3.3 Security proofs of $\mathsf{MAC}_{\mathsf{BLS}}$

In this section, we prove Theorem 7.1, which states that $\mathsf{MAC}_{\mathsf{BLS}}$ is UF-CMVA in the random oracle model. Let q_H be the maximum number of queries an algorithm $\mathscr{A}$ can make to $\mathscr{O}_{\text{Hash}}$, q_M the maximum number of queries to $\mathscr{O}_{\text{MAC}}$, and q_V the maximum number of queries to $\mathscr{O}_{\text{Verify}}$. Assume an algorithm $\mathscr{A}$ $(t, q_H, q_M, q_V, \epsilon)$−breaks the UF-CMVA security of $\mathsf{MAC}_{\mathsf{BLS}}$. We use $\mathscr{A}$ to construct an algorithm $\mathscr{B}$ that (t', ϵ)−breaks computational Diffie-Hellman (CDH) on the group $\mathbb{G}$. Using a sequence of security experiments, we use the forger $\mathscr{A}$ to build the algorithm $\mathscr{B}$ that breaks CDH on $\mathbb{G}$.

Setup. $\mathscr{B}$ is given the challenge (g, g^a, g^b). $\mathbb{G}$ being a group where DDH is easy but CDH remains hard (denoted gap Diffie-Hellman (GDH) group), $\mathscr{B}$ also has access to a DDH oracle $\mathscr{O}_{\text{DDH}}$ that outputs 1 if an argument of the form (g, g^a, h, h^b) is a valid Diffie-Hellman tuple. $\mathscr{B}$ maintains a list (h_i, τ_i) for all hash and MAC queries. $\mathscr{A}$ can request a hash on any message of its choice, to which $\mathscr{B}$ responds with the corresponding h_i. Similarly, $\mathscr{A}$ can request a tag on any message of its choice, and receives the corresponding τ_i where $i \in \{1, ..., l\}$ and $l \leq q_M$ is the number of requests to $\mathscr{O}_{\text{MAC}}$. $\mathscr{B}$ sets the operator's parameters $X = g^a$, thus implicitly setting $sk = a$.

For $1 \leq i \leq q_H$, $\mathscr{A}$ picks a random $i^* \xleftarrow{\$} \{1, ..., q_H\}$. $\mathscr{A}$ then selects a random $r_i \xleftarrow{\$} \mathbb{Z}_q^*$ for $1 \leq i \leq q_H$, sets $h_i \leftarrow (g^b) g^{r_{i^*}}$ for $i = i^*$, and $h_i \leftarrow g^{r_i}$ otherwise. If $i \neq i^*$, it sets $\tau_i \leftarrow (g^a)^{r_i}$. If $i = i^*$, it sets $\tau_{i^*} = \star$ which is a placeholder value.

Queries. When $\mathscr{A}$ requests a hash on m_i, $\mathscr{B}$ outputs h_i. When $\mathscr{A}$ requests a tag on m_i, if $i \neq i^*$ $\mathscr{B}$ outputs τ_i. Otherwise if $i = i^*$, $\mathscr{B}$ declares failure and aborts. $\mathscr{A}$ may also make verification queries on input (m_i, τ_i). $\mathscr{B}$ then uses its DDH oracle $\mathscr{O}_{\text{DDH}}$, and outputs the result obtained from $\mathscr{O}_{\text{DDH}}$ on the tuple (g, g^a, h_i, τ_i).

Response. At some point, $\mathscr{A}$ outputs a forgery (m^*, τ^*) such that m was never queried to $\mathscr{O}_{\text{MAC}}$, and $\mathsf{Verify}(sk, m, \tau) = 1$.

- The probability that $\mathscr{B}$ aborts is equal to $1/q_M$, which is inverse polynomial. We can reduce this value by repeating the experiment a polynomial number of times.
- The r_i's are selected at random, therefore h_i's are uniformly distributed in $\mathbb{G}$, making $\mathscr{O}_{\text{Hash}}$ a random oracle. Moreover, if $i \neq i^*$, the tags τ_i are all valid. If $\mathscr{B}$ does not abort, the simulation for $\mathscr{A}$ is indistinguishable from a real execution of the MAC algorithms.
- If $\mathscr{A}$ successfully outputs a forgery (m^*, τ^*) and $\mathscr{B}$ does not abort, $h^* = h_{i^*} = (g^b)g^{r_{i^*}}$. Therefore $h_{i^*}^a = (g^{ab}g^{r_{i^*}})$. $\mathscr{B}$ retrieves $g^{ab} = \frac{\tau^*}{(g^a)^{r_{i^*}}}$. $\mathscr{B}$ outputs the valid answer to the CDH challenge.
- $Adv_{\mathscr{A}}^{\text{UF-CMVA}}(1^\lambda) = \epsilon/q_M$. If we repeat the experiment k times, the probability that algorithm $\mathscr{B}$ outputs a valid answer to the CDH challenge is equal to $(\frac{\epsilon}{q_M})^k$ which is non-negligible.

7.3.4 Security proof of aggregate $\mathsf{MAC}_{\mathsf{BLS}}$

In this section, we give the proof by reduction of Theorem 7.2 in the random oracle model. We consider the setting where the reduction maintains a list KeyList of all public keys attached to each secret key sk_i.

Let $\mathscr{A}$ be an adversary against the UF-CMVA property of aggregate $\mathsf{MAC}_{\mathsf{BLS}}$. Using $\mathscr{A}$ as a subroutine, we construct a reduction $\mathscr{B}$ against the EUF-CMVA property of the $\mathsf{MAC}_{\mathsf{BLS}}$ scheme run by a challenger $\mathscr{C}$. $\mathscr{B}$ receives the system public parameters $pp = (\mathbb{G}, q, g)$, and the public key pk^* for an unknown identity $X_{i^*} = g^{x_{i^*}}$ where $i^* \in \{1, ..., l\}$ and $l = poly(\lambda)$. $\mathscr{B}$ has access to two oracles: an oracle $\mathscr{O}_{\mathsf{MAC}_{\mathsf{BLS}}, sk^*}$ (that we will refer to as $\mathscr{O}_{\mathsf{Mac}}$ to simplify notations) for the unknown key $x^* = x_{i^*}$, and the oracle $\mathscr{O}_{\mathsf{Verify}}$ that checks the validity of any message/tag pair. $\mathscr{C}$ initializes an empty list $\mathscr{M}$ that will store the subsequent messages from $\mathscr{O}_{\mathsf{Mac}}$ queries.

1. $\mathscr{B}$ chooses $i^* \xleftarrow{\$} \{1, ..., t\}$;
2. For $i = 1$ to t:
   - If $i \neq i^*$, select $r_i \xleftarrow{\$} \mathbb{Z}_q^*$, and set $h_i \leftarrow g^{r_i}$. Choose $x_i \xleftarrow{\$} \mathbb{Z}_q$. Set $X_i = g^{x_i}$ and add $(i, pk_i = X_i, sk_i = x_i)$ to KeyList. $ipp = \{pk_1, ..., pk_t\}$.
   - If $i = i^*$, do nothing but implicitely set $sk_{i^*} = (sk^*)$.
3. $\mathscr{B}$ runs $\mathscr{A}(pp, ipp)$ answering the queries as follows:

- $\underline{\mathsf{Hash}^*(m_i)}$:$\mathscr{B}$ uses its $\mathsf{MAC}_{\mathsf{BLS}}$ hash oracle $\mathscr{O}_{\mathrm{Hash}}$ to output the corresponding hash h_i.
- $\underline{\mathsf{AggMac}^*(\{m_1,...,m_q\})}$: $\mathscr{B}$ computes the aggregate tag τ_i using the known secret key sk^*. If there is $i \in \{1,...,q\}$ such that $i = i^*$, $\mathscr{B}$ queries its own oracle $\mathscr{O}_{\mathsf{Mac}}$ on m_i. Finally, $\mathscr{B}$ returns the τ_i.

4. At some point, $\mathscr{A}$ outputs $M = \{m_1,...,m_n\}$ and an aggregate tag τ. Let j be the first index such that $\mathscr{A}$ has never queried m_j to Mac^*, and $pk_j \neq pk^*$. If $j \neq i^*$ then $\mathscr{B}$ abort; Otherwise, proceed as follows:

   a) We therefore consider the case where $j = i^*$. $\mathscr{B}$ computes the tag τ^* from $\tau = \prod_{i=1}^{t} h_i^{x_i}$ the following way: $\tau^* = \frac{\tau}{\prod_{l=1,l\neq i^*}^{t} h_l^{x_l}}$.
   b) $\mathscr{B}$ has the secret keys corresponding to all identifiers that are not i^*, he is therefore able to compute a valid τ^*. When $j = i^*$, $\mathscr{B}$ simply queries $\mathsf{Mac}^*(m)$ to get the corresponding tag. Finally, $\mathscr{B}$ outputs (m_j, τ^*) as a valid forgery of a $\mathsf{MAC}_{\mathsf{BLS}}$ tag.

The proof of correctness follows from the following observations:

- The probability that $\mathscr{B}$ aborts is equal to $1/t$. If $\mathscr{B}$ does not abort, the simulation for $\mathscr{A}$ is indistinguishable from a real execution of the aggregate MAC algorithms.
- If $\mathscr{A}$ successfully outputs a forgery for the aggregate MAC scheme, and $\mathscr{B}$ does not abort, the assumption stipulates that $\mathscr{A}$ has never queried $\mathscr{B}$ on m_j. Therefore $\mathscr{B}$ has never queried $\mathsf{Mac}^*(m_j)$. The success of algorithm $\mathscr{A}$ means that $\tau = \prod_{i=1}^{t} \tau_i$ containing τ^*. The tag $\mathscr{B}$ outputs is therefore a valid forgery of a $\mathsf{MAC}_{\mathsf{BLS}}$ tag.

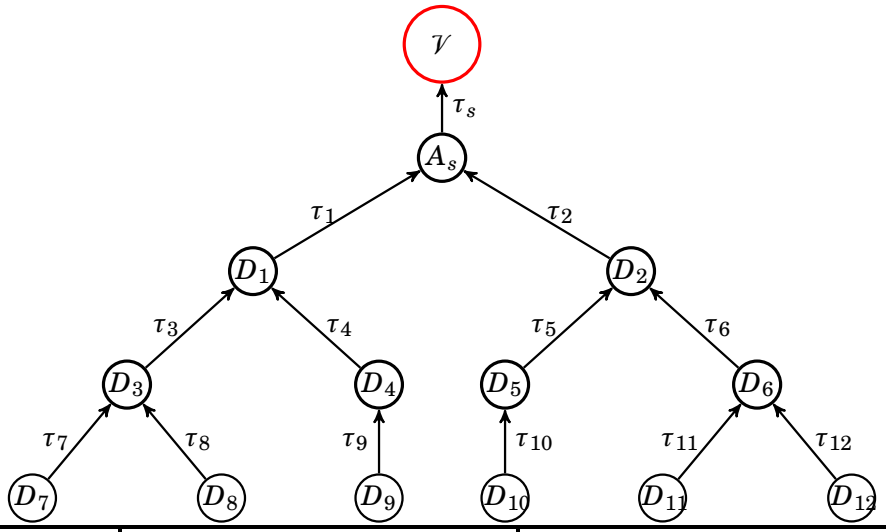
7.4 CoRA Protocol

7.4.1 Protocol construction

CoRA is a secure and scalable swarm attestation protocol, which consists of two phases, namely the *deployment phase* presented in Section 7.4.1.1 and the *attestation phase* in Section 7.4.1.2. The protocol also comprises an optional *detection phase*, presented in Section 7.4.2. The *deployment phase* is executed only once by the operator $\mathscr{O}$, who initializes the network. Following deployment, the verifier $\mathscr{V}$ can periodically verify the software integrity of all devices in the network by initiating the *attestation phase*. In the case where verification fails, $\mathscr{V}$ initiates the *detection phase*, in order to identify the origin of the failure. We demonstrate the security of CoRA in Section 7.5, based on the security of its underlying aggregation mechanism.

7.4.1.1 Deployment Phase

In the deployment phase, the network operaror $\mathscr{O}$ runs the Setup and KeyGen of the aggregate $\mathsf{MAC}_{\mathsf{BLS}}$ scheme, and obtains the public parameters pp, and the secret keys sk_i for every device D_i. Prior to deployment, $\mathscr{O}$ initializes each device with a $\mathsf{MAC}_{\mathsf{BLS}}$ secret key sk_i for $i \in \{1,...,n\}$, and its own $\mathsf{MAC}_{\mathsf{BLS}}$ public value $pk_0 = X_0$. The secret keys are used by devices to generate $\mathsf{MAC}_{\mathsf{BLS}}$ tags



Level 1	Level 2	Level 3 & 4
• $[\tau_7]$ =$\mathsf{Mac}_{sk_7}(s_7)$	• $[\tau_3]$ = $\mathsf{AggMac}(\{\tau_7,\tau_8,\tau_3\})$	• $[\tau_1]$ = $\mathsf{AggMac}(\{\tau_3,\tau_4,\tau_1\})$
• $[\tau_8]$ =$\mathsf{Mac}_{sk_8}(s_8)$	• $[\tau_4]$ = $\mathsf{AggMac}(\{\tau_9,\tau_4,\})$	• $[\tau_2]$ = $\mathsf{AggMac}(\{\tau_5,\tau_6,\tau_2\})$
⋮	⋮	
• $[\tau_{12}]$ =$\mathsf{Mac}_{sk_{12}}(s_{12})$	• $[\tau_6]$ = $\mathsf{AggMac}(\{\tau_{11},\tau_{12},\tau_6\})$	• $[\tau_s]$ = $\mathsf{AggMac}(\{\tau_1,\tau_2,\tau_s\})$

Figure 7.2: Aggregation tree containing nodes $\{A_s, D_1, ..., D_{12}\}$

τ_i on their internal state S_i. The tags will later be aggregated and forwarded to $\mathcal{V}$. The attestation and aggregation steps are developped in Section 7.4.1.2. In addition, each D_i is initialized with a counter value cnt_k, which corresponds to the attestation sequence k. cnt_k is initialized to 0, and incremented by the verifier $\mathcal{V}$ and each node after each attestation process. The counters are used to monitor the attestation sequence number, thus preventing replay attacks. Indeed, upon receiving the attestation request, node D_i verifies that the value of the associated counter is greater than or equal to the counter value stored locally. Each device in the network possesses a unique identifier ID_i, which can for example, be derived from its IP address. $\mathcal{O}$ is able to deploy new devices in the swarm at any time, by executing the initialization process described above.

7.4.1.2 Attestation Phase

The attestation phase allows the network operator $\mathcal{O}$ and the verifier $\mathcal{V}$, to verify the integrity of every device in the swarm.

Overview. The goal of the attestation phase is to iteratively authenticate each device D_i's internal software state S_i, and propagate said attestation proofs up the attestation tree until it reaches the verifier $\mathcal{V}$. The attestation phase of CoRA (Figure 7.2), is divided in three sub-phases, namely request dissemination, attestation, and verification. $\mathcal{V}$ initiates the request dissemination

phase by sending an attestation request $attReq$ to the closest device in his communication range A_s. Upon receiving $attReq$, A_s verifies that it is indeed a valid request from $\mathcal{V}$, and forwards to its children down the attestation tree. Each device in the tree receiving the request, verifies its authenticity and freshness. This initial step securely communicates to each device that a new attestation process has been initiated. Upon authenticating the request, leaf nodes generate the attestation report on their internal software state and send it to their parents. Intermediary nodes authenticate said reports, generate their own attestation, and in turn send the aggregated result to their parents. $\mathcal{V}$ receives the final report from A_s. He then verifies the validity of the aggregate report, hence confirming the integrity (or lack thereof) of every device in the network.

(1) Request dissemination. $\mathcal{V}$ initiates the attestation phase by disseminating an attestation request to the nearest device in its communication range A_s. The request $attReq$ is generated as follows:

- Generate a random nonce $N_k \xleftarrow{\$} \{0,1\}^{l_N}$ for the current attestation process, where l_N is the size of the nonce.
- Update $cnt_k \leftarrow cnt_k + 1$
- Choose a random collision-resistant hash function H
- Generate a signature of knowledge on message $m = N_k$ as $\pi_0 = SoK\{x_0 : X_0 = g^{x_0}\}(m)$.
- Define challenge $c_0 \leftarrow \{n, X_0, Cert_{X_0}, \pi_0\}$, where X_0 is the $\mathsf{MAC}_{\mathsf{BLS}}$ public key for $\mathcal{O}$ and $Cert_{X_0}$ a valid certificate on X_0. n is the size of the swarm.
- $attReq \leftarrow c_0$

Upon receiving $attReq$, A_s authenticates $\mathcal{V}$ by verifying π_0. It then propagates the request down the attestation tree. Intermediary nodes authenticate the request before forwarding it to their children. This authentication step mitigates against distributed DoS attacks, whereby an attacker might render devices unavailable by sending them false attestation requests.

(2) Attestation. Upon checking the authenticity and freshness of $attReq$, each leaf node D_i proceeds to measuring its internal software software state S_i. It then runs the attestation generation function **GenAtt**(sk_i, S_i) defined as follows:

GenAtt (sk_i, S_i)

1: Generate $\tau_i \leftarrow \mathsf{MAC}_{\mathsf{BLS}}(sk_i, m_i)$, where $m_i = (N_k || cnt_k || S_i)$. Including N_k and cnt_k in the message ensures the freshness of the tag.
2: Generate a signature of knowledge π_i, of the secret key on the tag τ_i, where $\pi_i = SoK\{x_i : \tau_i = h_i^{x_i} \wedge X_i = g^{x_i}\}(\tau_i)$.
3: Define an empty list $\mathscr{C}_i = \emptyset$
4: Return report $a_i \leftarrow (ID_i, \mathscr{H}(N_k || cnt_k), \tau_i, \pi_i, \mathscr{C}_i)$

Upon receiving the attestation reports a_i and a_j from its children, intermediary node D_l runs the attestation aggregation function **AggAtt**(τ_i, τ_j):

AggAtt (τ_i, τ_j)

1: Verify π_i and π_j. If the verification succeeds, proceed to step 2. Else, set $\mathscr{C}_l = \{ID_k\}_{k \in \{i,j\}}$, where the identifier(s) in $\mathscr{C}_l$ corresponds to those whose signature of knowledge verification failed.
2: Generate $\tau_l \leftarrow \mathsf{MAC}_{\mathsf{BLS}}(sk_l, m_l)$, where message $m_l = (N_k || cnt_k || S_l)$
3: Compute $\tau \leftarrow \mathsf{AggMAC}_{BLS}(\tau_i, \tau_j, \tau_l)$
4: Generate $\pi_l = SoK\{x_l : \tau_l = h_l^{x_l} \wedge X_l = g^{x_l}\}(\tau)$
5: Return $a_l \leftarrow (ID_l, \mathscr{H}(N_k || cnt_k), \tau, \pi_l, \mathscr{C}_l)$

At the end of the attestation generation, aggregation, and propagation phase, $\mathscr{V}$ receives the final attestation report a_s from A_s, and proceeds to verifying its validity.

(3) Verification. Upon receiving the final report in time $t \leq t_{exp}$, $\mathscr{V}$ runs the aggregated attestation verification function **VrfAggAtt**$(sk_1, ..., sk_n, \tau_1, ..., \tau_n, \{S_1, ..., S_n\})$ on the list of expected healthy states $\{S_1, ..., S_n\}$ it possesses. The time t_{exp} is defined through prior real-world experiments, in order to estimate an upper-bound of the overall attestation time. Such experiments must include network delays, transmission times, and individual attestation generation and aggregation times.

VrfAggAtt $(\{sk_i\}_{i=1}^n, \{\tau_i\}_{i=1}^n, \{S_i\}_{i=1}^n)$

1: $b \leftarrow \mathsf{AggVerifyMAC}_{BLS}(\{sk_i\}_{i=1}^n, \tau, \{S_i\}_{i=1}^n)$
2: If $b = 1$ **and** $\mathscr{C}_s = \emptyset$, return 1
3: Otherwise return 0

If the function returns 1, $\mathscr{V}$ concludes that the network is in a trustworthy state. Otherwise, he concludes that at least one device is compromised and proceeds to the detection phase.

7.4.2 Detection Phase

An attacker $\mathscr{A}$ who performed a mismatch attack, by aggregating an erroneous attestation report, will be identified in the detection phase. Let D_c be the target device in such attacks. Upon the function **VrfAggAtt** returning 0, $\mathscr{V}$ proceeds as follows:

1. $\mathscr{V}$ requests the signatures of knowledge $\{\pi_i\}_{i=1}^n$ for every node in the network.
2. $\mathscr{V}$ verifies the corresponding tags $\tau_i{}_{i=1}^n$.
3. $\mathscr{V}$ proceeds iteratively down the attestation tree until it reaches the node D_i which did not aggregate the valid tag/SoK pair.

The signature of knowledge ensures that each node remains accountable for the value it has previously aggregated during the attestation phase. Considering the assumption that no adversary can easily perform physical attacks, and generate a valid $\mathsf{MAC}_{\mathsf{BLS}}$ tag without a valid secret key sk_i, this ensures that $\mathscr{V}$ is able to identify each node that aggregates an erroneous report.

7.5 Security Analysis

In this section, we provide a proof of security of CoRA, as stated by the following theorem:

Theorem 7.3. *CoRA is secure against forgery attacks, under the assumption that aggregate MAC_{BLS} is unforgeable.*

As specified in Section 7.2.3, a swarm attestation scheme is secure against forgery if no adversary is able to produce a forgery for an uncompromised device (for which he doesn't know the secret key), even after accessing at most $n-1$ attestation results. We derive the security of our swarm attestation protocol by drawing the parallel between the security definition for swarm attestation forgery [FNRT14], and the security of the aggregate $\mathsf{MAC}_{\mathsf{BLS}}$ scheme.
The verifier $\mathcal{V}$ in a swarm attestation protocol returns 1 if $\mathsf{AggVerify}(pp, \{sk_i\}_{i=1}^n, \{m_i\}_{i=1}^n, \tau) = 1$, i.e. if τ is a valid aggregate tag on the set of expected valid states $\{S_i\}_{i=1}^n$, and the incorporated nonce N_k is the session nonce provided by $\mathcal{V}$. Let $\mathcal{A}$ be an attacker on the network whose goal is to produce a valid forgery of the attestation response. $\mathcal{A}$ has compromised prover D_c. We consider two types of adversaries:

Type 1: $\mathcal{A}_1$ does not alter the attestation of D_c to be included in the aggregate response.
Type 2: $\mathcal{A}_2$ modifies the internal state of D_c, and generates an attestation τ_c on $(ID_c, S_c||N_k||cnt_k)$.

In the first case, according to our assumptions, $\mathcal{A}_1$ cannot physically compromise D_c and retrieve its secret key x_c. $\mathcal{A}_1$'s only strategy is to use an attestation response of a previously generated attestation $\tau_{c_{\text{prev}}}$, on the same software configuration S_c and an old nonce N_{prev}. Let N_{curr} be the random nonce generated by $\mathcal{V}$ in the current round of attestation. The probability that $N_{\text{prev}} = N_{\text{curr}}$ is equal to 2^{-l_N}, which is negligible for a sufficiently large nonce. $\mathcal{A}_1$ will therefore output a valid attestation response only with negligible probability.

In the second case, we observe the following analogy: the security definition of resistance against forgery for a remote attestation protocol, is exactly the security definition of unforgeability of the aggregate $\mathsf{MAC}_{\mathsf{BLS}}$ scheme, namely forgery for an adversary who had access to up to $n-1$ tags. $\mathcal{A}_2$'s advantage in successfully producing a forgery is therefore the same as $\mathcal{A}_2$'s advantage in producing a valid aggregate $\mathsf{MAC}_{\mathsf{BLS}}$ forgery, which is negligible under the CDH assumption in the random oracle model (see proof in Section 7.3.3). Indeed, in order for $\mathcal{A}_2$ to generate a valid attestation on a modified state without physically compromising D_c, he needs to forge an aggregate $\mathsf{MAC}_{\mathsf{BLS}}$ scheme on all aggregated tags, assuming that at least one device in the network is honest. According to Theorem 7.2, aggregate $\mathsf{MAC}_{\mathsf{BLS}}$ is unforgeable provided that $\mathsf{MAC}_{\mathsf{BLS}}$ is unforgeable (see proof in Section 7.3.4). The probability of $\mathcal{A}_2$ generating said valid attestation is therefore negligible.

7.6 Complexity and Efficiency Analysis

In this section, we evaluate the complexity of CoRA with respect to state of the art swarm attestation protocols.

7.6.1 CoRA against state of the art constructions

Our aggregate algebraic MAC performs significantly better than existing aggregate signature schemes in the context of swarm attestation. Table 7.1 presents a comparison between our aggregate MAC scheme, and existing constructions. Table 7.2 provides a comparison between our scheme and existing state of the art swarm attestation protocol. The total number of operations are functions of the following parameters: n is the total number of devices, k is the height of the spanning tree, (A) denotes additions in the cyclic group $\mathbb{G}$, (M) denotes a multiplication in $\mathbb{G}$, (E) denotes an exponentiation in $\mathbb{G}$, and (P) denotes a pairing in $\mathbb{G}_T$ where $\mathbb{G}_T$ is a cyclic group of prime order q.

Scheme	**Type**	**Aggregation time**	**Verification time**	**MAC operation**
KL [KL08]	Sequential	n	$n(n-1)M$	Block cipher
EFG+ [EFG+10]	Sequential	n	$n(n-1)M$	Block cipher
MT [MT07]	Sequential	n	$n(n-1)M$	Block cipher
Aggregate $\mathsf{MAC}_{\mathsf{BLS}}$	Non-sequential	$k\log(n)$	nM	Algebraic group

Table 7.1: Comparison between the aggregate $\mathsf{MAC}_{\mathsf{BLS}}$ scheme and existing Aggregate MAC constructions.

In the following comparison table, "partial" detection denotes detection of a bad software configuration by trusted devices themselves. "Full" detection also considers the case when an adversary on the network tries to aggregate a bad value as the attestation of a target device, resulting in said device's attestation to be dropped.

Scheme	**Attestation**	**Verification**	**Detection Mode**	**Detection**
SEDA	$4n \cdot \text{MAC}$	$4n \cdot \text{Vfy(MAC)}$	Full	$n \cdot$ MAC
LISA	$n \cdot \text{MAC} + n \cdot \text{unicast}$	$n \cdot \text{Vfy(MAC)}$	N/A	N/A
SANA	$[k\log(n) \cdot M] + nE$	$n \cdot P + n \cdot M$	Partial	$n \cdot A$
CoRA	$[k\log(n) \cdot M] + nE$	nM	Full	$2n \cdot E + k\log(n) \cdot A$

Table 7.2: Comparison between CoRA and existing swarm attestation protocols.

7.6.2 Testbed implementation and performance analysis

We implemented CoRA on a Teensy 3.2 microcontroller. The Teensy 3.2 board is an Arduino-compatible microcontroller, featuring a 32 bit ARM processor with a 72 MHz Cortex-M4 core. It also features 256 kB Flash and 64 kB RAM memory. The board provides the minimal hardware

requirements for remote attestation as stated in Section 7.2.3. Cryptographic algorithms are implemented based on the micro-ecc[Mac17] library. As presented in Table 7.3, we use a SHA-256 hash function.

Algorithm	Run-time (ms)
SHA-256	0.244
SHA-256 HMAC	0.809

Table 7.3: Timings of cryptographic algorithms on Teensy 3.2

Mac Runtime (s)	Number of devices	AggMac Runtime (s)
0.047	100	5.55
	500	27.8
	1000	56.1
	1500	84.3

Table 7.4: Performances of the $\mathsf{MAC}_{\mathsf{BLS}}$ functions

Table 7.4 shows the runtime performance of $\mathsf{MAC}_{\mathsf{BLS}}$. The choice of algebraic MACs (in elliptic curve groups) provides a more efficient verification step, considering that computing AggVerify is the same as computing AggMac. The implementation makes use of secp160r1 curves [Res00].

7.7 Conclusion

We introduced in this chapter a new swarm attestation protocol CoRA, with a sequential erroneous attestation aggregation detection mechanism. CoRA leverages the aggregating property of its underlying in-network aggregation mechanism, namely aggregate $\mathsf{MAC}_{\mathsf{BLS}}$, to provide a highly scalable swarm attestation protocol with efficient verification. In order to detect the malicious injection of erroneous attestation, CoRA comprises a scalable detection algorithm, which leverages the algebraic property of algebraic MACs to generate proofs of knowledge, on a device's secret key. The detection method allows the identification of a compromised node in the network, thus preventing DoS attacks on the verifier. CoRA is suitable for IoT application domains such as the industrial Internet of Things, where identifying compromised nodes is an important requirement in order to ensure the control and management of safety-critical processes.

CHAPTER 8

CONCLUSIONS AND PERSPECTIVES

Devices in Internet of Things (IoT) applications are increasingly becoming the targets of attacks due to their limited resources and capabilities. Ensuring their integrity and the secure authentication of the data they exchange with other devices in the network is essential to the security, safety, and availability of applications. In addition, the pervasive nature of device deployment and the large amount of personal data they collect and process contributed to raising concerns over the privacy of end users. Attestation as a security mechanism, enables the verification of the integrity of devices, notably by leveraging trusted hardware modules they embed. Attestation mechanisms based on cryptographic protocols can also be used to authenticate devices, with the additional ability to implement privacy-preserving solutions. On one end, although privacy-preserving attestation schemes have been proposed over the years for devices implementing high-end embedded hardware, approaches in the literature suffer from a number of limitations that hamper their implementation on low-level hardware modules. This thesis addresses those limitations by introducing a new privacy-preserving pre-Direct Anonymous Attestation mechanism that is efficient enough to be implemented on low-level devices, yielding the development of secure and privacy-preserving authentication protocols in different IoT applications. On the other end, the security challenges inherent to the large scale deployment of devices in a number of IoT applications require the development of scalable remote attestation protocols. Approaches to remote device attestation based on scalable cryptographic primitives suffer from limitations that can lead to additional attacks on the networks, notably on the availability of the control system. In this thesis, we address those limitations by developing a new remote attestation protocol that offers a new way of preventing such attacks.

Contributions of this thesis

In the first part of this thesis, we introduce a new pre-DAA scheme, and we leverage its security properties to develop secure and privacy-preserving protocols for two different applications.

In Chapter 4, we design a new pre-DAA that we prove to be secure under the q—SDH assumption in the random oracle model. Our pre-DAA scheme satisfies stringent efficiency requirements, as proved through its implementation on a Global Platform compliant Java card SIM card for two distinct applications.
In Chapter 5, we introduce a new pseudonym scheme for vehicle-to-everything communication based on our pre-DAA scheme. Our pseudonym scheme provides a decentralized alternative to traditional public key infrastructures used in VANETs to provide an authentication framework for relaying safety-related messages. With our construction, vehicles can generate their own pseudonyms, and randomize said pseudonyms to remain anonymous while relaying messages over the network. This optimizes network bandwidth during the communication process, thus respecting the real-time constraints of a VANET. In addition, our scheme ensures vehicle and location privacy given that the vehicle remains anonymous, and their identifiers cannot be recovered from a given signature. Our protocol enforces driver and vehicle liability and non-repudiation, such that no vehicle can deny having sent a message over the network if it has indeed generated a signature on the corresponding message. We argue that by ensuring user/vehicle responsibility without compromising user privacy, our new and decentralized pseudonym scheme is suitable for widespread deployment.
In Chapter 6, we leverage the SIM card in a commuter's smartphone as a trusted execution environment, in order to provide secure and privacy-preserving authentication and access control on public transport networks. Our access control protocol PAYGO, ensures that honest users are not traceable on the network, hence providing location privacy. PAYGO pass validations can only be linked to a given pass whenever their is a fraud attempt, thus enforcing user responsibility. PAYGO leverages the security properties of our pre-DAA scheme to ensure strong anonymous authentication. Our protocol satisfies the stringent efficiency requirements of mobile-based pass validations on public transports.

The second part of this thesis focuses on the concept of remote attestation, notably in the context of device swarms. In Chapter 7, we introduced CoRA, a novel swarm attestation protocol based on an aggregate algebraic MAC primitive. Algebraic MACs are suitable for use cases where the entities generating the tag and the verifier share the secret MAC keys. The primitive therefore allows to efficiently verify the attestation report, without requiring computationally expensive operations such as pairings. CoRA leverages the aggregating properties of the aggregate MAC in order to provide a scalable swarm attestation protocol. CoRA also leverages the public values associated with each secret key in order to generate efficient signatures of knowledge on the secret. This latter property enforces the liability of each node, whose attestation is therefore augmented with a proof of knowledge. Consequently, whenever the final attestation verification fails, the verifier is able to perform a top-down tree traversal in order to identify which node generated an erroneous attestation. This last step is referred to as sequential detection. We prove the security of the aggregate MAC under the computational Diffie-Hellman assumption in the random oracle model, and subsequently prove the security of CoRA.

Perspectives

Future improvements to our pre-DAA schemes would include proving its security using the simulation-based approach. DAA schemes are complex, and capturing all the desired security properties in the security model has been a challenging task. Constructions using the simulation-based approach exist for DAA schemes, but not for pre-DAA schemes. In the simulation-based approach, the use of an ideal functionality which models all parties, that can be in different corrupted state, allows to capture all the desired security guarantees. This approach is instantiated in the *Universal Composability* (UC) framework, which can be designed to comprise all the desired security notions by definition. The UC model provides security guarantees that would imply that our protocol is secure in any configuration, including its concurrent implementation with other protocols. We believe that with the simulation-based approach, there is a lower risk of overlooking some security properties.
It would also be interesting to apply our pre-DAA scheme to other IoT use cases that require strong privacy, such as Smart Metering. Indeed, there are a number of privacy concerns regarding the deployment of smart meters, that collect extensive data on user consumption. A solution could be to include an encryption layer in the pre-DAA scheme, thus ensuring both the confidentiality and privacy-preserving authentication of meters. For such an application to be efficiently deployed, the pre-DAA scheme must introduce an additional *aggregation* function, which could optimize the amount of data forwarded by a collection of meters in the same borough for example.

Future improvements to CoRA specifically, and swarm attestation protocols generally, would be the implementation of a constant-time detection mechanism. Indeed, using aggregate cryptographic schemes does not allow a verifying entity to retrieve an individual signature from the aggregated result. This would indeed break the security of the underlying aggregate signature scheme. Such a property could however allow one to detect in constant time which node has generated an erroneous attestation result.
Future work in the development of efficient swarm attestation protocols could also include the development of aggregate signature schemes, where the verification step does not require any pairing computation.
Finally, it would be interesting to adapt the above constructions for devices that do not have the computational capabilities of resource-constrained devices such as SIM cards, notably sensors for example.

List of Publications

- [DGL+18] Aïda Diop, Saïd Gharout, Maryline Laurent, Jean Leneutre, and Jacques Traoré. Questioning the security and efficiency of the esiot approach. In Panos Papadimitratos, Kevin R. B. Butler, and Christina Pöpper, editors, *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks, WiSec 2018, Stockholm, Sweden, June 18-20, 2018*, pages 202–207. ACM, 2018.

- [DDR+19] Nicolas Desmoulins, Aïda Diop, Yvan Rafflé, Jacques Traoré, and Josselin Gratesac. Practical anonymous attestation-based pseudonym schemes for vehicular networks. In *2019 IEEE Vehicular Networking Conference, VNC 2019, Los Angeles, CA, USA, December 4-6, 2019*, pages 1–8. IEEE, 2019.

- [DLLT20] Aïda Diop, Maryline Laurent, Jean Leneutre, and Jacques Traoré. Cora: A scalable collective remote attestation protocol for sensor networks. In Steven Furnell, Paolo Mori, Edgar R. Weippl, and Olivier Camp, editors, *Proceedings of the 6th International Conference on Information Systems Security and Privacy, ICISSP 2020, Valletta, Malta, February 25-27, 2020*, pages 84–95. SCITEPRESS, 2020.

- [DDT] Nicolas Desmoulins, Aïda Diop, and Jacques Traoré. Pass-as-you-go: A contactless mobile transit pass service. In *submission (AsiaCCS 2021)*.

BIBLIOGRAPHY

[3GP17] 3GPP. Technical Specification Group Services and System Aspect Release 14 Description;Summary of Rel-14 Work Items(Release 14). Technical Report (TR) 21.914, 3rd Generation Partnership Project (3GPP), 05 2017. Version 14.0.0.

[ABI+15] N. Asokan, Franz Ferdinand Brasser, Ahmad Ibrahim, Ahmad-Reza Sadeghi, Matthias Schunter, Gene Tsudik, and Christian Wachsmann. SEDA: scalable embedded device attestation. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015*, pages 964–975, 2015.

[ABL13] ASSA ABLO. Evaluation of the world's first pilot using nfc phones for check-in and hotel room keys. Technical report, 2013. Available at `www.assaabloy.com`.

[ACI+16] Moreno Ambrosin, Mauro Conti, Ahmad Ibrahim, Gregory Neven, Ahmad-Reza Sadeghi, and Matthias Schunter. SANA: secure and scalable aggregate network attestation. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 731–742, 2016.

[AGL+13] Ghada Arfaoui, Sébastien Gambs, Patrick Lacharme, Jean-François Lalande, Roch Lescuyer, and Jean Claude Paillès. A privacy-preserving contactless transport service for NFC smartphones. pages 282–285, 2013.

[Alg09] Digital Signature Algorithm. Digital signature standard DSS. Technical report, National Institute of Standards and Technology, 2009.

[ALT+15] Ghada Arfaoui, Jean-François Lalande, Jacques Traoré, Nicolas Desmoulins, Pascal Berthomé, and Saïd Gharout. A practical set-membership proof for privacy-preserving NFC mobile ticketing. *PoPETs*, 2015(2):25–45, 2015.

[ARM09] ARM. Building a secure system using trustzone technology. Technical report, 2009. Available at `www.arm.com`.

[Ass14] Calypso Networks Association. Calypso functional specification - card application. Technical report, 2014. Available at `http://www.calypsostandard.net`.

[AvRDN19] Abhishta Abhishta, Roland van Rijswijk-Deij, and Lambert JM Nieuwenhuis. Measuring the impact of a successful ddos attack on the customer behaviour

of managed dns service providers. *ACM SIGCOMM Computer Communication Review*, 48(5):70–76, 2019.

[BB04] Dan Boneh and Xavier Boyen. Short signatures without random oracles. In *Advances in Cryptology - EUROCRYPT 2004, International Conference on the Theory and Applications of Cryptographic Techniques, Interlaken, Switzerland, May 2-6, 2004, Proceedings*, pages 56–73, 2004.

[BBS04] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short group signatures. In *Advances in Cryptology - CRYPTO 2004, 24th Annual International Cryptology-Conference*, 2004.

[BCC88] Gilles Brassard, David Chaum, and Claude Crépeau. Minimum disclosure proofs of knowledge. *Journal of computer and system sciences*, 37(2):156–189, 1988.

[BCC04] Ernest F. Brickell, Jan Camenisch, and Liqun Chen. Direct anonymous attestation. In *Proceedings of the 11th ACM Conference on Computer and Communications Security, CCS 2004, Washington, DC, USA, October 25-29, 2004*, pages 132–145, 2004.

[BCL08] Ernie Brickell, Liqun Chen, and Jiangtao Li. A new direct anonymous attestation scheme from bilinear maps. In *International Conference on Trusted Computing*, pages 166–178. Springer, 2008.

[BCL09] Ernie Brickell, Liqun Chen, and Jiangtao Li. Simplified security notions of direct anonymous attestation and a concrete scheme from pairings. *Int. J. Inf. Sec.*, 2009.

[BDGT17] Amira Barki, Nicolas Desmoulins, Saïd Gharout, and Jacques Traoré. Anonymous attestations made practical. In *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec 2017, Boston, MA, USA, July 18-20, 2017*, pages 87–98, 2017.

[BDPR98] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In *Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45. Springer, 1998.

[BEMS+15] Ferdinand Brasser, Brahim El Mahjoub, Ahmad-Reza Sadeghi, Christian Wachsmann, and Patrick Koeberl. Tytan: tiny trust anchor for tiny devices. In *Proceedings of the 52nd Annual Design Automation Conference*, pages 1–6, 2015.

[BFG+13] David Bernhard, Georg Fuchsbauer, Essam Ghadafi, Nigel P. Smart, and Bogdan Warinschi. Anonymous attestation with user-controlled linkability. *Int. J. Inf. Sec.*, 12(3):219–249, 2013.

[BGLS03] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. Aggregate and verifiably encrypted signatures from bilinear maps. In *Advances in Cryptology - EUROCRYPT 2003, International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, May 4-8, 2003, Proceedings*, pages 416–432, 2003.

[BKL+07] Andrey Bogdanov, Lars R Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew JB Robshaw, Yannick Seurin, and Charlotte Vikkelsoe. Present: An ultra-lightweight block cipher. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 450–466. Springer, 2007.

[BL11a] Ernie Brickell and Jiangtao Li. Enhanced Privacy ID from bilinear pairing for hardware authentication and attestation. *IJIPSI*, 2011.

[BL11b] Ernie Brickell and Jiangtao Li. Enhanced Privacy ID from bilinear pairing for hardware authentication and attestation. *International Journal of Information Privacy, Security and Integrity 2*, 1(1):3–33, 2011.

[BLS01] Dan Boneh, Ben Lynn, and Hovav Shacham. Short signatures from the weil pairing. In *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings*, pages 514–532, 2001.

[BNPS03] Mihir Bellare, Chanathip Namprempre, David Pointcheval, and Michael Semanko. The One-More-RSA-Inversion problems and the security of Chaum's blind signature scheme. *Journal of Cryptology*, 16(3), 2003.

[Bol03] Alexandra Boldyreva. Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme. In *International Workshop on Public Key Cryptography*, pages 31–46. Springer, 2003.

[Bon98] Dan Boneh. The decision diffie-hellman problem. In *International Algorithmic Number Theory Symposium*, pages 48–63. Springer, 1998.

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM conference on Computer and communications security*, pages 62–73, 1993.

[BSS+13] Ray Beaulieu, Douglas Shors, Jason Smith, Stefan Treatman-Clark, Bryan Weeks, and Louis Wingers. The simon and speck families of lightweight block ciphers. *IACR Cryptology ePrint Archive*, 2013(1):404–449, 2013.

[Can01] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 136–145. IEEE, 2001.

[CCD+17] Jan Camenisch, Liqun Chen, Manu Drijvers, Anja Lehmann, David Novick, and Rainer Urian. One tpm to bind them all: Fixing tpm 2.0 for provably secure anonymous attestation. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 901–920. IEEE, 2017.

[CD97] Ronald Cramer and Ivan Damgård. Linear zero-knowledge—a note on efficient zero-knowledge proofs and arguments. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 436–445, 1997.

[CD16] Victor Costan and Srinivas Devadas. Intel SGX explained. 2016.

[CDL] Jan Camenisch, Manu Drijvers, and Anja Lehmann. Universally composable direct anonymous attestation. In *Public-Key Cryptography - PKC 2016 - 19th IACR*.

[CDL16] Jan Camenisch, Manu Drijvers, and Anja Lehmann. Anonymous attestation using the strong diffie hellman assumption revisited. In *Trust and Trustworthy Computing - 9th International Conference, TRUST 2016, Vienna, Austria, August 29-30, 2016, Proceedings*, 2016.

[CDL17] Jan Camenisch, Manu Drijvers, and Anja Lehmann. Anonymous attestation with subverted tpms. In *Annual International Cryptology Conference*, pages 427–461. Springer, 2017.

[CDRT17] Xavier Carpent, Karim El Defrawy, Norrathep Rattanavipanon, and Gene Tsudik. Lightweight swarm attestation: A tale of two lisa-s. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2-6, 2017*, pages 86–100, 2017.

[CFPS09] Claude Castelluccia, Aurélien Francillon, Daniele Perito, and Claudio Soriente. On the difficulty of software-based attestation of embedded devices. In *Proceedings of the 16th ACM conference on Computer and communications security*, pages 400–409. ACM, 2009.

[CGH04] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited. *Journal of the ACM (JACM)*, 51(4):557–594, 2004.

[CH91] David Chaum and Eugène Van Heyst. Group signatures. In *Advances in Cryptology - EUROCRYPT '91, Workshop on the Theory and Application of of Cryptographic Techniques, Brighton, UK, April 8-11, 1991, Proceedings*, pages 257–265, 1991.

[Cha83] David Chaum. Blind signatures for untraceable payments. In *Advances in cryptology*, pages 199–203. Springer, 1983.

[Che09] Liqun Chen. A daa scheme requiring less tpm resources. In *International Conference on Information Security and Cryptology*, pages 350–365. Springer, 2009.

[CL01] Jan Camenisch and Anna Lysyanskaya. An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In *International conference on the theory and applications of cryptographic techniques*, pages 93–118. Springer, 2001.

[CL04] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *Annual International Cryptology Conference*, pages 56–72. Springer, 2004.

[CMRR19] Lily Chen, Dustin Moody, Andrew Regenscheid, and Karen Randall. Recommendations for discrete logarithm-based cryptography: Elliptic curve domain parameters. Technical report, National Institute of Standards and Technology, 2019.

[CMZ14] Melissa Chase, Sarah Meiklejohn, and Greg Zaverucha. Algebraic macs and keyed-verification anonymous credentials. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Scottsdale, AZ, USA, November 3-7, 2014*, pages 1205–1216, 2014.

[CNW11] Liqun Chen, Siaw-Lynn Ng, and Guilin Wang. Threshold anonymous announcement in vanets. *IEEE Journal on Selected Areas in Communications*, 29(3):605–615, 2011.

[Com17] European Commission. Cooperative, connected and automated mobility (CCAM) . Technical report, European Commisssion (EC), 2017. Available at `https://ec.europa.eu/transport/themes/its/c-its_en`.

[Coo08] Cooper, Dave. Internet x. 509 public key infrastructure certificate and certificate revocation list profile. 2008.

[Cor15] Intel Corporation. Intel®software guard extensions(intel®sgx), 2015. Available at `https://www.intel.com/content/www/us/en/architecture-and-technology/software-guard-extensions.html`.

[CPHL07] Giorgio Calandriello, Panos Papadimitratos, Jean-Pierre Hubaux, and Antonio Lioy. Efficient and robust pseudonymous authentication in vanet. In *Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks*, pages 19–28, 2007.

[CPS06] Haowen Chan, Adrian Perrig, and Dawn Xiaodong Song. Secure hierarchical in-network aggregation in sensor networks. In *Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, Ioctober 30 - November 3, 2006*, pages 278–287, 2006.

[CPS10] Liqun Chen, Dan Page, and Nigel P Smart. On the design and implementation of an efficient daa scheme. In *International Conference on Smart Card Research and Advanced Applications*, pages 223–237. Springer, 2010.

[Cri09] Common Criteria. Common criteria eal5+ certification security target. Technical report, 2009.

[CS97] Jan Camenisch and Markus Stadler. Efficient group signature schemes for large groups (extended abstract). In *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 1997, Proceedings*, pages 410–424, 1997.

[CWC13] Mauro Conti, Jeroen Willemsen, and Bruno Crispo. Providing source location privacy in wireless sensor networks: A survey. *IEEE Communications Surveys & Tutorials*, 15(3):1238–1280, 2013.

[Dam98] Ivan Damgård. Commitment schemes and zero-knowledge protocols. In *School organized by the European Educational Forum*, pages 63–86. Springer, 1998.

[DDR+19] Nicolas Desmoulins, Aïda Diop, Yvan Rafflé, Jacques Traoré, and Josselin Gratesac. Practical anonymous attestation-based pseudonym schemes for vehicular networks. In *2019 IEEE Vehicular Networking Conference, VNC 2019, Los Angeles, CA, USA, December 4-6, 2019*, pages 1–8. IEEE, 2019.

[DDT] Nicolas Desmoulins, Aïda Diop, and Jacques Traoré. Pass-as-you-go: A contactless mobile transit pass service. In *submission (AsiaCCS 2021)*.

[DFGMGTB14] José María De Fuentes, Lorena González-Manzano, Ana Isabel González-Tablas, and Jorge Blasco. Security models in vehicular ad-hoc networks: A survey. *IETE Technical Review*, 31(1):47–64, 2014.

[DFGTR11] José María De Fuentes, Ana Isabel González-Tablas, and Arturo Ribagorda. Overview of security issues in vehicular ad-hoc networks. In *Handbook of research on mobility and computing: Evolving technologies and ubiquitous impacts*, pages 894–911. IGI global, 2011.

[DGL+18] Aïda Diop, Saïd Gharout, Maryline Laurent, Jean Leneutre, and Jacques Traoré. Questioning the security and efficiency of the esiot approach. In Panos Papadimitratos, Kevin R. B. Butler, and Christina Pöpper, editors, *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks, WiSec 2018, Stockholm, Sweden, June 18-20, 2018*, pages 202–207. ACM, 2018.

[DH76] Whitfield Diffie and Martin Hellman. New directions in cryptography. *IEEE transactions on Information Theory*, 22(6):644–654, 1976.

[DKPW12] Yevgeniy Dodis, Eike Kiltz, Krzysztof Pietrzak, and Daniel Wichs. Message authentication, revisited. In *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, 2012.

[DLLT20] Aïda Diop, Maryline Laurent, Jean Leneutre, and Jacques Traoré. Cora: A scalable collective remote attestation protocol for sensor networks. In Steven Furnell, Paolo Mori, Edgar R. Weippl, and Olivier Camp, editors, *Proceedings of the 6th International Conference on Information Systems Security and Privacy, ICISSP 2020, Valletta, Malta, February 25-27, 2020*, pages 84–95. SCITEPRESS, 2020.

[DLST14] Nicolas Desmoulins, Roch Lescuyer, Olivier Sanders, and Jacques Traoré. Direct anonymous attestations with dependent basename opening. In *Cryptology and Network Security - 13th International Conference, CANS 2014, Heraklion, Crete, Greece, October 22-24, 2014. Proceedings*, pages 206–221, 2014.

[Dou02] John R Douceur. The sybil attack. In *International workshop on peer-to-peer systems*, pages 251–260. Springer, 2002.

[DPWD11] David Derler, Klaus Potzmader, Johannes Winter, and Kurt Dietrich. Anonymous ticketing for nfc-enabled mobile phones. In *International Conference on Trusted Systems*, pages 66–83. Springer, 2011.

[DSDFY94] Alfredo De Santis, Yvo Desmedt, Yair Frankel, and Moti Yung. How to share a function securely. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 522–533, 1994.

[DY83] Danny Dolev and Andrew Yao. On the security of public key protocols. *IEEE Transactions on information theory*, 29(2):198–208, 1983.

[EBPQ14] Richard Gilles Engoulou, Martine Bellaïche, Samuel Pierre, and Alejandro Quintero. Vanet security surveys. *Computer Communications*, 44:1–13, 2014.

[EFG+10] Oliver Eikemeier, Marc Fischlin, Jens-Fabian Götzmann, Anja Lehmann, Dominique Schröder, Peter Schröder, and Daniel Wagner. History-free aggregate message authentication codes. In *Security and Cryptography for Networks, 7th International Conference, SCN 2010, Amalfi, Italy, September 13-15, 2010. Proceedings*, pages 309–328, 2010.

[ERI] ERICSSON. Mobility report - internet of things forecast, 06. Available at `https://www.ericsson.com/en/mobility-report/internet-of-things-forecast`.

[ET11] Jan-Erik Ekberg and Sandeep Tamrakar. Mass transit ticketing with nfc mobile phones. In *International Conference on Trusted Systems*, pages 48–65. Springer, 2011.

[ETFP12] Karim Eldefrawy, Gene Tsudik, Aurélien Francillon, and Daniele Perito. Smart: Secure and minimal architecture for (establishing dynamic) root of trust. In *NDSS*, volume 12, pages 1–15, 2012.

[ETS] ETSI/3GPP. Etsi ts 151 011 - digital cellular telecommunications system (phase 2+);specification of the subscriber identity module -mobile equipment (sim-me) interface - (3gpp ts 51.011 version 4.15.0 release 4). Technical report (tr), European Telecommunications Standards Institute (ETSI), 09. Version 4.15.0, Available at `https://www.etsi.org/deliver/etsi_en/302600_302699/30263702/01.03.01_30/en_30263702v010301v.pdf`.

[ETS09] ETSI. Etsi tr 102 638 v1.1.1, intelligent transport systems (its);vehicular communications; basic set of applications; definitions. Technical report, ETSI, 2009. Available at =https://www.etsi.org.

[ETS14] ETSI. Intelligent Transport Systems (ITS);Vehicular Communications; Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service. Technical Report (TR) 302 637-2, European Telecommunications Standards Institute (ETSI), 09 2014. Version 1.3.1.

[ETS18] ETSI. ETSI TR 103 415 V1.1.1 (2018-04)Intelligent Transport Systems (ITS);Security; Pre-standardization study on pseudonym change management. Technical Report (TR) 103 415, European Telecommunications Standards Institute (ETSI), 04 2018. Version 1.1.1.

[EU] EU. Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation). Technical report.

[EVI08] EVITA. Seventh Research Framework Programme (2007-2013) of the European Community - ICT-2007.6.2: ICT for cooperative systems. Technical report (tr), E-safety vehicle intrusion protected applications, 2008. Project reference 224275.

[FFS88] Uriel Feige, Amos Fiat, and Adi Shamir. Zero-knowledge proofs of identity. *Journal of cryptology*, 1(2):77–94, 1988.

[FIP07] Federal Information Processing Standards Publication FIPS. Security requirements for cryptographic modules. Technical report, 2001, updated in 2007.

[FMC10] Nicolas Falliere, Liam O Murchu, and Eric Chien. W32.stuxnet dossier. *Symantec*, 2010.

[FNRT12] Aurélien Francillon, Quan Nguyen, Kasper Bonne Rasmussen, and Gene Tsudik. Systematic treatment of remote attestation. *IACR Cryptology ePrint Archive*, 2012.

[FNRT14] Aurélien Francillon, Quan Nguyen, Kasper B Rasmussen, and Gene Tsudik. A minimalist approach to remote attestation. In *2014 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1–6. IEEE, 2014.

[FRF+07] Julien Freudiger, Maxim Raya, Márk Félegyházi, Panos Papadimitratos, and Jean-Pierre Hubaux. Mix-zones for location privacy in vehicular networks. In *ACM Workshop on Wireless Networking for Intelligent Transportation Systems (WiN-ITS)*, number CONF, 2007.

[FS86] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology - CRYPTO '86, Santa Barbara, California, USA, 1986, Proceedings*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, 1986.

[FS01] Pierre-Alain Fouque and Jacques Stern. Fully distributed threshold RSA under standard assumptions. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 310–330. Springer, 2001.

[GCDD02] Blaise Gassend, Dwaine E. Clarke, Marten Van Dijk, and Srinivas Devadas. Silicon physical random functions. In Vijayalakshmi Atluri, editor, *Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, Washington, DC, USA, November 18-22, 2002*, pages 148–160. ACM, 2002.

[GD07] Gilles Guette and Bertrand Ducourthial. On the sybil attack detection in vanet. In *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*, pages 1–6. IEEE, 2007.

[Gem18a] Gemalto. Gemalto enables digitization of hong kong's octopus card into samsung pay. Technical report, 2018. Available at `https://www.samsung.com/hk_en/samsungpay/smartoctopus/`.

[Gem18b] Gemalto. Nfc tapping into brazilian transport market. Technical report, 2018. Available at `https://www.gemalto.com/transport/inspired/nfc-tapping-brazil`.

[GM84] Shafi Goldwasser and Silvio Micali. Probabilistic encryption. *Journal of computer and system sciences*, 28(2):270–299, 1984.

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In Robert Sedgewick, editor, *Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 6-8, 1985, Providence, Rhode Island, USA*, pages 291–304. ACM, 1985.

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on computing*, 18(1):186–208, 1989.

[GPS08] Steven D. Galbraith, Kenneth G. Paterson, and Nigel P. Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[GSM12] GSMA. White paper: Mobile nfc in transport. Technical report, 2012. Available at `http://www.smart-ticketing.org/downloads/papers/Mobile_NFC_in_Transport.pdf`.

[GSM19] GSMA. Sgp.02 - remote provisioning architecture for embedded uicc technical specification. Technical report, 2019.

[HABJ] Grant Hernandez, Orlando Arias, Daniel Buentello, and Yier Jin. Smart nest thermostat: A smart spy in your home.

[HCDF06] Thomas S. Heydt-Benjamin, Hee-Jin Chae, Benessa Defend, and Kevin Fu. Privacy for public transportation. In *Privacy Enhancing Technologies, 6th International Workshop, PET 2006, Cambridge, UK, June 28-30, 2006, Revised Selected Papers*, 2006.

[Hes02] Florian Hess. Efficient identity based signature schemes based on pairings. In *International Workshop on Selected Areas in Cryptography*, pages 310–324. Springer, 2002.

[HMR+19] Carmit Hazay, Gert Læssøe Mikkelsen, Tal Rabin, Tomas Toft, and Angelo Agatino Nicolosi. Efficient RSA key generation and threshold Paillier in the two-party setting. *Journal of Cryptology*, 32(2):265–323, 2019.

[HPJ03] Y-C Hu, Adrian Perrig, and David B Johnson. Packet leashes: a defense against wormhole attacks in wireless networks. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, volume 3, pages 1976–1986. IEEE, 2003.

[HPVP11] Jens Hermans, Andreas Pashalidis, Frederik Vercauteren, and Bart Preneel. A new RFID privacy model. In *European symposium on research in computer security*, pages 568–587. Springer, 2011.

[HYKD14] Charles Herder, Meng-Day (Mandel) Yu, Farinaz Koushanfar, and Srinivas Devadas. Physical unclonable functions and applications: A tutorial. *Proceedings of the IEEE*, 102(8):1126–1141, 2014.

[IEE16a] IEEE. IEEE Standard for Wireless Access in Vehicular Environments–Security Services for Applications and Management Messages. Technical Report (TR) 1609.2b-2019, Institute of Electrical and Electronics Engineers (IEEE) Standard Association, 01 2016. Version 1609.2b-2019.

[IEE16b] IEEE. Ieee standard for wireless access in vehicular environments (wave) 1609.12-2016. Technical report, IEEE, 2016. Available at =https://standards.ieee.org.

[ISO] ISO/IEC. Information security — lightweight cryptography — part 2: Block ciphers. Technical report (tr), ISO/IEC 29192-2:2019, 11. Available at `https://www.iso.org/standard/78477.html`.

[ISO13a] ISO/IEC. Information technology — security techniques — anonymous digital signatures — part 2: Mechanisms using a group public key: 20008-2. Technical report, ISO/IEC, 2013.

[ISO13b] ISO/IEC. Information technology — telecommunications and information exchange between systems — near field communication — interface and protocol (nfcip-1). Technical report, ISO/IEC 18092:2013, 2013.

[ISO18] ISO/IEC. Cards and security devices for personal identification — contactless proximity objects — part 1: Physical characteristics. Technical report, ISO/IEC 14443-1:2018, 2018.

[ISO19] ISO/IEC. Identification cards — integrated circuit cards — part 8: Commands and mechanisms for security operations. Technical report (tr), ISO/IEC 7816-8:2019, 2019.

[ISTZ16] Ahmad Ibrahim, Ahmad-Reza Sadeghi, Gene Tsudik, and Shaza Zeitouni. DARPA: device attestation resilient to physical attacks. In *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks, WISEC 2016, Darmstadt, Germany, July 18-22, 2016*, pages 171–182, 2016.

[ISZ17] Ahmad Ibrahim, Ahmad-Reza Sadeghi, and Shaza Zeitouni. Seed: secure non-interactive attestation for embedded devices. In *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec 2017, Boston, MA, USA, July 18-20, 2017*, pages 64–74, 2017.

[JBC16] Andreas Jacobsson, Martin Boldt, and Bengt Carlsson. A risk analysis of a smart home automation system. *Future Generation Computer Systems*, 56:719–733, 2016.

[JD08] Daniel Jiang and Luca Delgrossi. Ieee 802.11 p: Towards an international standard for wireless access in vehicular environments. In *VTC Spring 2008-IEEE Vehicular Technology Conference*, pages 2036–2040. IEEE, 2008.

[Jou00] Antoine Joux. A one round protocol for tripartite diffie–hellman. In *International algorithmic number theory symposium*, pages 385–393. Springer, 2000.

[JS08] Tibor Jager and Jörg Schwenk. On the equivalence of generic group models. In *International Conference on Provable Security*, pages 200–209. Springer, 2008.

[JTM+06] D. Jiang, V. Taliwal, A. Meier, W. Holfelder, and R. Herrtwich. Design of 5.9 ghz dsrc-based vehicular safety communication. *Wireless Commun.*, 13(5):36–43, October 2006. Available at `https://doi.org/10.1109/WC-M.2006.250356`.

[KBGK17] Florian Kohnhäuser, Niklas Büscher, Sebastian Gabmeyer, and Stefan Katzenbeisser. SCAPI: a scalable attestation protocol to detect software and physical attacks. In *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec 2017, Boston, MA, USA, July 18-20, 2017*, pages 75–86, 2017.

[KBK18] Florian Kohnhäuser, Niklas Büscher, and Stefan Katzenbeisser. SALAD: secure and lightweight attestation of highly dynamic and disruptive networks. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security, AsiaCCS 2018, Incheon, Republic of Korea, June 04-08, 2018*, pages 329–342, 2018.

[Ker83] Auguste Kerckhoffs. *La cryptographie militaire, ou, Des chiffres usités en temps de guerre: avec un nouveau procédé de déchiffrement applicable aux systèmes à double clef*. Librairie militaire de L. Baudoin, 1883.

[KL08] Jonathan Katz and Andrew Y. Lindell. Aggregate message authentication codes. In *Topics in Cryptology - CT-RSA 2008, The Cryptographers' Track at the RSA Conference 2008, San Francisco, CA, USA, April 8-11, 2008. Proceedings*, pages 155–169, 2008.

[KLL+18] Vireshwar Kumar, He Li, Noah Luther, Pranav Asokan, Jung-Min Park, Kaigui Bian, Martin BH Weiss, and Taieb Znati. Direct anonymous attestation with efficient verifier-local revocation for subscription system. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, pages 567–574, 2018.

[Kob87] Neal Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209, 1987.

[KSH16] M. Khan, B. N. Silva, and K. Han. Internet of things based energy aware smart home control system. *IEEE Access*, 4:7556–7566, 2016.

[KSSV14] Patrick Koeberl, Steffen Schulz, Ahmad-Reza Sadeghi, and Vijay Varadharajan. Trustlite: A security architecture for tiny embedded devices. In *Proceedings of the Ninth European Conference on Computer Systems*, page 10. ACM, 2014.

[LRSW99] Anna Lysyanskaya, Ronald L. Rivest, Amit Sahai, and Stefan Wolf. Pseudonym systems. In *Selected Areas in Cryptography, 6th Annual International Workshop, SAC'99*, pages 184–199, 1999.

[LXL+12] Jing Liu, Yang Xiao, Shuhui Li, Wei Liang, and CL Philip Chen. Cyber security and privacy issues in smart grids. *IEEE Communications Surveys & Tutorials*, 14(4):981–997, 2012.

[MAB+13a] Frank McKeen, Ilya Alexandrovich, Alex Berenzon, Carlos V. Rozas, Hisham Shafi, Vedvyas Shanbhogue, and Uday R. Savagaonkar. Innovative instructions and software model for isolated execution. In *Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy*, New York, NY, USA, 2013. Association for Computing Machinery. Available at `https://doi.org/10.1145/2487726.2488368`.

[MAB+13b] Frank McKeen, Ilya Alexandrovich, Alex Berenzon, Carlos V Rozas, Hisham Shafi, Vedvyas Shanbhogue, and Uday R Savagaonkar. Innovative instructions and software model for isolated execution. *Hasp@ isca*, 10(1), 2013.

[Mac17] Ken MacKay. MICRO-ECC library. 2017. Available at `https://github.com/kmackay/micro-ecc`.

[Mau05] Ueli Maurer. Abstract models of computation in cryptography. In *IMA International Conference on Cryptography and Coding*, pages 1–12. Springer, 2005.

[MBM+17] Cédric Marchand, Lilian Bossuet, Ugo Mureddu, Nathalie Bochard, Abdelkarim Cherkaoui, and Viktor Fischer. Implementation and characterization of a physical unclonable function for iot: a case study with the tero-puf. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(1):97–109, 2017.

[MBSM16] Kerry A. McKay, Larry Bassham, Meltem Sönmez, and Turan Nicky Mouha. Report on lightweight cryptography. Technical report (tr), NIST - Computer Security Division Information Technology Laboratory, 08 2016. Available at `https://csrc.nist.gov/CSRC/media/Publications/nistir/8114/draft/documents/nistir_8114_draft.pdf`.

[MD16] Steve Mansfield-Devine. DDoS goes mainstream: how headline-grabbing attacks could make this threat an organisation's biggest nightmare. *Network Security*, 2016.

[Mil85] Victor S Miller. Use of elliptic curves in cryptography. In *Conference on the theory and application of cryptographic techniques*, pages 417–426. Springer, 1985.

[MPSW19] Gregory Maxwell, Andrew Poelstra, Yannick Seurin, and Pieter Wuille. Simple schnorr multi-signatures with applications to bitcoin. *Designs, Codes and Cryptography*, 87(9):2139–2164, 2019.

[MR12] Bill Miller and Dale Rowe. A survey SCADA of and critical infrastructure incidents. In *Proceedings of the 1st Annual conference on Research in information technology*, pages 51–56, 2012.

[MS11] Abhranil Maiti and Patrick Schaumont. Improved ring oscillator puf: An fpga-friendly secure primitive. *Journal of cryptology*, 24(2):375–397, 2011.

[MT07] Di Ma and Gene Tsudik. Forward-secure sequential aggregate authentication. *IACR Cryptology ePrint Archive*, 2007.

[MV14] Charlie Miller and Chris Valasek. A survey of remote automotive attack surfaces. *black hat USA*, 2014:94, 2014.

[Nao91] Moni Naor. Bit commitment using pseudorandomness. *Journal of cryptology*, 4(2):151–158, 1991.

[Nao03] Moni Naor. On cryptographic assumptions and challenges. In *Annual International Cryptology Conference*, pages 96–109. Springer, 2003.

[NOVY92] Moni Naor, Rafail Ostrovsky, Ramarathnam Venkatesan, and Moti Yung. Perfect zero-knowledge arguments for np can be based on general complexity assumptions. In *Annual International Cryptology Conference*, pages 196–214. Springer, 1992.

[OPC19] OPCC. The personal information protection and electronic documents act (pipeda), 2019. Available at `https://www.priv.gc.ca/index_f.asp`.

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, pages 223–238, 1999.

[PBH+08] Panagiotis Papadimitratos, Levente Buttyan, Tamás Holczer, Elmar Schoch, Julien Freudiger, Maxim Raya, Zhendong Ma, Frank Kargl, Antonio Kung, and Jean-Pierre Hubaux. Secure vehicular communication systems: design and architecture. *IEEE Communications Magazine*, 46(11):100–109, 2008.

[Ped91] Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*, pages 129–140, 1991.

[PH12] Roel Peeters and Jens Hermans. Wide strong private RFID identification based on zero-knowledge. *IACR Cryptol. ePrint Arch.*, 2012:389, 2012.

[PIP16] PIPC. Act on the protection of personal information, 2016. Available at `https://www.ppc.go.jp/en/`.

[PKHK06] Panagiotis Papadimitratos, Antonio Kung, Jean-Pierre Hubaux, and Frank Kargl. Privacy and identity management for vehicular communication systems: a position paper. In *Workshop on standards for privacy in user-centric identity management*, number CONF, 2006.

[Pol10] Jonathan Pollet. Electricity for Free - The Dirty Underbelly of SCADA and Smart Meters, 2010.

[PP04] Duong Hieu Phan and David Pointcheval. On the security notions for public-key encryption schemes. In *International Conference on Security in Communication Networks*, pages 33–46. Springer, 2004.

[PS00] David Pointcheval and Jacques Stern. Security arguments for digital signatures and blind signatures. *Journal of cryptology*, 13(3):361–396, 2000.

[PS16] David Pointcheval and Olivier Sanders. Short randomizable signatures. In *Cryptographers' Track at the RSA Conference*, pages 111–126, 2016.

[PS18] David Pointcheval and Olivier Sanders. Reassessing security of randomizable signatures. In *Topics in Cryptology - CT-RSA 2018 - The Cryptographers' Track at the RSA Conference 2018, San Francisco, CA, USA, April 16-20, 2018, Proceedings*, pages 319–338, 2018.

[PT10] Daniele Perito and Gene Tsudik. Secure code update for embedded devices via proofs of secure erasure. In *European Symposium on Research in Computer Security*, pages 643–662. Springer, 2010.

[Res00] Certicom Research. SEC 2: Recommended elliptic curve domain parameters. 2000. Available at `https://www.secg.org/SEC2-Ver-1.0.pdf`.

[RHBP13] Andy Rupp, Gesine Hinterwälder, Foteini Baldimtsi, and Christof Paar. P4R: privacy-preserving pre-payments with refunds for transportation systems. In *Financial Cryptography and Data Security - 17th International Conference, FC 2013, Okinawa, Japan, April 1-5, 2013, Revised Selected Papers*, 2013.

[RPH06] Maxim Raya, Panos Papadimitratos, and Jean-Pierre Hubaux. Securing vehicular communications. *IEEE wireless communications*, 13(5):8–15, 2006.

[RWI+09] Alastair Ruddle, Benjamin Weyl, Sajid Idrees, Y. Roudier, Michael Friedewald, Timo Leimbach, A. Fuchs, S. Gürgens, O. Henninger, Roland Rieke, M. Ritscher, H. Broberg, L. Apvrille, R. Pacalet, and Gabriel Pedroza. Security requirements for automotive on-board networks based on dark-side scenarios. deliverable d2.3: Evita. e-safety vehicle intrusion protected applications. *Fraunhofer ISI*, 01 2009.

[Sch89] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In *Conference on the Theory and Application of Cryptology*, pages 239–252. Springer, 1989.

[Sch91] Claus-Peter Schnorr. Efficient signature generation by smart cards. *J. Cryptology*, 4(3):161–174, 1991.

[Sha49] Claude E Shannon. Communication theory of secrecy systems. *The Bell system technical journal*, 28(4):656–715, 1949.

[Sha71] Daniel Shanks. Class number, a theory of factorization, and genera. In *Proc. of Symp. Math. Soc., 1971*, volume 20, pages 41–440, 1971.

[Sha84a] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Workshop on the theory and application of cryptographic techniques*, pages 47–53. Springer, 1984.

[Sha84b] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in Cryptology, Proceedings of CRYPTO'84*, 1984.

[Sho04] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. *IACR Cryptology ePrint Archive*, 2004:332, 2004.

[SKAZ19] Alireza Shamsoshoara, Ashwija Korenda, Fatemeh Afghah, and Sherali Zeadally. A survey on hardware-based security mechanisms for internet of things. *arXiv preprint arXiv:1907.12525*, 2019.

[SRR16] Soubhagya Sutar, Arnab Raha, and Vijay Raghunathan. D-puf: An intrinsically reconfigurable dram puf for device authentication in embedded systems. In *2016 International Conference on Compliers, Architectures, and Sythesis of Embedded Systems (CASES)*, pages 1–10. IEEE, 2016.

[SVW08] Ahmad-Reza Sadeghi, Ivan Visconti, and Christian Wachsmann. User privacy in transport systems based on RFID e-tickets. In *Proceedings of the 1st International Workshop on Privacy in Location-Based Applications, October 9, 2008*, 2008.

[SWW15] Ahmad-Reza Sadeghi, Christian Wachsmann, and Michael Waidner. Security and privacy challenges in industrial internet of things. In *2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2015.

[TE13] Sandeep Tamrakar and Jan-Erik Ekberg. Tapping and tripping with nfc. In *International Conference on Trust and Trustworthy Computing*, pages 115–132. Springer, 2013.

[TPM19] Trusted platform module library part 1: Architecture family "2.0" level 00 revision 01.59. Technical report (tr), Trusted Computing Group (TCG), 11 2019. Available at `https://trustedcomputinggroup.org/resource/tpm-library-specification/`.

[USD17] USDC. Eu-u.s. and swiss-u.s. privacy shield frameworks, 2017. Available at `https://www.privacyshield.gov/Program-Overview`.

[Wal15] John Walker. Intelligent transportation systems report for mobile. Technical report, 2015. Available at `https://www.gsma.com/iot/wp-content/uploads/2015/06/ITS-report.pdf`.

[WCG+17] Jorden Whitefield, Liqun Chen, Thanassis Giannetsos, Steve Schneider, and Helen Treharne. Privacy-enhanced capabilities for vanets using direct anonymous attestation. In *2017 IEEE Vehicular Networking Conference (VNC)*, pages 123–130. IEEE, 2017.

[WCS+19] Jorden Whitefield, Liqun Chen, Ralf Sasse, Steve Schneider, Helen Treharne, and Stephan Wesemeyer. A symbolic analysis of ecc-based direct anonymous attestation. In *Proceedings of the 4th IEEE European Symposium on Security and Privacy*. Institute of Electrical and Electronics Engineers (IEEE), 2019.

[Win08] Johannes Winter. Trusted computing building blocks for embedded linux-based arm trustzone platforms. In *Proceedings of the 3rd ACM workshop on Scalable trusted computing*, pages 21–30, 2008.

[WNT+19] Stephan Wesemeyer, Christopher JP Newton, Helen Treharne, Liqun Chen, Ralf Sasse, and Jorden Whitefield. Formal analysis and implementation of a tpm 2.0-based direct anonymous attestation scheme. 2019.

[ZCNC11] Tong Zhou, Romit Roy Choudhury, Peng Ning, and Krishnendu Chakrabarty. P2dap—sybil attacks detection in vehicular ad hoc networks. *IEEE journal on selected areas in communications*, 29(3):582–594, 2011.

[ZG13] Kai Zhao and Lina Ge. A survey on the internet of things security. In *2013 Ninth international conference on computational intelligence and security*, pages 663–667. IEEE, 2013.

List of Tables

TABLE **Page**

List of Figures

FIGURE **Page**

INSTITUT POLYTECHNIQUE DE PARIS

Titre : Protocoles Cryptographiques pour l'Authentification et l'Attestation Anonymes dans l'Internet des Objets

Mots clés : Internet des Objets, Authentification, Attestation, Sécurité Prouvée

Résumé : L'Internet des Objets (IdO) et le Machine-to-Machine (M2M) ont révolutionné l'architecture de sécurité des systèmes de communication. Dans cette nouvelle configuration, la sécurité et la protection des données à caractère personnel échangées puis stockées par ces systèmes est devenu un enjeu primordial. Ces objets présentent des contraintes physiques fortes impactant leurs fonctionnalités, notamment en termes de capacité de calcul et de mémoire, d'énergie, et d'exigences de sécurité en fonction du cas d'usage et de l'application concernés. Nous nous intéresserons dans cette thèse aux problématiques liées à l'intégrité des communications entre les objets et les systèmes embarqués pour différents cas d'usages dans l'Internet des Objets, ainsi qu'à l'intégrité des micro-logiciels des dits systèmes. En d'autres termes, nous nous intéressons aux mécanismes d'authentification et d'attestation adaptés pour des environnements contraints.
Cette thèse se focalise sur le développement de mécanismes d'authentification et d'attestation anonymes et efficaces. Nous introduisons dans une première partie deux protocoles d'authentification se basant sur une nouvelle primitive d'attestation anonyme (pre-Direct Anonymous Attestation (pre-DAA)), pour les cas d'usages des véhicules connectés et d'authentification anonyme pour les pass de transports mobiles. Notre nouvelle primitive de pre-DAA est prouvée sûr dans le modèle de l'oracle aléatoire, sous une variante de l'hypothèse q-SDH. Nous développons dans un premier temps un protocole de communication décentralisé et anonyme pour les réseaux de véhicules connectés (VANET). Nous proposons ensuite un nouveau protocole de contrôle d'accès anonyme sur mobile pour les réseaux de transport publiques. Notre protocole permet à un usager de s'authentifier sur le réseau, sans pouvoir être tracé par l'opérateur de transport publique. La deuxième partie de cette thèse se concentre ensuite sur le développement d'un mécanisme d'attestation d'essaims d'objets, avec une propriété de détection lorsque l'un des objets fournit une fausse attestation.

Title : Cryptographic Mechanisms for Device Authentication and Attestation in the Internet of Things

Keywords : Internet of Things, Authentication, Attestation, Provable Security

Abstract : Internet of Things (IoT) applications are pervasive and present in the vast majority of industries, with the aim of increasing efficiency and safety in industrial processes and consumer-driven applications. IoT devices have recently become the target of malware attacks due to their limited computational capacities, limited energy source, and limited memory. Most notably, two of the main challenges consist in securing the communication between IoT devices, and ensuring that devices in the network have not been compromised or tampered with, thus attesting of the integrity of the entire network. In addition, new privacy concerns affecting users in IoT applications have risen, and require implementing privacy-friendly authentication and attestation mechanisms. Authentication mechanisms allow systems to identify themselves on the network, and provide solutions for the first challenge. Remote Attestation is a security mechanism which enables control systems to verify the software state of devices in the network, thus detecting any tampering or remote malware injection attacks.
In this thesis, we aim to contribute to the development of new and privacy-preserving authentication and attestation mechanisms, which are particularly adapted for implementation in constrained environments.
In the first part of this thesis, we leverage a cryptographic mechanism deployed in trusted computing, namely Direct Anonymous Attestation (DAA), in order to provide decentralized, and privacy-preserving authentication protocols adapted for constrained environments. The first application of our pre-DAA scheme consists in the design of a decentralized architecture for secure communication in vehicular ad hoc networks (VANETs), which removes the need for a centralize Public Key Infrastructure. The second application of our pre-DAA scheme is the design of a mobile-based access control protocol for public transport systems, which addresses the issue of user traceability inherent to current access control control protocols for transport systems.
In the second part of this thesis, we address the device integrity verification challenge by designing a remote attestation protocol which enables the secure and efficient attestation of groups (or swarms) of devices.

Institut Polytechnique de Paris
91120 Palaiseau, France

INSTITUT POLYTECHNIQUE DE PARIS