Privacy in RFID Systems

Tania Martin

Thesis submitted in partial fulfillment of the requirements for the Degree of Doctor in Applied Sciences

June 2013

Institute of Information & Communication Technologies,
Electronics and Applied Mathematics
(ICTEAM institute)
Université catholique de Louvain
Louvain-la-Neuve
Belgium

Thesis Committee:

Gildas **Avoine** (Advisor)	UCL/ICTM/INGI
Olivier **Bonaventure**	UCL/ICTM/INGI
Flavio **Garcia**	University of Birmingham, UK
Charles **Pecheur** (Chair)	UCL/ICTM/INGI
Olivier **Pereira**	UCL/ICTM/ELEN
Serge **Vaudenay**	EPFL, Switzerland

Abstract

Radio-Frequency IDentification (RFID) is the current trend that allows the identification and/or authentication of objects or persons without physical contact. The rise of wireless systems based on RFID has brought up major concerns on privacy. Indeed nowadays, when such a system is deployed, informed customers yearn for guarantees that their privacy will not be threatened. One formal way to perform this task is to evaluate the privacy level of the RFID system with an adversary model. However, if the chosen model does not reflect the assumptions and requirements of the analyzed system, it may misevaluate its privacy level. Therefore, selecting the most appropriate model among all the existing ones is not an easy task. In parallel, authentication for RFID is a booming research topic, where the challenge is to develop secure protocols using the most lightweight cryptography, yet ensuring privacy. This led to the publication of hundred of RFID authentication protocols during the last decade. This thesis investigates the problems of privacy in RFID systems and the solutions to assess their privacy level.

The first step of this research is the thorough analysis of the eight most well-known RFID privacy models, which demonstrates that none of these models is comprehensive enough to compare protocols meaningfully. Subsequently, further investigations on data protection give rise to two new kinds of attack that threaten the privacy-friendliness of RFID protocols, namely time attacks and compromised readers. These results lead to the proposal of a new privacy model that is operational where the previous ones were not. Finally, the thesis addresses the privacy question in broader IT environments, namely ubiquitous computing systems, and lays the foundation stones in the development of a standardized privacy certification that would assess the privacy level of such systems.

Remerciements

Force est de constater que la rédaction des remerciements est une étape inévitable et délicate qui sonne la fin de cinq années d'une vie. Ces quelques lignes sont le témoignage de ma gratitude envers toutes les personnes qui m'ont épaulée, encouragée et réconfortée, tant professionnellement que personnellement, depuis mon arrivée en Belgique en 2008.

Tout d'abord, je tiens à remercier de tout mon coeur la personne qui m'a permis de réaliser cette thèse, mon promoteur, "le chef" : Gildas Avoine. Son investissement dans ma recherche m'a apporté la confiance et la motivation nécessaire pour ne jamais lâcher le morceau et toujours tenir bon, même dans les durs moments de doutes. Il a été un formidable mentor qui a su me guider et me conseiller durant tout le parcours de ma thèse. Merci à Gildas, pour tout.

Mes remerciements s'adressent également aux membres de mon jury qui, parfois sans le savoir, ont participé directement et indirectement à l'aboutissement de ma thèse. C'est donc avec beaucoup d'humilité que je remercie Serge Vaudenay de l'EPFL et Flavio Garcia de l'Université de Birmingham, ainsi qu'Olivier Bonaventure, Olivier Pereira et Charles Pecheur de l'UCL.

Comme il n'y a pas de thèse sans publication, et pas de publication sans coauteur, je souhaite aussi remercier ces derniers. Tout d'abord, merci à la compagnie RFIDea pour son aide sur le problème des lecteurs compromis. Merci à Cédric Lauradoux qui m'a appris à rédiger un article et à me motiver à faire du sport, tant cérébral que physique. Merci à Jean-Pierre Szikora pour sa gentillesse, ses nombreuses connaissances sur les cartes à puce et son aide précieuse sur ma compréhension technique de la RFID. Si vous ne savez pas comment vous occuper un long dimanche pluvieux, lui le saura ! Merci à Benjamin Martin pour sa créativité et

son grand savoir mathématique. Merci à Xavier Carpent pour sa vivacité de réflexion et son esprit critique. Enfin, un grand merci en particulier à Iwen Coisel pour les différentes recherches effectuées ensemble. Son expérience, ses idées, sa diplomatie et son soutien m'ont permis de donner le meilleur de moi-même. J'aurai aussi appris qu'il ne faut pas négliger les heures du repas de midi !

Je ne me serais pas non plus lancée dans cette thèse sans ma formation universitaire. C'est pourquoi je tiens aussi à remercier les professeurs de l'Université Bordeaux 1 pour m'avoir fait découvrir et aimé la cryptologie : Gilles Zémor, Christine Bachoc, Karim Belabas, Henri Cohen et Emmanuel Fleury sans qui je n'aurais pas rencontré Gildas.

Durant ma thèse à l'UCL, j'ai pu également faire la connaissance de personnalités très intéressantes. Dès mon arrivée, le Sindaca m'a prise sous son aile et je l'en remercie profondément. Merci à Seb pour ses connaissances précieuses en LaTeX, Simon pour sa patience lorsque je venais trop souvent embêter Seb, et Florence pour avoir partagé nos pauses cigarette. Merci à Adrien, Chong Hee, Jonathan et Rolando avec qui j'ai passé de très bons moments durant ma thèse. Merci à Sylvie Baudine du Groupe Crypto pour avoir relu et corrigé bon nombre de mes articles. Je remercie aussi tous les membres du Pôle INGI qui ne tournerait pas aussi bien sans eux, et en particulier le personnel administratif et technique pour leur aide et convivialité : ils font un travail formidable.

Pendant ces quelques années d'expat en Belgique, de grandes et fortes amitiés se sont créées. Tout d'abord, j'ai été ravie de faire la connaissance du "groupe des jeunes" qui sait mettre l'ambiance où qu'il soit : JB, Fanfwé, Xa, Jey, Sam, Ka, Nico et Romain. Il y a aussi les "zamis du vendredi" avec qui j'ai libéré Cuba de nombreuses fois : Mel, Cath, Marco, Marinette et Daminou (le plus français des Belges). Je remercie Flo pour toute sa tendresse et patience, et les bons moments à venir. Enfin, mes derniers remerciements vont à Benjamin, a.k.a. "Benj", "le poulet" ou encore "le rasta chauve", pour avoir partagé beaucoup de temps forts avec moi depuis quatre ans, mais surtout le bureau, les questions existentielles, le reggae, Zion, les rendez-vous hebdomadaires avec Grant et son amitié.

Pour finir, je dédicace cette thèse à ma famille et tout particulièrement à mes parents qui ont toujours été là pour moi et m'ont soutenue par tous les moyens possibles et imaginables.

Acknowledgements

The drafting of acknowledgements is an inevitable and important step that signals the end of five years of dedication. These few lines are the testimony of my gratitude to all those who have supported, encouraged and comforted me, both professionally and personally, since my arrival in Belgium in 2008.

First of all, I want to thank the person who allowed me to realize this thesis with all my heart, my advisor, "the boss": Gildas Avoine. His investment in my research has brought me the confidence and motivation to never give up and always keep going on, even in the hard times of doubt. He has been a formidable mentor who guided and advised me throughout the course of my thesis. Thank you Gildas, for everything.

My gratitude also goes to the members of my jury who, sometimes without knowing it, directly and indirectly participated to the fulfillment of my thesis. It is with high humility that I thank Serge Vaudenay from the EPFL and Flavio Garcia from the University of Birmingham, and Olivier Bonaventure, Olivier Pereira and Charles Pecheur from the UCL.

As there is no thesis without publication, and no publication without coauthor, I also want to thank them. First of all, I thank the company RFIDea for its help on the problem of compromised readers. I thank Cédric Lauradoux for having taught me how to write an article and for having motivated me to do sports, whether cerebral or physical. I thank Jean-Pierre Szikora for his kindness, knowledge on smartcards, and valuable help on my technical understanding of RFID. If you do not know how to spend a long rainy Sunday, he does! I thank Benjamin Martin for his great creativity and mathematical knowledge. I thank Xavier Carpent for his sharp mindedness and critical thinking. Finally, I particularly thank Iwen Coisel for the various research that we carried

out together. His experience, ideas, diplomacy and support allowed me to give the best of myself. I have also learned that lunch times should never be neglected!

I would never have started this thesis without going through my university education. That is why I also want to thank the professors of the University of Bordeaux 1 for making me discover and love cryptology: Gilles Zémor, Christine Bachoc, Karim Belabas, Henri Cohen and Emmanuel Fleury without whom I would not have met Gildas.

During my PhD at the UCL, I also had the opportunity to get to know very interesting people and build long lasting friendships. Upon my arrival, the Sindaca took me under its wing and I deeply thank it. I thank Seb for his valuable knowledge in LaTeX, Simon for his patience when I too oftenly came to bother Seb, and Florence for our cigarette breaks. I thank Adrien, Chong Hee, Jonathan and Rolando with whom I spent a great time during my PhD. I thank Sylvie Baudine from the Crypto Group for having proofread many of my articles. I also thank all the members of the INGI Pole which would not operate as well without them, especially the administrative and technical staff for their help and friendliness: they do a great job.

During these few years as an expat in Belgium, strong friendships were born. Firstly, I was delighted to get to know the "group of the young guys" who knows how to liven things up anywhere: JB, Fanfwé, Xa, Jey, Sam, Ka, Nico and Romain. There are also the "Friday friends" with whom I freed Cuba many times: Mel, Cath, Marco, Marinette and Daminou (the Frenchiest Belgian guy). I also want to thank Flo for all his tenderness and patience, and the future good times. Finally, my last thanks goes to Benjamin, aka "Benj", "le poulet" or "the bold rasta", for having shared many highlights with me for four years, but especially the office, existential questions, the reggae music, Zion, the weekly meetings with Grant, and his friendship.

Finally, I dedicate this thesis to my family and especially my parents who have always been there for me and supported me in every possible and imaginable way.

List of Publications

Scientific Publications

- Iwen Coisel and Tania Martin. Untangling RFID Privacy Models. In *Journal of Computer Networks and Communications*, July 2012. Hindawi.

- Gildas Avoine, Iwen Coisel and Tania Martin. A Privacy-Restoring Mechanism for Offline RFID Systems. In *5th ACM Conference on Wireless Network Security – WiSec 2012*. Tucson, AZ, USA, April 2012. ACM.

- Gildas Avoine, Benjamin Martin and Tania Martin. Tree-Based RFID Authentication Protocols Are Definitively Not Privacy-Friendly. In *6th International Workshop on RFID Security – RFIDSec 2010*. Istanbul, Turkey, June 2010. Springer.

- Gildas Avoine, Iwen Coisel and Tania Martin. Time Measurement Threatens Privacy-Friendly RFID Authentication Protocols. In *6th International Workshop on RFID Security – RFIDSec 2010*. Istanbul, Turkey, June 2010. Springer.

- Gildas Avoine, Cédric Lauradoux and Tania Martin. When Compromised Readers Meet RFID. In *10th International Workshop on Information Security Applications – WISA 2009*. Busan, Korea, August 2009. Springer.
  (Extended version published at *RFIDSec 2009*).

Publications for the General Public

- Gildas Avoine, Xavier Carpent, Benjamin Martin and Tania Martin. Chapitre X: la Sécurité du Sans Contact et ses Specificités (trad.: "Chapter X: the Security of Contactless Technology and its Specificities"). In *La Carte à Puce, Vecteur de Système de Confiance*, to appear, 2013. HERMES Science.

- Gildas Avoine, Tania Martin et Jean-Pierre Szikora. Lire son Passe Navigo en un Clin d'Oeil (trad.: "How to Read your Navigo Pass in a Flash"). In *Multi-system & Internet Security Cookbook (MISC) Magazine, No.48*, March-April 2010. Diamond.

- Gildas Avoine, Tania Martin et Jean-Pierre Szikora. NXP Mifare Classic: une Star Déchue (trad.: "NXP Mifare Classic: a Fallen Star"). In *Multi-system & Internet Security Cookbook (MISC) Magazine, Special Issue No.2*, November-December 2008. Diamond.

Softwares

- **MOBIB Extractor** : free software to read the transit pass of the Brussels public transportation company (STIB).
  `http://www.uclouvain.be/sites/security/mobib.html`

Contents

Introduction

Radio Frequency IDentification (RFID) is the current trendy wireless technology that allows the identification and/or authentication of objects or persons without physical contact. In a nutshell, *tags* are embedded into objects and can communicate with *readers* through radio frequency channels. The first use of RFID goes back to the early 1940's, during World War II, when the Royal Air Force deployed the IFF (Identify Friend or Foe) system to identify the Allies airplanes [4, 54, 163]. Today, RFID can be found in many daily-life applications such as access control, anti-theft cars, anti-counterfeiting, library management, pet identification, or even electronic passports, to name a few. In mobile environments, public transportation and mass events also take advantage of this technology to increase both security and customer flows.

Context of the Thesis

The main goal of RFID is *identification*, i.e., readers should retrieve the identity of the tags communicating with them. For some applications such as pet identification or product tracking, this clearly defines the purpose of the system. For other more secure applications like access control or payment, readers need to obtain evidence of the identity. This concept is known as *authentication*. Depending on the application, other (often post-authentication) operations are also possible, e.g., reading or updating data stored on the tag.

One of the most important security challenges in RFID is to ensure safe and effective authentication given the various hardware restrictions on tags (e.g., low calculation capacity, low available energy, limited storage, etc.) and the – potentially high – number of tags in a system.

Although this is an active area of research and solutions already exist, RFID-based applications are continuously requested to provide more functionalities and more security without increasing tags capabilities.

As predictable, many problems come up with the large-scale deployment of this technology. From the security point of view, the main drawbacks of RFID can be summarized as follow: (i) the entities communicate using radio waves, and (ii) tags generally respond to any query without the explicit consent of its holder. These two facts simplify the life of an "adversary"[1] who can then easily listen or trigger communications. Note that the forward channel (reader to tag) is sometimes differentiated from the backward channel (tag to reader) in security analyses because the signal of the former is generally much more powerful and easier to listen than the signal of the latter [80]. However, it is generally accepted that none of these channels is secure against eavesdropping.

Practical studies also showed that it is possible to recover the information stored in a tag by analyzing its chip with an electron microscope. This attack was for example applied to the Mifare Classic (see [18] for a survey on the attacks performed on this tag). Although this is not within the reach of anybody, it is generally accepted that an adversary may be able to collect a limited number of tags in order to obtain the information they contain.

According to these drawbacks, the fundamental security requirements that should be ensured at the application level of an RFID system are summarized below.

Soundness. An adversary should not be able to impersonate a legitimate tag (i.e., that is part of the system) when she is interrogated by a reader of the system. For instance, a system should prevent an adversary from guessing the secret data used by a legitimate tag when it is communicating with a reader, e.g., by querying this tag beforehand with her own reader or, when possible, by eavesdropping a legitimate interaction of this tag. This property is the one that systems based on authentication seek to provide.

[1] In the literature related to security and cryptography, the term "adversary" is generally used to refer to any entity or individual with malicious purpose vis-à-vis the system.

Availability. An adversary should not be able to make a system partially or completely unusable. There are many ways to achieve such attacks at the physical and communication levels[2] of an RFID system. For instance, a system should prevent an adversary from modifying the secrets stored on a tag that are used to get authenticated by legitimate readers.

Privacy. An adversary should not be able to extract sensitive data about a tag from its emitted messages, or to track a tag at different places or times by eavesdropping its interactions. For instance, a system should prevent an adversary from distinguishing two tags (whose identities are not known by the adversary) by observing their outputs. Note that privacy can also be studied at the physical and communication levels of an RFID system (see [20, 45] for more details).

In 2002, Sarma, Weis, and Engels have been among the first authors to mention in [148] the potential privacy issues, particularly traceability, as new substantial security threats for RFID systems. In the broad sense of the word, "privacy" commonly links to individuals, not to objects, and represents their ability to protect their personal data:

> "*Privacy may be defined as the right of the individual to determine for himself when, how, and to what extent he will release personal information about himself*" [130].

Yet, the growing of RFID-based applications and services gave rise to a major controversy related to privacy from customers. Indeed in many systems, tags are embedded into items carried by persons. Consequently, if tags can be read without the explicit consent of their holder, they may reveal data that would threaten the privacy of their holder. For example, these data can be stored, associated to the holder, or even put in relation with other information.

[2]For example, physical attacks can simply destroy or damage the resources of the system. By the nature of the technology, radio waves can also be used to disrupt communications between readers and tags (e.g., with an RFID jammer that introduces electromagnetic noise), or even destroy them (e.g., with an RFID zapper that emits a electromagnetic signal powerful enough to cause a burn-out). These types of attack are discarded in this thesis, since no RFID system provides under such attacks.

Nowadays, privacy is so important that it is unconceivable to widespread an IT solution without addressing the privacy issues. The dangers of RFID with respect to privacy have been clearly pointed out, and the authorities are now aware of this problem. For instance, Ontario Information and Privacy Commissioner Cavoukian aims to advocate the concept of "privacy-by-design" [39] which states that privacy should be put in place in every IT system before its widespread use. Privacy Rights Clearinghouse, an American non-profit consumer education and advocacy group, also publishes many fact sheets on privacy problems [140]. In 2009 in the European Union, Viviane Reding, Commissioner for Justice, Fundamental Rights and Citizenship, signed the Recommendation 2009/387/EC [60] which strongly supports the implementation of privacy and data protection in RFID-based applications.

Problematics of the Thesis

This thesis addresses the privacy issues in RFID systems from the cryptography and information security point of view.

RFID Privacy Models. Various researches have emerged these last years to propose application-level protocols that provide privacy in RFID. However, the search for a generic, efficient, and secure solution that can be implemented in reasonably-costly tags remains open [11, 15, 19, 117]. Solutions are usually designed empirically and analyzed with ad-hoc methods that do not detect all their weaknesses.

Yet, one of the major concerns of cryptography and more generally information security is to establish proofs of security. Such proofs only make sense if they are built in a well-established model. Consequently, many investigations have been conducted to formalize the notion of privacy in RFID. In 2005, Avoine was the first researcher to present a formal privacy model [9] within this context. His goal was to be able to analyze the privacy-friendliness of RFID identification/authentication protocols. Since then, many attempts [31, 34, 44, 47, 77, 82, 99, 106, 107, 111, 136, 139, 162] have been carried out to propose a convenient and appropriate model for RFID.

Based upon these theoretical frameworks, a system designer can formally assess and prove the privacy level of his RFID system within a

cryptographic model. Yet, if the designer is unfamiliar with privacy, he can get confused with so many existing models and may not use the most adapted model to analyze his system. Consequently, providing an analysis and a comparison of RFID privacy models is meaningful to help a system designer in his choice.

New Privacy Threats. In practice, existing privacy models typically consider that (i) an RFID system is composed of tags, readers, and a centralized back-end server, and (ii) readers and back-end are continuously connected online together and form an unique entity, simply called reader. Each one is able to analyze the privacy of an RFID system against a predefined set of adversaries. However, none of these models takes into account the two following privacy threats.

In RFID systems where each tag is associated to a unique secret, the reader generally performs a SearchID procedure to retrieve the corresponding secret to identify/authenticate a given tag. The simplest way to carry out this task is with a linear exhaustive search in the whole reader database. In such a case, the reader might always scan its database in the same way, and the time spent by the reader to identify/authenticate a given tag is always the same for every protocol execution. Consequently, an adversary can deduce which tag is interacting with the reader by only observing/computing this time. This *time information* is clearly an important privacy issue, but it has not yet been included in any existing RFID privacy model.

Another assumption ignored so far in formal analyses is that readers may not always be connected online to the back-end in widespread RFID systems. Readers may be mobile embedded devices that only have an intermittent access to it. In such real-life scenarios, two issues can be raised. Firstly, readers must be able to authenticate the tags of the system during their offline periods: in practice, they must carry the tags secrets to perform their task. Secondly, the ubiquity of the readers, usually located in unprotected areas, increases the risk that an adversary steals one of them. Consequently, if an adversary corrupts a stolen reader and obtain its stored secrets by tampering with it, then the security of the whole system is threatened by such a *compromised* reader. Furthermore, the privacy of the system is completely lost as the adversary is clearly able to trace any tag thanks to the retrieved secrets. Hence, it is critical for

an RFID system to be able to maintain its security level while restoring its privacy upon detection of a compromised reader.

Research Goal

After several years of research on cryptographic models for privacy in RFID systems, it appears that no widely accepted model has been designed yet. Experience shows that security experts usually prefer to use their own ad-hoc model than the existing ones which are perceived as rigid and intricate. In this context, it is thus manifest that unifying and simplifying the models would help the research community to assess RFID protocols meaningfully. The process may take several years to converge towards a universally accepted model, but defining such a model and obtaining a consensus on it is a question of high interest.

The need to define a unified privacy model is also enforced by the international regulatory authorities. For example, the European Commission Recommendation 2009/387/EC [60] stresses that stakeholders should perform a *Privacy Impact Assessment* (PIA) [58] of their RFID solutions before their wide deployment. Similar initiatives related to RFID or less specific ubiquitous devices exist elsewhere, e.g., in Australia [132], Canada [39], and the USA [149] to name a few. A cryptographic privacy model is definitely not enough to design a privacy-friendly solution but it forms the cornerstone of the construction.

This thesis strives to contribute to this effort. From the scientific point of view, the goal is to propose a new privacy model for RFID systems that better fits the real-life expectations. In less restrictive ubiquitous computing environments, the objective is to add a piece to the complex privacy puzzle. This would help the legal and scientific communities to have a better overview of the actual and future challenges related to privacy in broader IT[3] systems. The final aim is to lay the foundation stones in the development of a standardized privacy certification that would assess the privacy level of ubiquitous computing systems.

[3]Information Technology.

Road Map

Chapter 1: RFID Fundamentals

This chapter presents the field of RFID from its origins to its applications. It also introduces the concept of protocols as the application-level mechanisms used between RFID entities to communicate together. The work presented in Chapter 1 has been published in [12].

Chapter 2: Breaking Privacy in Tree-Based Systems

This chapter is an illustration of the privacy issues in RFID systems with a thorough analysis of HSH [78], an authentication protocol based on a key-tree infrastructure combined with lightweight cryptography. HSH is claimed to be light and fast, and to preserve tag privacy under the assumption that tags are tamper-resistant. This chapter firstly proposes a new LPN-based authentication protocol that complies with the threat model considered in [78] and whose reader complexity is lower than the one of HSH, while reaching the same security level as HSH. This protocol further reduces the tag memory requirement, compared to HSH. Secondly, if the assumption of tag tamper-resistance in the HSH threat model is relaxed, as it is commonly admitted in the literature [2, 3], this chapter demonstrates that tampering with one or few tags threatens the privacy of the whole system. The results presented in Chapter 2 have been published in [17] where the attack is the main contribution.

Chapter 3: RFID Privacy Models

This chapter investigates the field of RFID privacy models, and chronologically presents eight well-known models designed to analyze systems based on identification/authentication protocols preserving privacy. Some of them are very popular like [9, 99, 162]. Other ones have interesting frameworks [44, 47, 107] or are valuable successors of [162], such as [34, 82]. The survey presented in Chapter 3 is part of the publication [43].

Chapter 4: Untangling RFID Privacy Models

To go one step further in the study, this chapter provides an analysis and a comparison of the models presented in Chapter 3 by highlighting their strengths and weaknesses. It first analyzes five different authentication protocols with each of these models. This study exhibits the lack of granularity of these models, meaning that no model can fairly analyze and compare protocols that are designed with different security levels. Then, the eight models are thoroughly compared regarding their different features and privacy notions. The results show that no model encompasses all the others. They however point out the most appropriate model(s) to use for analyzing a protocol in specific scenarios. The results presented in Chapter 4 are part of the publication [43].

Chapter 5: Time Attacks Threatens Privacy-Friendly Systems

This chapter introduces the *time attacks* through the formalization of a new privacy level called TIMEFUL. It then displays the weaknesses of several existing protocols when facing this TIMEFUL adversary. The privacy analyses show that none of these protocols resists to such an adversary. Finally, various solutions to ensure TIMEFUL-privacy are proposed. They consist in combining an appropriate choice for the reader database structure with a pertinent SearchID procedure: the approaches are based on rainbow tables, hash tables, B-trees, and random search. The results presented in Chapter 5 have been published in [13].

Chapter 6: When Compromised Readers Meet RFID

This chapter formally models the "compromised reader" attack. Then, it demonstrates that a multi-reader-based RFID authentication protocol does not ensure privacy in such a context. Finally, two solutions based on privacy-restoring mechanisms to face the problem of compromised readers are proposed: one for semi-offline systems, and another one for offline systems. The offline solution is shown to be deployable in practice by analyzing the efficiency of its privacy-restoring mechanism during a 3-day automobile race that took place in 2010. Up to our knowledge, *restoring* privacy in RFID systems is a new concept introduced here.

The results presented in Chapter 6 have been published in [14, 16].

Chapter 7: Untraceability Model for RFID

This chapter first presents additional arguments that emphasize the necessity to define a new model capable of comparing protocols meaningfully. It consequently issues an untraceability model that is operational where the previous models were not. The model aims to be easily understandable and manageable. This spirit led to a modular model where adversary actions (*oracles*), capabilities (*selectors* and *restrictions*), and goals (*experiment*) emerge in a way that is natural and intuitive. This design enhances the ability to (i) formalize new adversarial assumptions (such as time attacks or compromised readers introduced in Chapters 5 and 6) and future evolutions of the technology, and (ii) provide a finest privacy evaluation of the protocols.

Chapter 8: Toward Privacy Certification

Finally, this chapter extends the privacy question to the less restrictive area of ubiquitous computing systems. As no current standardized certification is available today to practically assess the privacy level such environments, this chapter aims to fill that gap. Firstly, it introduces the context in which we foresee the development of a privacy certification. This chapter then displays the privacy landscape in Europe from the legal, societal and non-academic information security point of views. In particular, it presents (i) the most important European legislations related to the protection of citizens' privacy and (ii) the existing approaches that have been developed to provide and evaluate privacy in IT environments. Finally, this chapter lays the foundation stones on a general methodology that will help IT stakeholders in designing together an international privacy certification able to assess the privacy level of ubiquitous computing systems.

Chapter 1

RFID Fundamentals

Often referred as the new trend, RFID is a wireless technology that allows to remotely identify and/or authenticate transponders called *tags* without line of sight. The device querying tags is called a *reader*, even if it would be more appropriate to call it an interrogator. It is difficult to precisely define what is RFID because everyone has his own vision. Yet, two fundamental characteristics always appear: each tag has a unique identifier and tags respond to requests performed by readers but cannot communicate with other tags. In 2009, the European Commission published in its Recommendation 2009/387/EC [60] the following definition.

> "*RFID means the use of electromagnetic radiating waves or reactive field coupling in the radio frequency portion of the spectrum to communicate to or from a tag through a variety of modulation and encoding schemes to uniquely read the identity of a radio frequency tag or other data stored on it.*" (Article 3.a).

Such definition does not remove all the ambiguities nor the boundary between RFID and contactless smartcards. Smartcards manufacturers usually prefer to make a distinction between these two concepts because the term of RFID reflects the idea of weak security. In turn, RFID manufacturers consider that contactless smartcards are a form of RFID.

In this chapter, we present the field of RFID from its origins to its applications. Then, we introduce the concept of protocols as the mechanisms used between RFID entities to communicate together.

1.1 Origins of RFID

Although RFID has prominently grown in recent years, its history rooted in the middle of the 20th century. The invention of the technology is commonly associated with the design of the IFF[1] system by the RAF[2] to identify Allies aircrafts. It is impossible to link the creation of RFID to one person, but it is undeniable that Charles A. Walton greatly contributed to it with the publication of numerous patents, including the one recorded in 1973 on a passive transponder [164].

Today, RFID with real computing power is derived from a cross of knowledge in the areas of microchip and identification by radio frequency. The 80s have been a turning point in the history of contactless technology with the first commercial applications, including livestock identification and highway tolls. However, RFID only took off in the 90s, especially with the massive sale of the Mifare Classic [128] developed by Mikron (acquired by Philips Semiconductors, now NXP Semiconductors), where several hundred million of copies have been sold since its introduction in the public market. The general public only realized the magnitude of the phenomenon with the deployment of applications that became essential, such as public transport ticketing, pet identification or e-passports.

1.2 RFID Technology

1.2.1 Architecture

As depicted in Figure 1.1, an RFID system is generally composed of three types of device: tags, readers, and a centralized back-end server.

A tag is a transponder, i.e., an integrated circuit coupled with an antenna, embedded into a remote item. It may have different power sources (either its own or the one given by the reader). Its memory can vary from a hundred of bits (as for EPC tags [56]) to a few Kbytes (such as contactless smartcards [86, 126]). It may have different levels of computational capabilities: some tags can only perform logic operations, while others can compute symmetric-key (SK) cryptography, hash functions, or even public-key (PK) cryptography. A tag may not always

[1] Identify Friend or Foe.
[2] Royal Air Force.

be tamper-resistant: it can be corrupted by an adversary. Its distance range of communication is between few centimeters to several meters. The characteristics of RFID tags are thoroughly detailed in Section 1.2.2.

A reader is a transreceiver. It can communicate with a tag when the latter is in its electromagnetic field. It may also communicate with other readers or the back-end through other channels (e.g., ethernet, WiFi). A reader is generally more powerful than a tag: its computation capacities can be compared to a small computer. It can be either fixed (e.g., at the entrance of a building) or mobile (e.g., a PDA), and is generally assumed to be tamper-resistant.

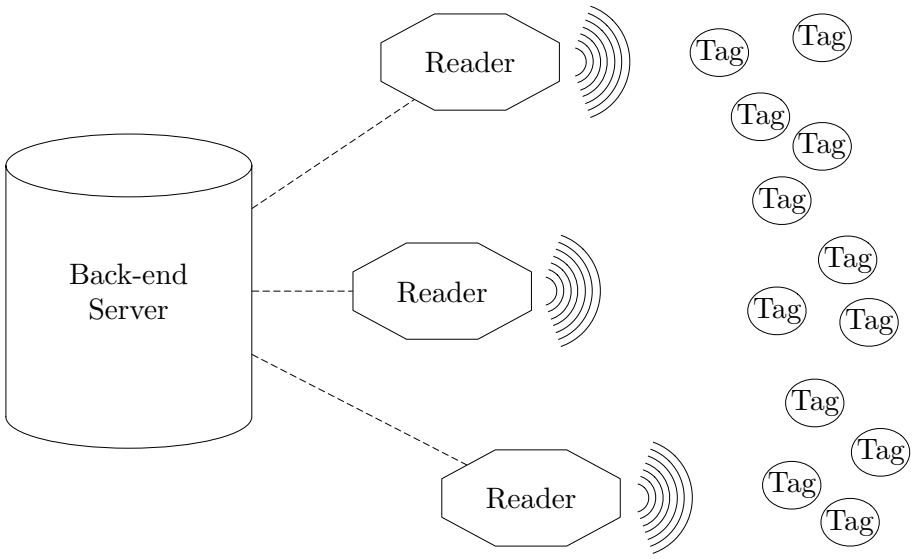


Figure 1.1: A typical RFID architecture.

The back-end server potentially contains a database DB that contains data related to each tag and reader of the system (e.g., tag identifiers). However, the back-end server can also be a kind of switch which only forwards the communications between readers. In any case, it can only communicate with readers. Note that the back-end may not always be needed: for instance, if a system is composed of a unique and autonomous reader, then this device may play as well the role of back-end. In other

systems, the back-end and the readers are connected online all together through a secure channel, and thus are considered as a single entity, simply called reader.

These entities interact together via some communication protocols (where messages are exchanged) in order to reach a given objective (e.g., identify/authenticate the tags).

1.2.2 Tags Characteristics

Establishing a comprehensive classification of the RFID technologies is difficult since many characteristics must be considered to define a tag. It is only possible to identify the correct technological needs – thus the most appropriate tag – of an RFID application during its design. Yet, the main technical characteristics of RFID tags are shown in Figure 1.2 and described below.

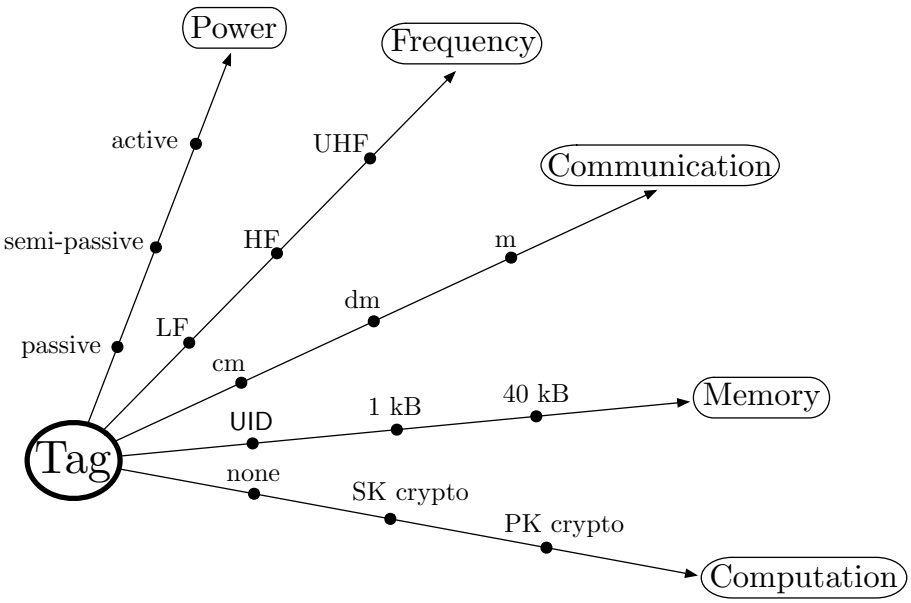


Figure 1.2: Main characteristics of tags.

Power Source. There are two main classes of tags according to the power source used. Tags powered by their own battery for internal com-

putations and communication with readers are called *active*. Tags that get their power from the reader electromagnetic field are called *passive*. Note that the terminology active/passive is not related to the computation or communication capacities of tags, but only how they are powered. Finally, tags are said to be *semi-passive* when they use their own battery for computations, but the energy supplied by the reader for communication. This type of tag is much less present in the market.

The majority of tags used today are passive, and the unqualified term RFID is generally used to designate them. For instance, tags for animal tattoo, to replace bar codes, passports or public transportation tickets are passive, while tags for opening car doors or highway tolls are active. In this thesis, the term "RFID tag(s)" will refer to passive tag(s).

Frequency. The RFID technology mainly operates at five frequency bands as defined in the standard ISO/IEC 18000 [95]. These frequencies are listed below, along with their most representative application areas.

- 124 – 135 kHz (LF): animal tattoo.
- 13.56 MHz (HF): payment, access control, ticketing.
- 433 MHz: car park access control.
- 860 – 960 MHz (UHF): supply chain.
- 2.45 GHz: highway tolls, containers identification.

Three of these five frequency bands are the most usually found in deployed RFID applications. The first one is the 124 – 135 kHz (LF) frequency because it allows a good penetration in environments consisting of metals and liquids. The second one is 13.56 MHz (HF) frequency as it provides a tag with enough power to perform cryptographic operations. The last one is the 860 – 960 MHz (UHF) frequency since it provides better communication distance in the case of passive tags.

Of course, the frequency choice is more complex than what is presented here and depends on many other arguments, such as the size of the antenna, the regulations on the frequency bands according to the country where the RFID system is deployed, or the ease and therefore the cost of production.

Communication Range. The communicating distance between a tag and a reader depends on many parameters, including the frequency, but also the transmission power, the environment, the antennas, etc. For passive tags, it is possible to approximately observe the following communication ranges depending on the frequency band:

- Low Frequency (LF): few centimeters,
- High Frequency (HF): few centimeters to few decimeters,
- Ultra-High Frequency (UHF): few meters.

Note that these ranges are given according to the standards and specifications of the manufacturers. However, several studies have shown that these ranges can be substantially increased if specific reading material is used [80, 160].

Memory. As for the other parameters, the amount of available memory on the tag depends on the requirements related to the application. It is necessary to have at least a few dozen bits to store the Unique IDentifier (UID) of the tag. This UID is generally determined by the manufacturer and cannot be changed later. Besides this UID, the tag commonly has additional EEPROM[3], typically one or two kilobytes. This memory may exceptionally be much larger, for example between 30 and 70 kilobytes for an electronic passport.

Computation Capabilities. Tags are clearly very limited in terms of computations. Some of them can only perform logical operations (e.g., to compare a received password with another stored). However, the 90s has been the witness of the emergence of passive tags with cryptographic capabilities, typically a stream encryption algorithm. Today, it is even common to use passive tags with a block cipher algorithm such as 3DES or AES. Expensive (but still available on the market) passive tags can also achieve public-key cryptography, such as electronic passports [21]. There is therefore a wide range of tags with diverse computation capabilities that integrators seek to minimize for a given application.

[3]Electrically Erasable Programmable Read-Only Memory.

1.2.3 Standards

Many standards related to RFID already exist, either at the physical, communication, or application layers, as shown in Figure 1.3. This section presents the principal standards related to contactless technologies.

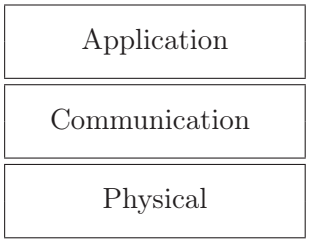


Figure 1.3: Simplified RFID layers model.

ISO/IEC 14443 [93] and ISO/IEC 15693 [92]. These standards cover the physical and communication layers of the 13.56 MHz frequency and form the cornerstone of most applications that do not rely on proprietary standards. The first one is dedicated to the *proximity* tags (communication range around 10 cm), while the second one targets the *vicinity* tags (communication range around 80 cm). For instance, the electronic passports standardized by the ICAO[4] [88] are based on the ISO/IEC 14443 standard.

EPC Class 1 Gen 2 [56]. The deployment of low-cost tags has been boosted by the *Auto-ID Center*, a consortium created in the United States in 1999. This organization, now composed of the *EPC Global Network* and *Auto-ID Labs*, aims to standardize and promote RFID in supply chain managements. The EPC Class 1 Gen 2 standard published by the EPC Global Network is now widely deployed and followed by many manufacturers. It covers both the physical and the communication layers, as well as the application one.

[4]International Civil Aviation Organization.

NFC. The NFC[5] is a wireless communication technology that operates within the 13.56 MHz frequency band. It comes from a consortium[6] created in 2004 by Sony, Philips and Nokia, which is now composed of more than 150 members. This technology is compatible with RFID – especially with the ISO/IEC 14443 standard – and allows tag reading or simulation. Consequently, two NFC devices, typically mobile phones, can communicate using the RFID technology. This allows low bandwidth communications of short distance that are established much faster than Bluetooth or WiFi communications. The ISO/IEC 18092 [94] standard defines the main features of NFC, and also offers a specific format (NDEF[7]) to store and exchange information in order to enhance the interoperability between NFC devices.

Although there is a tendency to distinguish NFC from RFID, we must admit that NFC simply is an extension of some RFID standards. The security issues of NFC are thus similar to those of RFID. However, the solutions to solve these problems may be different, e.g., taking advantage of the capabilities of the phone. Note that it is possible to develop an application compliant with NFC, without necessarily using mobile phones, by staying in a reader/tag model as described previously.

1.3 Application Examples

There exists many RFID-based applications that are present in industry. These applications can still be grouped into three categories: (i) access control, (ii) tracking and production control, and (iii) secured applications. This section presents these three areas and illustrates their user-friendliness with examples from everyday life.

1.3.1 Access Control

The smartcard technology has been used in access control systems since many decades. Yet, the arrival of the contactless technology on the market enabled a user-friendliness that was unknown so far. In fact, access control has been firstly processed with infrared passes, then magnetic

[5] Near Field Communication.
[6] `http://www.nfc-forum.org/`
[7] NFC Data Exchange Format.

stripe cards, and finally contact smartcards. The main drawback of the latter is that the user must insert his card into a reader which is a significant loss of time, especially for mass access controls. In addition, maintenance of readers is a handicap since readers are directly accessible and often vandalized. The RFID technology can reduce these problems and facilitate the access control by integrating the tag in a card, a key ring, or bracelet. That way, the user simply passes the item in front of the RFID reader that manages the access control. Below are presented a few applications for access control.

Ski Pass. The access control in ski resorts has greatly benefited from the RFID technology. It generally follows the ISO/IEC 15693 standard that allows a slightly larger communication range than the ISO/IEC 14443 standard (around 50 centimeters). RFID facilitates the skier's life: he has no need to seek his access card, he just passes his pass (potentially in his pocket) in front of the reader. The introduction of RFID is also an asset for the sky lifts company as it speeds up the flow of skiers while maintaining a systematic control.

Automobile. Since the early 90s, the car industry has also adopted RFID-based access control. Such systems allow to either get into the car or to start it, and generally use LF frequencies. The purpose of RFID in these applications is to strengthen the security of the access control, especially for the starting up.

Highway Toll. The use of RFID in highway tolls is relatively old. As for ski lifts, this system eases the life of drivers and of the company. The driver does not have to completely stop his vehicle at the toll-booth. Some systems may even allow the driver to go through the gates at full speed. The company gains in efficiency and reduces its costs. RFID automates control which allows the company to reduce the labor needed at the toll area. As the required reading distance is relatively long (around 5 meters), UHF active tags can be an adequate technological choice.

Public Transportation. RFID is an advantage for both the traveler and the transport company. The traveler gains in simplicity: it is easier for him to pass its ticket in front of the reader than to insert it into a

reader. Furthermore, if the transport company bases its fares on trip distance, the traveler does not need to take care of this information: he only has to validate his pass at the beginning and at the end of his journey. Transport companies also acquire more accurate statistics on the visit of their infrastructure than a simple traveler count. The RFID technology also enables the company to reduce the counterfeiting of tickets, as it is more difficult to produce fake RFID tags than paper tickets. Nowadays, many cities opted for RFID-based public transportation such as Paris, Brussels, London, Berlin, Seoul, Hong Kong, Moscow, Washington or New York to name a few. For instance, in 2008, the Brussels public transportation company (STIB) launched an RFID-based ticketing system, called MOBIB, that will eventually replace the former system based on magnetic strip cards. MOBIB relies on the Calypso standard [87] that is already used by more than 20 countries.

In 2009, we studied in depth this Belgian system and developed the software MOBIB Extractor[8] which allows anyone to read a MOBIB card. We especially pointed out several flaws demonstrating that the card does not ensure privacy to its holder. Indeed, the MOBIB card stores the name, date of birth, and zip code of its holder, and further records the date, hour, metro/tramway/bus line and stop of its three last stampings. All these data are written in clear on the card, and their reading is not locked with any authentication. The card holder is thus subject to an obvious threat of privacy that could have been reduced with a simple data encryption or an authentication mechanism to access the data.

1.3.2 Tracking and Production Control

RFID is the new alternative to barcodes for logistics that furnishes two main advantages. The first one is the reading distance: barcodes can only be read at a very short distance, whereas the RFID technology offers the possibility to read a tag at several meters without a necessary line of sight. Such a reading distance increases the company efficiency, since it is possible to scan in one pass all the pallets inside a container or to check the stocks status in real time. The second advantage of RFID is a greater resistance to external elements. Indeed, if a barcode is torn, bent or dusty, then reading is impossible. Reading conditions also play

[8] `http://sites.uclouvain.be/security/ourlabs.html/`

an important role, e.g., a barcode may not be read if there is too much light. RFID does not suffer from these problems, although other issues related to the radio frequency communication feature are possible.

Pet Identification. The emergence of RFID for pet identification goes back to the early 80s. RFID is used for inventory and production controls, but also to automate cattle feed. Tags are attached to the ears of the animals, or embedded in necklaces or rings, depending on the type and size of the animal. RFID can also track the animal origin to perform quality and health control (e.g., to contain epidemics, the most famous being the mad cow disease, or more recently the swine flu). This use requires a full interoperability between the different production companies. To do so, standards have been set up, especially the ISO/IEC 11784 [90] and ISO/IEC 11785 [91] ones, based on the LF frequency. These tags can be found in many forms. Some of them are about the size of a rice grain that are subcutaneously injected using a syringe. Other tags are of the size of a candy bar that are ingested by animals.

Library. RFID facilitates book loan and inventory management in libraries. It automates the procedures of book loan and return, and detects if a book has been placed in the wrong location on the shelves. In addition, when gates are installed at the library exits, then books that are not recorded on loan in the system can sound an alarm. Libraries typically use the HF technology based on the ISO/IEC 15693 standard.

Supply Chain Management. The American supermarket giant Walmart was one of the first retailers to use RFID to manage its supply chain. Its project began in 2003, when it imposed its main suppliers to equip their products with RFID tags by the end of 2005. Since the cost was too high for the suppliers, Walmart revised its expectations downward and only required RFID tagged pallets. Tags used by the distributor are EPC Class 1 Gen 2 that allow pallets to be read at a distance of about two meters.

1.3.3 Secured Applications

Electronic Passport. The ICAO is the organization that launches the introduction of electronic passports. The tags embedded in the passports meet the ICAO Doc 9303 standard [88] for the application layer and the ISO/IEC 14443 standard [93] for the lower layers. The information about the passport holder is stored in data groups (DGs). In particular, DG1 contains all the data (e.g., the date of birth, gender, name of the passport holder) written on the passport area used for automated reading (called MRZ[9]). DG2 contains the photo of the holder, and DG3 (mainly used in Europe) contains the fingerprints of the holder.

The integrity and validity of the data on the tag is secured by various mechanisms described in [21]. Data reading is also protected by an authentication protocol (called BAC[10]). The details of BAC are out of the scope of this thesis. Note however that the complete execution of this protocol requires the knowledge of the second line of the passport MRZ. This implies to mandatorily have the passport open in your hands to access the tag contents. Passports that store fingerprints also use a strengthened authentication mechanism (called EAC[11]) which only provides access to the fingerprints to the authorities in charge of checks.

Contactless Payment. One recent application using RFID is the contactless payment. It is already studied by several manufacturers and, for now, its use is limited to small amounts of money. One of the best known examples is the Speedpass, introduced in 1997, enabling American car drivers to pay their fuel by contactless payment in the Exxon, Mobil and Esso gas stations. In such systems, tags are integrated into key chains, and operate at the LF frequency.

Major banking groups MasterCard, American Express and Visa also investigate such applications. Since the early 2000s, each one develops its own contactless payment system which, unlike Speedpass, operates at the HF frequency with the ISO/IEC 14443 standard.

Contactless payment is also a booming field for NFC, especially in the case of public transportation. Many countries (e.g., France, Japan, United States, China) test NFC solutions to centralize all the various

[9] Machine-Readable Zone.
[10] Basic Access Control.
[11] Extended Access Control.

card systems of users on their mobile phone. According to the related firms, it will significantly increase the benefits for the user: the phone will act as a credit card, public transportation pass, health insurance card, and potentially identity card. The user will (almost) not need a wallet: his phone will be enough.

1.4 Primer on Protocols

At the application layer of the RFID model given in Figure 1.3, the system entities communicate together through a protocol.

Identification vs. Authentication. Nowadays, RFID is generally used for systems whose purposes are either the identification of items (e.g., libraries), or their authentication (e.g., access control). Clearly, these two concepts do not target the same objectives: *identification* is the act of acquiring the identity of an item, whereas *authentication* is the act of confirming that the item is authentic. Yet, most of the scientific literature considers that there is no difference between the two notions. For example, [116] only provides one definition for both concepts. To withdraw this ambiguity, the two notions are explicitly defined.

Definition 1.1 (Identification Protocol). *An* identification protocol *is a process whereby a reader obtains the identity of a tag communicating with it, but does not ask any proof of it.*

Definition 1.2 (Authentication Protocol). *An* authentication protocol *is a process whereby a reader is convinced about the identity of a tag communicating with it.*

In a simple identification protocol, the reader sends a request ("who are you?") to a tag, and the tag answers its identifier. At the end, the reader knows which tag is communicating with it. In a typical authentication protocol, the classical used method is called "challenge/response", and the tag must demonstrate to the reader the knowledge of a secret that is also known by the reader. The reader first sends a random challenge number to the tag, and the tag answers its identifier and $f(challenge, secret)$, where f is a designated function (e.g., an encrypting function). The challenge is a method that avoid replay attacks. Indeed

without a challenge, an adversary could simply eavesdrop the fixed response of a tag, and replay it to get authenticated by the reader. With the challenge, this attack is no longer possible unless the adversary gets the same challenge: the probability that such an event occurs is generally considered as negligible when the size of the challenge is long enough. At the end, the reader is assured of the tag identity since f has been applied on the secret and on the challenge it chose.

This thesis focuses on protocols that achieve both properties where a tag gets authenticated by a reader, and the latter should be the only entity able to obtain the identity of this tag.

Notations. In the sequel, we consider that an RFID system $\mathcal{S}$ with security parameter λ is composed of a set of n tags and a single legitimate reader $\mathcal{R}$ containing the system database DB. The entities of the system $\mathcal{S}$ communicate together through a protocol. To simplify the notations, the figures of the protocols presented in this thesis represent the reader database DB for one tag $\mathcal{T}$, and:

- the tag $\mathcal{T}$ has a unique identifier $\mathsf{ID}_{\mathcal{T}}$;
- all the secrets of the system are considered to be independent;
- the protocol is aborted (i) when a check is not correct or (ii) when the reader $\mathcal{R}$ does not find a compatible match in its database DB;
- the reader $\mathcal{R}$ correctly authenticates the tag $\mathcal{T}$ when the protocol correctly ends.

F and G refer to pseudo random functions, while f and g refer to one-way functions, and h refers to a cryptographic hash function. (Enc/Dec) refers to a cryptosystem. (Sign/Verif) refers to a signature scheme. A comma "," within the scope of these cryptographic operations denotes the concatenation "||".

Let $\{0,1\}^k$ denote the set of all binary strings of size k, then $e \in_R \{0,1\}^k$ means that the element e is randomly chosen in the set $\{0,1\}^k$ using a uniform distribution.

Let $poly(.)$ denote a polynomial function. The function $\epsilon(\lambda) : \mathbb{N} \rightarrow \mathbb{R}$ denotes a negligible function in λ if, for every positive polynomial function $poly(.)$, there exists an integer N such that, for all $\lambda > N$, $|\epsilon(\lambda)| < \frac{1}{poly(\lambda)}$.

Reader Complexity. In the presented protocols, the goal of the reader $\mathcal{R}$ is to successfully authenticate a tag $\mathcal{T}$. To achieve this task, $\mathcal{R}$ must search in its database the data used by $\mathcal{T}$ during a given protocol execution (i.e., its identifier and/or secret(s)). The number of cryptographic operations required by the reader to achieve this task is called "complexity of the reader database search", or simply "reader complexity". In the next chapters, we will see that the reader complexity is not identical for all the protocols.

Chapter 2

Breaking Privacy in Tree-Based Systems

Existing RFID protocols based on symmetric-key cryptography suffer from a large computational complexity on the reader side when each tag is attached to a unique secret key. Generally, a reader hase to perform $O(n)$ operations to authenticate and/or identify a tag, where n is the total number of tags in the system. In 2004, Molnar and Wagner proposed in [117] a tree-based key infrastructure (called MW) to decrease the reader complexity of such protocols to $O(\log(n))$. In 2009, Halevi, Saxena, and Halevi presented in [78] a lightweight RFID privacy-friendly authentication protocol (called HSH in what follows) that also aims to reduce the reader complexity. The HSH protocol, presented in details in the next section, is based on the HB^+ protocol combined with the MW key infrastructure. It is claimed to be light and fast, and to preserve tag privacy under the assumption that tags are tamper-resistant.

In this chapter, we first propose a new LPN-based authentication protocol that complies with the threat model considered in [78] and whose reader complexity is lower than the one of HSH, while reaching the same security level as HSH. Our protocol further reduces the tag memory requirement, compared to HSH. Secondly, we consider that the assumption of tag tamper-resistance in the HSH threat model is too strong, and we demonstrate that relaxing this assumption, as it is commonly admitted in the literature [2, 3], threatens the privacy of the whole system.

2.1 HSH Building Blocks

The heart of HSH is that all its design relies on the HB-family protocols combined with the MW tree-based key infrastructure. Before explaining in details the HSH protocol, these two building blocks are presented.

2.1.1 HB$^+$ Protocol

HB Family. In 2000, Hopper and Blum [84, 85] took benefit of the *Learning Parity from Noise* (LPN) problem to design a human-to-computer authentication protocol, today known as HB.

The LPN problem is one of the most well-known hard learning problems in cryptography that can be summarize as follows. Given that k is a secret k-bit vector, a is a random known k-bit vector, $\epsilon \in]0, \frac{1}{2}[$ is a noise parameter, and η is a bit noise where $\Pr(\{\eta = 1\}) = \epsilon$, then it is hard to recover k from the result $r = a \cdot \mathsf{k} \oplus \eta$.

Many attempts on identification and authentication protocols relying on the LPN problem have been proposed so far, such as all the HB-family protocols [7, 25, 26, 27, 71, 78, 79, 84, 85, 100, 108, 112, 119, 145, 146, 165], or the LPN-C protocol of Gilbert, Robshaw and Seurin [72]. This section presents HB$^+$, one of the most famous protocols of the HB-family.

Description of the Protocol. In 2005, Juels and Weis [100] noticed the link between the human-to-computer and tag-to-reader authentication paradigms: the computation capabilities of the provers are quite restricted in both cases. The authors also showed that the HB protocol is not resistant against an active adversary. Indeed, such an adversary can query as much as she wants a target tag, and obtain a set of equations that allows her to recover the tag secret using the Gaussian elimination technique. Juels and Weis therefore proposed in [100] the HB$^+$ protocol to improve the original HB protocol against such attacks.

At the system setup, each tag $\mathcal{T}$ has a unique pair of secret keys $(\mathsf{k}_{\mathcal{T},1}, \mathsf{k}_{\mathcal{T},2})$ known by every reader $\mathcal{R}$, where $|\mathsf{k}_{\mathcal{T},1}| = |\mathsf{k}_{\mathcal{T},2}| = k$. $\mathcal{T}$ is also given a random noise parameter $\epsilon \in]0, \frac{1}{2}[$.

Then, λ rounds of challenge/response are required by the reader to authenticate the tag $\mathcal{T}$, where λ is a security parameter. In Figure 2.1, these rounds are represented by the size λ of the matrices. $\mathcal{R}$ selects a $\lambda \times k$ random binary matrix $M_{\mathcal{R}}$ and sends it to $\mathcal{T}$. Then, $\mathcal{T}$ chooses a

$\lambda \times k$ random binary matrix $M_{\mathcal{T}}$ and a λ-bit noise vector ν, and answers $E = (M_{\mathcal{R}} \cdot \mathsf{k}_{\mathcal{T},1}) \oplus (M_{\mathcal{T}} \cdot \mathsf{k}_{\mathcal{T},2}) \oplus \nu$. $\mathcal{R}$ recovers $\mathcal{T}$'s identity if the Hamming weight $w_{\mathcal{H}}$ of the supposed noise vector ν is lower than $\lambda\epsilon$.

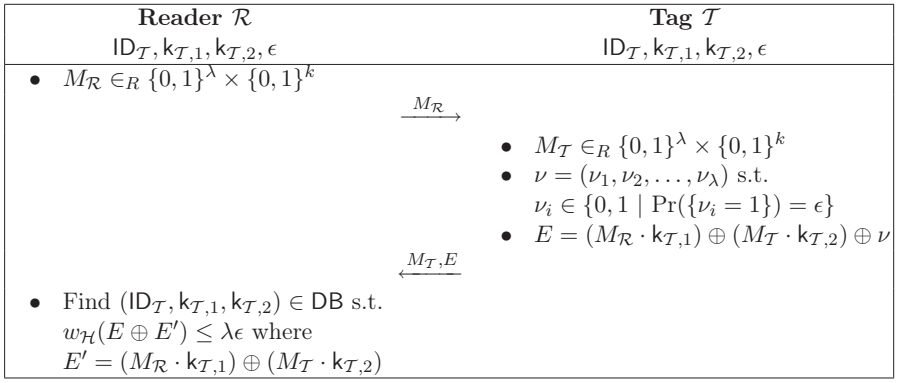


Figure 2.1: HB$^+$ authentication protocol.

Reader Complexity The reader performs a linear exhaustive search on its database to authenticate a tag. Its complexity for the HB$^+$ protocol is thus in $O(n)$.

2.1.2 MW Key Infrastructure

The tree-based key infrastructure has been proposed by Molnar and Wagner in [117] at ACM CCS 2004. Such a key infrastructure (called MW in the sequel) has been created in order to decrease the reader complexity of every protocol based on it to $O(\log(n))$, and is built as follows.

In a system of n tags, a key-tree is generated with $\beta^d \geq n$ leaves, where d is its depth and β is its branching factor. Each leaf is randomly associated to a tag $\mathcal{T}$ of the system. Each node is associated to a unique secret key $\mathsf{k}_{i,j}$, where i and j are respectively the depth and the branch of the node. W.l.o.g., $(p_0, p_1, p_2, \ldots, p_d)$ denotes the path in the tree, from the root node (denoted p_0) to the leaf (denoted p_d), that is associated to the tag $\mathcal{T}$.

2.2 HSH Protocol

This protocol has been proposed in [78] and is divided in two stages: a tree traversal and an authentication. The following table gives the notations and recommended values for HSH.

Notation	Meaning	Recommended Values [78]
d	depth of the tree	$d \in \{2, 3\}$
β	tree branching factor	$\beta \in \{100, 1000\}$
k_1, k_2	key lengths	$k_1 = 80$, $k_2 \in [224, 512]$
λ	system parameter	$\lambda \in [80, 212]$
ϵ	noise parameter	$\epsilon \in [\frac{1}{8}, \frac{1}{4}]$

The choice of the HB-family comes from the fact that such protocols perfectly fit in low-cost RFID tags. Here, we present the HSH version using HB$^+$ (this choice is given by the HSH authors). Then, HSH is built with the MW key infrastructure because it significantly reduces the reader complexity during the identification process: $\beta\lceil\log_\beta n\rceil$ operations in the worst case, instead of n.

2.2.1 Description of HSH

Initialization. It follows the same procedure as the MW key-tree presented in Section 2.1.2. Given a system with one reader $\mathcal{R}$ and n tags, the parameters β and d are chosen at the system setup, such that they define a MW key-tree with $\beta^d \geq n$ leaves. Each leaf is randomly associated to a tag of the system. During the setup of the system, each tag $\mathcal{T}$ is initialized with:

- the k_2-bit keys of its path nodes $\{\mathsf{k}_{p_1}, \mathsf{k}_{p_2}, \ldots, \mathsf{k}_{p_d}\}$,
- a unique pair of secret keys $(\mathsf{k}_{\mathcal{T},1}, \mathsf{k}_{\mathcal{T},2})$, where $|\mathsf{k}_{\mathcal{T},1}| = k_1$ and $|\mathsf{k}_{\mathcal{T},2}| = k_2$.

At the end of the setup, $\mathcal{R}$ stores the entire key-tree and all the pairs $(\mathsf{k}_{\mathcal{T},1}, \mathsf{k}_{\mathcal{T},2})$ of each tag $\mathcal{T}$.

Tree Traversal. First of all, $\mathcal{R}$ must recover the right pair $(\mathsf{k}_{\mathcal{T},1}, \mathsf{k}_{\mathcal{T},2})$ to correctly authenticate the tag $\mathcal{T}$. To do so, $\mathcal{T}$ chooses a $\lambda \times k_2$ random binary matrix $M_{\mathcal{T}}$ and a λ-bit random noise vector ν^i for every level i

in the tree. Then, $\mathcal{T}$ computes $E^i = M_{\mathcal{T}} \cdot \mathsf{k}_{p_i} \oplus \nu^i$, and sends $M_{\mathcal{T}}$ and E^i for every level i of the tree to $\mathcal{R}$.

Upon reception of these data, $\mathcal{R}$ visits the key-tree, node by node, using the E^i's. Firstly, $\mathcal{R}$ computes for every child node of the root: $E^{\text{child}} = M_{\mathcal{T}} \cdot \mathsf{k}_{\text{child}}$, where $\mathsf{k}_{\text{child}}$ is a child key. $\mathcal{R}$ visits the child node whose answer is the closest to the data E^1 sent by $\mathcal{T}$. The same procedure is iterated for every level i in the tree.

At the end, $\mathcal{R}$ reaches one leaf of the tree, and identifies the tag $\mathcal{T}$ with its corresponding pair of secret keys $(\mathsf{k}_{\mathcal{T},1}, \mathsf{k}_{\mathcal{T},2})$.

Authentication. After the tree traversal, $\mathcal{R}$ performs the HB$^+$ protocol on this pair $(\mathsf{k}_{\mathcal{T},1}, \mathsf{k}_{\mathcal{T},2})$ to confirm its result and authenticate $\mathcal{T}$ as follows. $\mathcal{R}$ sends a $\lambda \times k_1$ random challenge binary matrix $M_{\mathcal{R}}$ to the tag $\mathcal{T}$. Then, $\mathcal{T}$ chooses a λ-bit random noise vector ν, and sends back the result $E = M_{\mathcal{R}} \cdot \mathsf{k}_{\mathcal{T},1} \oplus M_{\mathcal{T}} \cdot \mathsf{k}_{\mathcal{T},2} \oplus \nu$. $\mathcal{R}$ computes $E' = M_{\mathcal{R}} \cdot \mathsf{k}_{\mathcal{T},1} \oplus M_{\mathcal{T}} \cdot \mathsf{k}_{\mathcal{T},2}$ with the pair found at the end of the tree traversal stage, and computes the Hamming distance between E and E'. If this value is under the threshold $\lambda\epsilon$, then $\mathcal{R}$ accepts $\mathcal{T}$, otherwise it rejects it.

2.2.2 Threat Model

Attacks on the HB Family. The main weakness of HB-based protocols is related to the simple nature of the messages exchanged during protocol executions. Indeed, each tag answers a linear combination of its secrets with (known) random values and (unknown) noise. Such a mathematical property favors key-recovery attacks against this family of protocols [38, 66, 69, 70, 75, 101, 102, 103, 137, 153]. Most of them rely on man-in-the-middle attacks, where the adversary is active and can intercept, relay, modify, delete the messages exchanged during a protocol execution, or inject new ones.

Attacks on the MW Key Infrastructure. Several traceability attacks have already demonstrated that tree-based protocols (i.e., with the MW key infrastructure) do not provide tag privacy. The first attack (called ADO in this thesis) has been presented by Avoine, Dysli, and Oechslin in [15], where the authors showed that corrupting one or several tags (by tampering with the tag(s) and recovering its(their) secrets)

allows an adversary to trace all the tags of the system with a certain level of accuracy. Then, Buttyán, Holczer and Vajda presented in [32] an attack based on the notion of anonymity set [40]. Finally, Nohl and Evans proposed in [125] another traceability attack based on ADO, with a slightly different adversary model (i.e., the adversary can only play with a subset of all the tags belonging to the system).

HSH Threat Model. As HSH relies on the MW key infrastructure and on the HB$^+$ protocol, it naturally inherits from MW's and HB$^+$'s weaknesses. Since most protocols of the HB family are vulnerable to active adversaries, [78] considers adversaries who can eavesdrop all the communications and interact with both $\mathcal{R}$ and $\mathcal{T}$, but who cannot perform man-in-the-middle attacks. Additionally, the MW key infrastructure is known to enable traceability attacks when tags can be corrupted and their secrets are revealed to the adversary. Thus, [78] considers that tags are tamper-resistant.

2.3 Our Protocol

2.3.1 Problem Statement

Reader Complexity. As explained in the previous section, the reader complexity of HSH is in $O(\log n)$ which is better than the one in $O(n)$ of a classical symmetric-key-based protocol.

Considering the same threat model as HSH (where tags are said to be tamper-resistant), we decide to put a unique pair of symmetric keys $(\mathsf{k}_1, \mathsf{k}_2)$ shared between $\mathcal{R}$ and all the tags of the system, in order to decrease $\mathcal{R}$'s complexity. Thus, having a common pair of keys for the whole system is better for $\mathcal{R}$'s computation search, rather than n pairs (i.e., one unique per tag in classical symmetric-key-based cryptography). The reader complexity of our protocol will thus be in $O(1)$.

Tag Identification. In a classical HB-based protocol, each tag $\mathcal{T}$ has a unique (pair of) symmetric secret key(s) to authenticate itself to the reader $\mathcal{R}$. During the protocol execution, $\mathcal{T}$ adds some noise to its answer according to the parameter ϵ. Then, $\mathcal{R}$ scans its database (which contains the secret keys all the tags) and stops when it finds a match enough close

to $\mathcal{T}$'s answer (with respect to the noise probability ϵ). Clearly, $\mathcal{R}$ does not try all the secret keys to find the one whose computation will be the closest to $\mathcal{T}$'s answer. Consequently, HB-based protocols provide tag authentication, but $\mathcal{R}$ is not sure of the real identity of a tag.

To counter this issue, we associate a unique secret identifier per tag, known by $\mathcal{R}$. Each tag will use its own identifier to compute its answer, and $\mathcal{R}$ will be able to uniquely identify each tag.

2.3.2 Protocol Description

We propose a variant of the LPN-C protocol proposed by Gilbert, Robshaw and Seurin in [72] that provides low reader complexity and correct tag identification. In our proposal, the tag answer is built in the same way as for HB$^+$ (see Section 2.1.1). We further add a challenge sent by the reader to the tag to avoid the problem of replay attacks that are inherent in LPN-C. Then contrary to LPN-C, our protocol challenges are matrix whereas our protocol secrets are vectors defined as in HB$^+$, i.e., two secret keys instead of one. This choice is to store less information than LPN-C on the tag, and thus minimize the tag memory needed by the protocol. Consequently, such a choice reduces the potential price of the tag, since tags with few memory are generally low-cost. The final achievement of our protocol is to allow a reader to correctly authenticate and identify a tag.

Initialization. When the system is set up, the reader and all the tags share a pair $(\mathsf{k}_1, \mathsf{k}_2)$ of secret keys. Each tag $\mathcal{T}$ is assigned to a unique secret identifier $\mathsf{ID}_\mathcal{T}$ known by $\mathcal{R}$, where $|\mathsf{ID}_\mathcal{T}| = \lambda$. The notations and values that will be used in the protocol are given below.

Notation	Meaning	Usual choices
k_1	length of the key k_1	$k_1 = 80$
k_2	length of the key k_2	$k_2 \in [224, 512]$
λ	system security parameter	$\lambda \in [80, 212]$
ϵ	noise parameter	$\epsilon \in [\frac{1}{8}, \frac{1}{4}]$

Let define $\mathfrak{C}$ as the code of all the tag identifiers of our system. For a given codeword $\mathsf{ID}_\mathcal{T} \in \mathfrak{C}$, let consider $\mathfrak{B}_{\mathsf{ID}_\mathcal{T}}$ as being the ball $\mathfrak{B}(\mathsf{ID}_T, \varrho)$ of radius $\varrho = \lceil \lambda\epsilon \rceil$ around $\mathsf{ID}_\mathcal{T}$. Each ball represents all the codewords

$\mathfrak{c}$ such that $w_{\mathcal{H}}(\mathsf{ID}_{\mathcal{T}} \oplus \mathfrak{c}) \leq \varrho$, where $w_{\mathcal{H}}$ denotes the Hamming weight. The volume of $\mathfrak{B}_{\mathsf{ID}_{\mathcal{T}}}$ is the number of all these codewords $\mathfrak{c}$ defined as:

$$\mathrm{Vol}(\mathfrak{B}_{\mathsf{ID}_{\mathcal{T}}}) = \mathrm{Vol}(\mathfrak{B}(0, \varrho)) = \sum_{i=0}^{\varrho} \binom{\lambda}{i}.$$

To make viable tag identification, the identifiers are distributed such that all the balls are pairwise disjoint.

Authentication. The protocol is given in Figure 2.2.

Reader $\mathcal{R}$ $\mathsf{k}_1, \mathsf{k}_2, \mathsf{ID}_{\mathcal{T}}, \epsilon$		**Tag** $\mathcal{T}$ $\mathsf{k}_1, \mathsf{k}_2, \mathsf{ID}_{\mathcal{T}}, \epsilon$
• $M_{\mathcal{R}} \in_R \{0,1\}^{\lambda} \times \{0,1\}^{k_1}$		
	$\xrightarrow{M_{\mathcal{R}}}$	
		• $M_{\mathcal{T}} \in_R \{0,1\}^{\lambda} \times \{0,1\}^{k_2}$ • $\nu = (\nu_1, \nu_2, \ldots, \nu_{\lambda})$ s.t. $\nu_i \in \{0, 1 \mid \Pr(\{\nu_i = 1\}) = \epsilon\}$ • $E = (M_{\mathcal{R}} \cdot \mathsf{k}_1) \oplus (M_{\mathcal{T}} \cdot \mathsf{k}_2)$ $\oplus\ \mathsf{ID}_{\mathcal{T}} \oplus \nu$
	$\xleftarrow{M_{\mathcal{T}}, E}$	
• $E' = (M_{\mathcal{R}} \cdot \mathsf{k}_1) \oplus (M_{\mathcal{T}} \cdot \mathsf{k}_2)$ • Find $(\mathsf{ID}_{\mathcal{T}}) \in \mathsf{DB}$ s.t. $w_{\mathcal{H}}(E \oplus E' \oplus \mathsf{ID}_{\mathcal{T}}) \leq \varrho$		

Figure 2.2: Authentication protocol.

Remark. When the reader receives the tag answer, it computes $E' = (M_{\mathcal{R}} \cdot \mathsf{k}_1) \oplus (M_{\mathcal{T}} \cdot \mathsf{k}_2)$, and recovers instantaneously $E \oplus E' = \mathsf{ID}_{\mathcal{T}} \oplus \nu$. Then for each identifier $\mathsf{ID}_{\mathcal{T}}$, $\mathcal{R}$ computes the Hamming distance between $E \oplus E'$ and $\mathsf{ID}_{\mathcal{T}}$. Since all the identifiers are well-distributed, when this distance is lower than ϱ, that means $E \oplus E'$ only belongs to $\mathfrak{B}_{\mathsf{ID}_{\mathcal{T}}}$. Thus $\mathcal{R}$ retrieves the real identifier $\mathsf{ID}_{\mathcal{T}}$.

This step of the authentication process can be improved. Clearly, $E \oplus E'$ is the tag identifier XORed with some noise vector ν, i.e., $\mathsf{ID}_{\mathcal{T}}$ containing at most ϱ error bits (ϱ being the Hamming weight of ν). Instead of computing naively the Hamming distance between $E \oplus E'$

and all the identifiers of the database, $\mathcal{R}$ can use an appropriate error-correcting code to recover $\mathsf{ID}_{\mathcal{T}}$ without the ϱ errors. This extension is out of the scope of this thesis, though.

2.3.3 Analysis

Besides the assumption that all the balls $\mathfrak{B}_{\mathsf{ID}_{\mathcal{T}}}$ are pairwise disjoint, let assume that (i) the identifiers space is large enough and, (ii) the tag identifiers are uniformly distributed for security reasons. First, the distance between two identifiers must be at least two times the radius of a ball, i.e., 2ϱ. This enables the reader to identify every tag without any mistake, since every $E \oplus E'$ result belongs to a unique ball. However, if the identifiers space is too small and if all the balls exactly cover the space, the security is nonexistent: an adversary can send a value at random and be sure to be identified by the reader. Hence, the success probability of an adversary to send a random value that could match a result into a ball should be negligible. Therefore, the identifiers space must be large enough.

We compare the practicability of our protocol to the one of HSH when considering a system of $n = \beta^d = 10^6$ tags. When the parameters β, d, ϵ, k_1 and k_2 are fixed (to the values given by the authors of HSH), the following results (illustrated in Table 2.1) are worth to be mentioned.

- The FAR[1] is equal to $n\dfrac{\mathrm{Vol}(\mathfrak{B}(0,\varrho))}{2^\lambda}$ for our protocol, and equal to $\dfrac{\mathrm{Vol}(\mathfrak{B}(0,\varrho))}{2^\lambda}$ for HSH.
- The memory needed on the tag is $k_1 + k_2 + \lambda$ bits for our protocol, and $k_1 + k_2(d+1)$ bits for HSH.
- The number $\mathrm{Com}_{\mathrm{bit}}$ of bits exchanged during a protocol execution is equal to $\lambda(k_1 + k_2 + 1)$ bits for our protocol, and equal to $\lambda(k_1 + k_2 + d + 1)$ bits for HSH.
- The tag computation complexity $\mathbb{C}_{\mathcal{T}}$ is of $\lambda(k_1 + k_2)$ bit operations for our protocol, and of $\lambda d k_2 + \lambda(k_1 + k_2)$ bit operations for HSH.

[1] False Accept Rate, i.e., the probability to guess a correct tag answer at random.

- The reader computation complexity $\mathbb{C}_R$ is of $\lambda(k_1 + k_2)$ bit operations for our protocol (plus the decoding method to retrieve $\mathsf{ID}_\mathcal{T}$), and of $\beta\lambda d k_2 + \lambda(k_1 + k_2)$ bit operations for HSH.

	FAR	λ	Tag memory	Com_{bit}	$\mathbb{C}_\mathcal{T}$	$\mathbb{C}_R$
HSH	$2^{-41.3}$	86	1400	44978	$2^{16.9}$	$2^{26.2}$
Our protocol	$2^{-41.5}$	128	648	66688	2^{16}	

Table 2.1: Comparison of HSH and our protocol when $\beta = 1000$, $d = 2$, $\epsilon = 0.125$, $k_1 = 80$ and $k_2 = 440$.

In our protocol, λ must be a large number to reach the same security level as HSH (i.e., FAR $\approx 2^{-41}$). This leads to the only drawback of our protocol in this comparison with HSH: the number Com_{bit} of bits exchanged for one execution of our protocol is almost the double than for HSH. However, our protocol needs less tag memory to achieve the same security level (around half less). The reader and tag complexities to process our protocol are also lower than the ones for HSH. This conclusion is further observable at the reader side: the reader tree traversal stage of HSH to find the right pair of secret keys increases consequently the reader time search.

2.4 Attack on HSH

In Section 2.3, we assumed that tags are tamper-resistant, and we provided a protocol better than HSH under this assumption. Now, we consider that tags are no longer tamper-resistant, and we show that corrupting a few tags (i.e., revealing their keys by tampering with them) smashes the privacy of an RFID system $\mathcal{S}$ based on HSH. In particular, we demonstrate that (i) HSH is predisposed to tag traceability when an adversary $\mathcal{A}$ can tamper with one tag, and that (ii) this situation is further worse when $\mathcal{A}$ can tamper with several tags.

2.4.1 Adversary Game

As explained in Section 2.2.2, ADO [15] has been the first published traceability attack on tree-based protocols. In the sequel, we show that

ADO can be generalized to break the untraceability (UNT) property of HSH, and thus its privacy.

Let consider that $\mathcal{A}$ performs her attack against an honest entity, called *challenger* $\mathcal{C}$, following the experiment $Exp^{\mathsf{HSH-UNT}}_{\mathcal{S},\mathcal{A}}$ detailed in Figure 2.3. For the sake of clarity, $Exp^{\mathsf{HSH-UNT}}_{\mathcal{S},\mathcal{A}}$ is simplified to the case of tampering with one tag.

Experiment $Exp^{\mathsf{HSH-UNT}}_{\mathcal{S},\mathcal{A}}[\lambda]$

1. $\mathcal{C}$ initializes the system $\mathcal{S}$, randomly chooses one tag $\mathcal{T}_c$, and gives it to $\mathcal{A}$.
2. $\mathcal{A}$ interacts with $\mathcal{T}_c$, and possibly corrupts it.
3. $\mathcal{A}$ gives back $\mathcal{T}_c$ to $\mathcal{C}$ who randomly chooses a target tag $\mathcal{T}$, and gives it to $\mathcal{A}$.
4. $\mathcal{A}$ interacts with $\mathcal{T}$, but cannot tamper with it.
5. $\mathcal{A}$ gives back $\mathcal{T}$ to $\mathcal{C}$ who chooses a random bit b and two tags $\mathcal{T}_0$ and $\mathcal{T}_1$ such that $\mathcal{T} = \mathcal{T}_b$, and gives $\mathcal{T}_0$ and $\mathcal{T}_1$ to $\mathcal{A}$.
6. $\mathcal{A}$ interacts with $\mathcal{T}_0$ and $\mathcal{T}_1$ without tampering with them.
7. $\mathcal{A}$ outputs a guess bit b'.

$Exp^{\mathsf{HSH-UNT}}_{\mathcal{S},\mathcal{A}}$ succeeds if $b = b'$.

Figure 2.3: Untraceability experiment of HSH.

The untraceability of the system $\mathcal{S}$ is ensured if:

$$\left|\Pr(Exp^{\mathsf{HSH-UNT}}_{\mathcal{S},\mathcal{A}}[\lambda] \text{ succeeds}) - \frac{1}{2}\right| \leq \epsilon(\lambda)$$

where ϵ is a negligible function in the security parameter λ.

2.4.2 Tampering with one Tag

The purpose of this section is to evaluate the probability that the experiment $Exp^{\mathsf{HSH-UNT}}_{\mathcal{S},\mathcal{A}}$ succeeds. To formalize the analysis, we denote

the keys of $\mathcal{T}$, $\mathcal{T}_0$, $\mathcal{T}_1$ and $\mathcal{T}_c$ by $[\mathsf{k}^1, \ldots, \mathsf{k}^d]$, $[\mathsf{k}_0^1, \ldots, \mathsf{k}_0^d]$, $[\mathsf{k}_1^1, \ldots, \mathsf{k}_1^d]$ and $[\mathsf{k}_c^1, \ldots, \mathsf{k}_c^d]$ respectively. At a given level i in the tree ($1 \leq i \leq d$), the ADO attack considers four possibilities:

- $\mathrm{A}_1^i = \{\mathsf{k}_c^i = \mathsf{k}_0^i\} \wedge \{\mathsf{k}_c^i \neq \mathsf{k}_1^i\}$,
- $\mathrm{A}_2^i = \{\mathsf{k}_c^i \neq \mathsf{k}_0^i\} \wedge \{\mathsf{k}_c^i = \mathsf{k}_1^i\}$,
- $\mathrm{A}_3^i = \{\mathsf{k}_c^i \neq \mathsf{k}_0^i\} \wedge \{\mathsf{k}_c^i \neq \mathsf{k}_1^i\}$,
- $\mathrm{A}_4^i = \{\mathsf{k}_c^i = \mathsf{k}_0^i = \mathsf{k}_1^i\}$.

The ADO attack is performed on a classical challenge/response protocol using a pseudo-random function. Here, the HSH tree traversal stage is based on a challenge/response protocol using HB$^+$. The noise inherent to HB$^+$ does not allow to apply the ADO attack directly.

We first define $E_\ell^i = M_{\mathcal{T}_\ell}\mathsf{k}_\ell^i \oplus \nu_\ell^i$ as the answer of the tag $\mathcal{T}_\ell$ at level i of the HSH tree traversal stage. Then, we define $\mathfrak{B}_\ell^i$ as the ball of radius $\varrho = \lceil \lambda\epsilon \rceil$ (as defined in Section 2.3.3) around E_ℓ^i. A direct consequence of HSH is that $\mathcal{A}$ can only evaluate the possibility that $\mathcal{T}_c$'s key k_c^i was used to compute E_ℓ^i, i.e., $M_{\mathcal{T}_\ell}\mathsf{k}_c^i \in \mathfrak{B}_\ell^i$ or not. For our attack, we thus consider the four subsequent cases related to the ADO ones:

- $\mathrm{C}_1^i = \{M_{\mathcal{T}_0}\mathsf{k}_c^i \in \mathfrak{B}_0^i\} \wedge \{M_{\mathcal{T}_1}\mathsf{k}_c^i \notin \mathfrak{B}_1^i\}$,
- $\mathrm{C}_2^i = \{M_{\mathcal{T}_0}\mathsf{k}_c^i \notin \mathfrak{B}_0^i\} \wedge \{M_{\mathcal{T}_1}\mathsf{k}_c^i \in \mathfrak{B}_1^i\}$,
- $\mathrm{C}_3^i = \{M_{\mathcal{T}_0}\mathsf{k}_c^i \notin \mathfrak{B}_0^i\} \wedge \{M_{\mathcal{T}_1}\mathsf{k}_c^i \notin \mathfrak{B}_1^i\}$,
- $\mathrm{C}_4^i = \{M_{\mathcal{T}_0}\mathsf{k}_c^i \in \mathfrak{B}_0^i\} \wedge \{M_{\mathcal{T}_1}\mathsf{k}_c^i \in \mathfrak{B}_1^i\}$.

Consequently, the events that are taken into account for this attack are:

- $\mathrm{Ev}_1^i = \mathrm{A}_1^i \wedge \mathrm{C}_1^i$, then the attack succeeds,
- $\mathrm{Ev}_2^i = \mathrm{A}_2^i \wedge \mathrm{C}_2^i$, then the attack succeeds,
- $\mathrm{Ev}_3^i = \mathrm{A}_3^i \wedge \mathrm{C}_3^i$, then the attack definitely fails,
- $\mathrm{Ev}_4^i = \mathrm{A}_4^i \wedge \mathrm{C}_4^i$, then the attack fails at level i but can move to level $i+1$,

where all the Ev_j^i events are pairwise disjoint. The success of this attack means that $\mathcal{A}$ has been able to distinguish $\mathcal{T}_0$ from $\mathcal{T}_1$.

For the sake of clarity, we explicitly denote two probabilities to compare $\mathcal{T}_c$'s key k_c^i and a given $\mathcal{T}_\ell$'s key k_ℓ^i at level i.

- $\Pr(\{M_{\mathcal{T}_\ell}\mathsf{k}_c^i \in \mathfrak{B}_\ell^i | \{\mathsf{k}_c^i = \mathsf{k}_\ell^i\}) = \sum_{j=0}^{\varrho} \binom{\lambda}{j} \epsilon^j (1-\epsilon)^{\lambda-j} = S_\varrho$
- $\Pr(\{M_{\mathcal{T}_\ell}\mathsf{k}_c^i \in \mathfrak{B}_\ell^i | \{\mathsf{k}_c^i \neq \mathsf{k}_\ell^i\}) = \Pr(w_{\mathcal{H}}(M_{\mathcal{T}_\ell}\mathsf{k}_c^i \oplus E_\ell^i) \leq \varrho) = \dfrac{\mathrm{Vol}(\mathfrak{B}(0,\varrho))}{2^\lambda} = V_\varrho$

The probability of the event Ev_1^i is:

$$\begin{aligned}
\Pr(\mathrm{Ev}_1^i) &= \Pr(\mathrm{A}_1^i \wedge \mathrm{C}_1^i) \\
&= \Pr(\{\mathsf{k}_c^i = \mathsf{k}_0^i\} \wedge \{M_{\mathcal{T}_0}\mathsf{k}_c^i \in \mathfrak{B}_0^i\}) \\
&\quad \times \Pr(\{\mathsf{k}_c^i \neq \mathsf{k}_1^i\} \wedge \{M_{\mathcal{T}_1}\mathsf{k}_c^i \notin \mathfrak{B}_1^i\}) \\
&= \frac{1}{\beta} \times S_\varrho \times \frac{\beta-1}{\beta} \times (1 - V_\varrho) \quad \text{(Bayes' law)}.
\end{aligned}$$

The probabilities of the events Ev_2^i, Ev_3^i, Ev_4^i are computed in the same way. The final results are:

$$\begin{aligned}
\Pr(\mathrm{Ev}_1^i) = \Pr(\mathrm{Ev}_2^i) &= S_\varrho(1-V_\varrho)\left(\frac{\beta-1}{\beta^2}\right), \\
\Pr(\mathrm{Ev}_3^i) &= (1-V_\varrho)^2\left(\frac{\beta-1}{\beta}\right)^2, \\
\Pr(\mathrm{Ev}_4^i) &= \left(\frac{S_\varrho}{\beta}\right)^2.
\end{aligned}$$

Following the ADO attack, the overall probability $\mathrm{P}_{\mathrm{succ}}$ that the whole attack succeeds when the adversary tampers with one tag is:

$$\begin{aligned}
\mathrm{P}_{\mathrm{succ}} &= 2\Pr(\mathrm{Ev}_1^1) + \sum_{i=2}^{d}\Big(2\Pr(\mathrm{Ev}_1^i) \times \prod_{j=1}^{i-1}\Pr(\mathrm{Ev}_4^j)\Big) \\
&= 2S_\varrho(1-V_\varrho)\Big(\frac{\beta-1}{\beta^2}\Big)\left(\frac{1-(\frac{S_\varrho}{\beta})^{2d}}{1-(\frac{S_\varrho}{\beta})^2}\right).
\end{aligned}$$

2.4.3 Adversary Probability $\mathbf{P_{succ}}$ when Tampering with Several Tags

We now analyze the adversary success probability of tracing a tag when she tampers with α tags ($\alpha \geq 1$). As before, we denote the keys of $\mathcal{T}$, $\mathcal{T}_0$ and $\mathcal{T}_1$ by $[\mathsf{k}^1, \ldots, \mathsf{k}^d]$, $[\mathsf{k}_0^1, \ldots, \mathsf{k}_0^d]$, and $[\mathsf{k}_1^1, \ldots, \mathsf{k}_1^d]$ respectively. At a given level i of the tree, we consider $\mathcal{K}^i = \{\mathsf{k}^{i,1}, \mathsf{k}^{i,2}, \ldots, \mathsf{k}^{i,\alpha_i}\}$ the set of keys retrieved by $\mathcal{A}$ from the tags corrupted in step 2 of $Exp_{S,\mathcal{A}}^{\mathsf{HSH-UNT}}$. The ADO attack considers five possibilities at each level i of the tree:

- $\mathrm{A}_1^i = \{\mathsf{k}_0^i \in \mathcal{K}^i\} \wedge \{\mathsf{k}_1^i \notin \mathcal{K}^i\}$,
- $\mathrm{A}_2^i = \{\mathsf{k}_0^i \notin \mathcal{K}^i\} \wedge \{\mathsf{k}_1^i \in \mathcal{K}^i\}$,
- $\mathrm{A}_3^i = \{\mathsf{k}_0^i \in \mathcal{K}^i\} \wedge \{\mathsf{k}_1^i \in \mathcal{K}^i\} \wedge \{\mathsf{k}_0^i \neq \mathsf{k}_1^i\}$,
- $\mathrm{A}_4^i = \{\mathsf{k}_0^i \notin \mathcal{K}^i\} \wedge \{\mathsf{k}_1^i \notin \mathcal{K}^i\}$,
- $\mathrm{A}_5^i = \{\mathsf{k}_0^i \in \mathcal{K}^i\} \wedge \{\mathsf{k}_1^i \in \mathcal{K}^i\} \wedge \{\mathsf{k}_0^i = \mathsf{k}_1^i\}$.

In the same vein as the previous section, $\mathcal{A}$ can only evaluate if one of the retrieved keys (i.e., belonging to $\mathcal{K}^i$) has been used to compute E_ℓ^i. We thus consider the five subsequent cases related to the ADO ones:

- $\mathrm{C}_1^i = \{\exists \mathsf{k}^{i,j} \in \mathcal{K}^i \colon M_{\mathcal{T}_0} \mathsf{k}^{i,j} \in \mathfrak{B}_0^i\} \wedge \{\forall \mathsf{k}^{i,j} \in \mathcal{K}^i \colon M_{\mathcal{T}_1} \mathsf{k}^{i,j} \notin \mathfrak{B}_1^i\}$,
- $\mathrm{C}_2^i = \{\forall \mathsf{k}^{i,j} \in \mathcal{K}^i \colon M_{\mathcal{T}_0} \mathsf{k}^{i,j} \notin \mathfrak{B}_0^i\} \wedge \{\exists \mathsf{k}^{i,j} \in \mathcal{K}^i \colon M_{\mathcal{T}_1} \mathsf{k}^{i,j} \in \mathfrak{B}_1^i\}$,
- $\mathrm{C}_3^i = \{\exists \mathsf{k}^{i,j} \in \mathcal{K}^i \colon M_{\mathcal{T}_0} \mathsf{k}^{i,j} \in \mathfrak{B}_0^i\} \wedge \{\exists \mathsf{k}^{i,j'} \in \mathcal{K}^i \colon M_{\mathcal{T}_1} \mathsf{k}^{i,j'} \in \mathfrak{B}_1^i\}$ $\wedge \{\mathsf{k}^{i,j} \neq \mathsf{k}^{i,j'}\}$,
- $\mathrm{C}_4^i = \{\forall \mathsf{k}^{i,j} \in \mathcal{K}^i \colon M_{\mathcal{T}_0} \mathsf{k}^{i,j} \notin \mathfrak{B}_0^i\} \wedge \{\forall \mathsf{k}^{i,j} \in \mathcal{K}^i \colon M_{\mathcal{T}_1} \mathsf{k}^{i,j} \notin \mathfrak{B}_1^i\}$,
- $\mathrm{C}_5^i = \{\exists \mathsf{k}^{i,j} \in \mathcal{K}^i \colon M_{\mathcal{T}_0} \mathsf{k}^{i,j} \in \mathfrak{B}_0^i\} \wedge \{\exists \mathsf{k}^{i,j'} \in \mathcal{K}^i \colon M_{\mathcal{T}_1} \mathsf{k}^{i,j'} \in \mathfrak{B}_1^i\}$ $\wedge \{\mathsf{k}^{i,j} = \mathsf{k}^{i,j'}\}$.

Therefore, the events that are taken into account for this attack are:

- $\mathrm{Ev}_1^i = \mathrm{A}_1^i \wedge \mathrm{C}_1^i$, then the attack succeeds,
- $\mathrm{Ev}_2^i = \mathrm{A}_2^i \wedge \mathrm{C}_2^i$, then the attack succeeds,
- $\mathrm{Ev}_3^i = \mathrm{A}_3^i \wedge \mathrm{C}_3^i$, then the attack succeeds,

- $\mathrm{Ev}_4^i = \mathrm{A}_4^i \wedge \mathrm{C}_4^i$, then the attack definitely fails,
- $\mathrm{Ev}_5^i = \mathrm{A}_5^i \wedge \mathrm{C}_5^i$, then the attack fails at level i but can move to level $i+1$.

The probability $\Pr(\mathrm{Ev}_i^1)$ is computed as follows, where "$\exists!x$" denotes that "there exists a *unique* x".

$$\begin{aligned}
\Pr(\mathrm{Ev}_1^i) &= \Pr(\mathrm{A}_1^i \wedge \mathrm{C}_1^i) \\
&= \Pr(\{\mathsf{k}_0^i \in \mathcal{K}^i\} \wedge \{\exists \mathsf{k}^{i,j} \in \mathcal{K}^i \colon M_{\mathcal{T}_0}\mathsf{k}^{i,j} \in \mathfrak{B}_0^i\}) \\
&\quad \times \Pr(\{\mathsf{k}_1^i \notin \mathcal{K}^i\} \wedge \{\forall \mathsf{k}^{i,j} \in \mathcal{K}^i \colon M_{\mathcal{T}_1}\mathsf{k}^{i,j} \notin \mathfrak{B}_1^i\}) \\
&= \frac{\alpha_i}{\beta} \times (1-(1-S_\varrho)^{\alpha_i}) \times \frac{\beta-\alpha_i}{\beta} \\
&\quad \times (1 - \Pr(\{\exists! \mathsf{k}^{i,j} \in \mathcal{K}^i \colon M_{\mathcal{T}_1}\mathsf{k}^{i,j} \in \mathfrak{B}_1^i\} | \{\mathsf{k}_1^i \notin \mathcal{K}^i\}))^{\alpha_i} \\
&= \frac{\alpha_i}{\beta} \times (1-(1-S_\varrho)^{\alpha_i}) \times \frac{\beta-\alpha_i}{\beta} \times (1-V_\varrho)^{\alpha_i}
\end{aligned}$$

The probabilities of the events Ev_2^i, Ev_3^i, Ev_4^i and Ev_5^i at level i are computed in the same way. The results are the following ones.

$$\begin{aligned}
\Pr(\mathrm{Ev}_1^i) = \Pr(\mathrm{Ev}_2^i) &= \left(\frac{\alpha_i(\beta-\alpha_i)}{\beta^2}\right)\Big(1-(1-S_\varrho)^{\alpha_i}\Big)(1-V_\varrho)^{\alpha_i} \\
\Pr(\mathrm{Ev}_3^i) &= \left(\frac{\alpha_i(\alpha_i-1)}{\beta^2}\right)\Big(1-(1-S_\varrho)^{\alpha_i}\Big)^2 \\
\Pr(\mathrm{Ev}_4^i) &= \left(\frac{\beta-\alpha_i}{\beta}\right)^2 (1-V_\varrho)^{2\alpha_i} \\
\Pr(\mathrm{Ev}_5^i) &= \left(\frac{\alpha_i}{\beta^2}\right)\Big(1-(1-S_\varrho)^{\alpha_i}\Big)^2
\end{aligned}$$

Following the ADO attack, the overall probability $\mathrm{P}_{\mathrm{succ}}$ that the whole attack succeeds when the adversary tampers with α tags is:

$$\begin{aligned}
\mathrm{P}_{\mathrm{succ}} &= \Pr(\mathrm{Ev}_1^1 \vee \mathrm{Ev}_2^1 \vee \mathrm{Ev}_3^1) \\
&\quad + \sum_{i=2}^{d} \left(\Pr(\mathrm{Ev}_1^i \vee \mathrm{Ev}_2^i \vee \mathrm{Ev}_3^i) \times \prod_{j=1}^{i-1} \Pr(\mathrm{Ev}_5^j) \right) \qquad (2.1)
\end{aligned}$$

where α_i, the number of different keys known by the adversary at level i, is given by the ADO attack:

$$\alpha_1 = \beta\left(1-\left(\frac{\beta-1}{\beta}\right)^{\alpha}\right) \quad \text{and} \quad \alpha_i = \beta\left(1-\left(\frac{\beta-1}{\beta}\right)^{g(\alpha_i)}\right) \quad (2 \leq i \leq d)$$
$$\text{where} \quad g(\alpha_i) = \alpha \prod_{\ell=1}^{i-1} \frac{1}{\alpha_\ell}.$$

Remark. When $\epsilon = 0$, there is a perfect match between the ADO attack and ours. In such a case, no noise is added in tag answers which influences the values of S_ϱ and V_ϱ ($S_\varrho = 1$ and $V_\varrho = 0$).

Table 2.2 gives numerical values of Eq. 2.1 when the security and noise parameters are fixed in order to illustrate the efficiency of our attack. Note that the case $\alpha = 1$ represents the value of $\mathrm{P}_{\mathrm{succ}}$ given in Section 2.4.2.

α \ β	2	20	100	500	1000
1	33.7%	5.8%	1.2%	0.2%	0.1%
20	56.1%	84.3%	32.9%	7.7%	3.9%
50	56.3%	95.8%	63.0%	18.1%	9.5%
100	56.3%	96.9%	86.0%	33.0%	18.1%
200	56.3%	98.1%	97.4%	55.0%	33.0%

Table 2.2: Numerical values of the probability $\mathrm{P}_{\mathrm{succ}}$ of tracing a tag $\mathcal{T}$ according to branching factor β when the adversary tampers with α tags. The system contains 2^{20} tags, $\lambda = 86$ and $\epsilon = 0.125$.

2.4.4 Adversary Probability $\mathrm{P}_{\mathrm{luck}}$ when Tampering with Several Tags

Contrary to a classical tree-based protocol where $\mathcal{A}$ can always determine with probability 1 that a given key has been used to generate an answer, HSH does not provide such a deterministic verification procedure. In other words, $\mathcal{A}$ can be unlucky, meaning that she checks the right key but concludes that the key is wrong due to the noise. She can also be lucky, meaning that the noise makes her observe something wrong but, nevertheless, she provides the correct result.

$\mathrm{P}_{\mathrm{luck}}$ reflects the fact that $\mathcal{A}$ is lucky and finds the right answer even if she makes a mistake. It is divided in two events during the tree traversal:

- B = {$\mathcal{A}$ separates too soon $\mathcal{T}_0$ and $\mathcal{T}_1$, while this can be done later},
- L = {$\mathcal{A}$ separates too late $\mathcal{T}_0$ and $\mathcal{T}_1$},

which define the adversary probability of luck as:

$$\mathrm{P}_{\mathrm{luck}} = \Pr(\mathrm{B}) + \Pr(\mathrm{L}) \qquad \text{where } \mathrm{B} \wedge \mathrm{L} = \emptyset. \tag{2.2}$$

Figures 2.4 and 2.5 are illustrations of the events B and L. As legend, the branches _ _ _ represent the paths whose keys are known from tag corruption; $\xleftarrow{\mathcal{T}_0}$ and $\xleftarrow{\mathcal{T}_1}$ represent the paths supposed by $\mathcal{A}$ for $\mathcal{T}_0$ and $\mathcal{T}_1$, respectively.

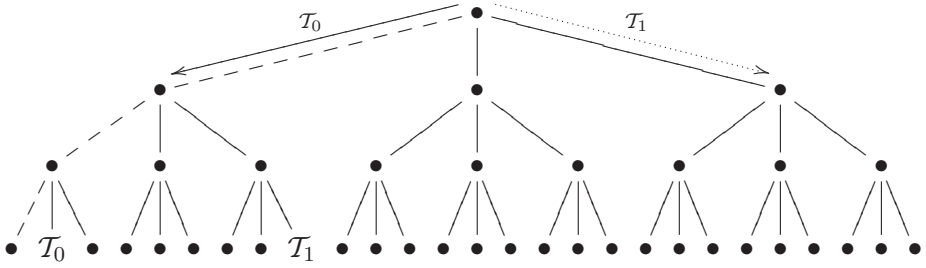


Figure 2.4: An example of the event B.

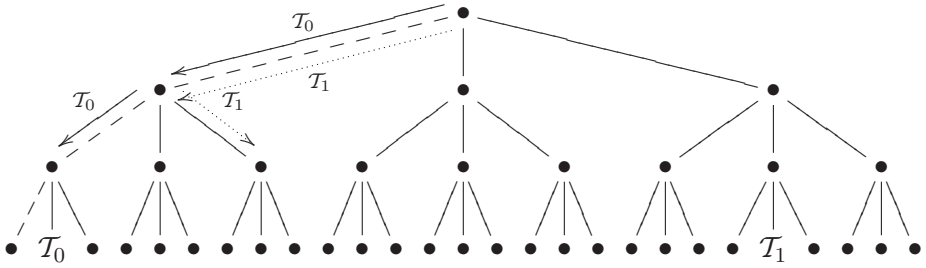


Figure 2.5: An example of the event L.

Below are the formulas for Pr(B) and Pr(L). Note that $\Pr(\mathrm{Ev}_5^0) = 1$.

$$\Pr(\mathrm{B}) \;=\; \sum_{i=1}^{d-2} \left(\Pr(\text{B-Sep}^i) \times \prod_{j=0}^{i-1} \Pr(\mathrm{Ev}_5^j) \right)$$

$$\Pr(\text{L}) \;=\; \sum_{i=2}^{d}\left(\Pr(\text{L-Sep}^i)\times\sum_{k=1}^{i-1}\left(\prod_{j=0}^{k-1}\Pr(\text{Ev}_5^j)\times\prod_{\substack{\ell=i-1\\ \ell=\ell-1}}^{k}\Pr(\text{L-Fol}^\ell)\right)\right)$$

Computation of B-Sep^i. This event is divided in four cases:

- $\text{BS}_1^i = \{\mathcal{T}_0 \text{ is identified by its real key } \mathsf{k}_0^i \in \mathcal{K}^i\} \wedge \{\mathcal{T}_1 \text{ is not identified at all (even if it should)}\} = \text{Normal}_{\text{BS}^i}^{\mathcal{T}_0} \wedge \text{False}_{\text{BS}^i}^{\mathcal{T}_1}$,
- $\text{BS}_2^i = \{\mathcal{T}_0 \text{ is identified by its real key } \mathsf{k}_0^i \in \mathcal{K}^i\} \wedge \{\mathcal{T}_1 \text{ is identified by a wrong known key } \mathsf{k}^{i,j} \neq \mathsf{k}_0^i\} = \text{Normal}_{\text{BS}^i}^{\mathcal{T}_0} \wedge \text{Fake}_{\text{BS}_2^i}^{\mathcal{T}_1}$,
- $\text{BS}_3^i = \{\mathcal{T}_0 \text{ is identified by a wrong known key}\} \wedge \{\mathcal{T}_1 \text{ is not identified at all (even if it should)}\} = \text{Fake}_{\text{BS}^i}^{\mathcal{T}_0} \wedge \text{False}_{\text{BS}^i}^{\mathcal{T}_1}$,
- $\text{BS}_4^i = \{\mathcal{T}_0 \text{ is identified by a wrong known key } \mathsf{k}^{i,j}\} \wedge \{\mathcal{T}_1 \text{ is identified by a wrong known key } \mathsf{k}^{i,j'} \neq \mathsf{k}^{i,j}\} = \text{Fake}_{\text{BS}^i}^{\mathcal{T}_0} \wedge \text{Fake}_{\text{BS}_4^i}^{\mathcal{T}_1}$.

The whole probability of this event is:

$$\Pr(\text{B-Sep}^i) \;=\; \Pr(\text{BS}_1^i) + \Pr(\text{BS}_3^i) + \Pr(\text{BS}_3^i) + \Pr(\text{BS}_4^i).$$

The probability of each sub-case can be decomposed as follows.

$$\begin{aligned}
\Pr(\text{False}_{\text{BS}^i}^{\mathcal{T}_1}) &= \Pr(\{\nexists \mathsf{k}^{i,j} \in \mathcal{K}^i \colon M_{\mathcal{T}_1}\mathsf{k}^{i,j} \in \mathfrak{B}_1^i\} \wedge \{\mathsf{k}_1^i \in \mathcal{K}^i\}) \\
&= \frac{\alpha_i(1-S_\varrho)^{\alpha_i}}{\beta} \\
\Pr(\text{Fake}_{\text{BS}_2^i}^{\mathcal{T}_1}) &= \Pr(\{\exists! \mathsf{k}^{i,j} \in \mathcal{K}^i \colon M_{\mathcal{T}_1}\mathsf{k}^{i,j} \in \mathfrak{B}_1^i\} \wedge \{\mathsf{k}_1^i \in \mathcal{K}^i \setminus \{\mathsf{k}^{i,j}, \mathsf{k}_0^i\}\}) \\
&= \frac{(\alpha_i - 2)V_\varrho}{\alpha_i} \\
\Pr(\text{Fake}_{\text{BS}^i}^{\mathcal{T}_0}) &= \Pr(\{\exists! \mathsf{k}^{i,j} \in \mathcal{K}^i \colon M_{\mathcal{T}_0}\mathsf{k}^{i,j} \in \mathfrak{B}_0^i\} \wedge \{\mathsf{k}_0^i \in \mathcal{K}^i \setminus \{\mathsf{k}^{i,j}\}\}) \\
&= \frac{(\alpha_i - 1)V_\varrho}{\alpha_i} \\
\Pr(\text{Fake}_{\text{BS}_4^i}^{\mathcal{T}_1}) &= \Pr(\text{Fake}_{\text{BS}^i}^{\mathcal{T}_0}) + \Pr(\text{Fake}_{\text{BS}_2^i}^{\mathcal{T}_1}) \;=\; \frac{(2\alpha_i - 3)V_\varrho}{\alpha_i} \\
\Pr(\text{Normal}_{\text{BS}^i}^{\mathcal{T}_0}) &= \Pr(\{\exists! \mathsf{k}^{i,j} \in \mathcal{K}^i \colon M_{\mathcal{T}_0}\mathsf{k}^{i,j} \in \mathfrak{B}_0^i\} \wedge \{\mathsf{k}_0^i = \mathsf{k}^{i,j} \in \mathcal{K}^i\}) \\
&= \frac{S_\varrho}{\alpha_i}
\end{aligned}$$

Computation of L-Sep^i. This event is divided in four cases where $\mathcal{T}_1$ is no longer part of the current sub-tree:

- $\mathrm{LS}_1^i = \{\mathcal{T}_0$ is identified by its real key $\mathsf{k}_0^i \in \mathcal{K}^i\} \wedge \{\mathcal{T}_1$ is not identified at all$\} = \mathrm{Normal}_{\mathrm{LS}^i}^{\mathcal{T}_0} \wedge \mathrm{False}_{\mathrm{LS}^i}^{\mathcal{T}_1}$,
- $\mathrm{LS}_2^i = \{\mathcal{T}_0$ is identified by its real key $\mathsf{k}_0^i \in \mathcal{K}^i\} \wedge \{\mathcal{T}_1$ is identified by a wrong known key $\mathsf{k}^{i,j} \neq \mathsf{k}_0^i\} = \mathrm{Normal}_{\mathrm{LS}^i}^{\mathcal{T}_0} \wedge \mathrm{Fake}_{\mathrm{LS}^i}^{\mathcal{T}_1}$,
- $\mathrm{LS}_3^i = \{\mathcal{T}_0$ is identified by a wrong known key$\} \wedge \{\mathcal{T}_1$ is not identified at all$\} = \mathrm{Fake}_{\mathrm{LS}^i}^{\mathcal{T}_0} \wedge \mathrm{False}_{\mathrm{LS}^i}^{\mathcal{T}_1}$,
- $\mathrm{LS}_4^i = \{\mathcal{T}_0$ is identified by a wrong known key $\mathsf{k}^{i,j}\} \wedge \{\mathcal{T}_1$ is identified by a wrong known key $\mathsf{k}^{i,j'} \neq \mathsf{k}^{i,j}\} = \mathrm{Fake}_{\mathrm{LS}^i}^{\mathcal{T}_0} \wedge \mathrm{Fake}_{\mathrm{LS}^i}^{\mathcal{T}_1}$.

The whole probability of this event is:

$$\Pr(\text{L-Sep}^i) = \Pr(\mathrm{LS}_1^i) + \Pr(\mathrm{LS}_2^i) + \Pr(\mathrm{LS}_3^i) + \Pr(\mathrm{LS}_4^i)$$

The probability of each sub-case can be decomposed as follows.

$$\begin{aligned}
\Pr(\mathrm{Normal}_{\mathrm{LS}^i}^{\mathcal{T}_0}) &= \Pr(\mathrm{Normal}_{\mathrm{BS}^i}^{\mathcal{T}_0}) = \frac{S_\varrho}{\alpha_i} \\
\Pr(\mathrm{Fake}_{\mathrm{LS}^i}^{\mathcal{T}_1}) &= \Pr(\{\exists! \mathsf{k}^{i,j} \in \mathcal{K}^i \colon M_{\mathcal{T}_1}\mathsf{k}^{i,j} \in \mathfrak{B}_1^i\} \wedge \{\mathsf{k}_1^i \notin \mathcal{K}^i\}) \\
&= \frac{(\beta - \alpha_i)V_\varrho}{\beta} \\
\Pr(\mathrm{Fake}_{\mathrm{LS}^i}^{\mathcal{T}_0}) &= \Pr(\mathrm{Fake}_{\mathrm{BS}^i}^{\mathcal{T}_0}) = \frac{(\alpha_i - 1)V_\varrho}{\alpha_i} \\
\Pr(\mathrm{False}_{\mathrm{LS}^i}^{\mathcal{T}_1}) &= \Pr(\{\nexists \mathsf{k}^{i,j} \in \mathcal{K}^i \colon M_{\mathcal{T}_1}\mathsf{k}^{i,j} \in \mathfrak{B}_1^i\} \wedge \{\mathsf{k}_1^i \notin \mathcal{K}^i\}) \\
&= \frac{(\beta - \alpha_i)(1 - V_\varrho)^{\alpha_i}}{\beta}
\end{aligned}$$

Computation of L-Fol$^\ell$. This event is defined as $\{\mathcal{T}_0$ is identified by its real key $\mathsf{k}_0^\ell \in \mathcal{K}^\ell\} \wedge \{\mathcal{T}_1$ follows the same branch as $\mathcal{T}_0$ at level ℓ (which is a false branch for $\mathcal{T}_1)\} = \mathrm{Normal}_{\mathrm{LS}^\ell}^{\mathcal{T}_0} \wedge \mathrm{Fol}_\ell^{\mathcal{T}_1}$. The probability of these events are:

$$\begin{aligned}
\Pr(\mathrm{Fol}_\ell^{\mathcal{T}_1}) &= \Pr(\mathrm{Fake}_{\mathrm{LS}^\ell}^{\mathcal{T}_0}) + \Pr(\mathrm{Fake}_{\mathrm{LS}^\ell}^{\mathcal{T}_1}) = V_\varrho\left(\frac{\alpha_\ell - 1}{\alpha_\ell} + \frac{\beta - \alpha_\ell}{\beta}\right), \\
\Pr(\text{L-Fol}^\ell) &= \frac{S_\varrho\, V_\varrho}{\alpha_\ell^2}\left(\frac{\alpha_\ell - 1}{\alpha_\ell} + \frac{\beta - \alpha_\ell}{\beta}\right).
\end{aligned}$$

Table 2.3 gives numerical values of Eq. 2.2 when the security and noise parameters are fixed in order to illustrate that the adversary luck is a non-negligible element of the attack success. However, we only give values for $\beta = \{2, 20, 100\}$ because they are not significant for a larger β.

α \ β	2	20	100
1	13.10%	1.189%	0.238%
20	7.33%	0.023%	$1.2 * 10^{-10}$%
50	7.25%	0.012%	$5.9 * 10^{-11}$%
100	7.23%	$2.7 * 10^{-7}$%	$3.9 * 10^{-11}$%
200	7.22%	$1.0 * 10^{-7}$%	$2.9 * 10^{-11}$%

Table 2.3: Numerical values of the probability $\mathrm{P}_{\mathrm{luck}}$ of tracing a tag $\mathcal{T}$ according to branching factor β when the adversary tampers with α tags. The system contains 2^{20} tags, $\lambda = 86$ and $\epsilon = 0.125$.

2.4.5 Adversary Probability $\mathrm{P}_{\mathbf{fail}}$ when Tampering with Several Tags

During her attack, $\mathcal{A}$ may not be able to take any rational decision given her observations. The probability $\mathrm{P}_{\mathrm{fail}}$ corresponds to theses special cases where $\mathcal{A}$ correctly answers at random at step 7 of $Exp^{\mathsf{HSH-UNT}}_{\mathcal{S},\mathcal{A}}$.

$$\mathrm{P}_{\mathrm{fail}} = \frac{1}{2}\left(\Pr(\mathrm{Ev}_4^1) + \sum_{i=2}^{d-1}\left(\Pr(\mathrm{Ev}_4^i) \times \Pr(\mathrm{Cont}^{i-1})\right)\right) \tag{2.3}$$

In details, Cont^i is the event where $\mathcal{A}$ continues the attack until level i, and is composed of the following three main cases:

- for the i-th levels, $\mathcal{T}_0$ and $\mathcal{T}_1$ are identified by their real key: $\{\mathsf{k}_0^i \in \mathcal{K}^i\} \wedge \{\mathsf{k}_1^i \in \mathcal{K}^i\} \wedge \{\mathsf{k}_0^i = \mathsf{k}_1^i\} = \mathrm{Ev}_5^i$,
- for the i-th levels, $\mathcal{T}_0$ and $\mathcal{T}_1$ are identified by the same key, which is the real one only for $\mathcal{T}_0 = \text{L-Fol}^i$,
- for the i-th levels, $\mathcal{T}_0$ and $\mathcal{T}_1$ are identified by the same wrong key $= \mathrm{Wrg}^i = \mathrm{Fol}_i^{\mathcal{T}_0} \wedge \mathrm{Fol}_i^{\mathcal{T}_1} \wedge \{\mathsf{k}^{i,j} = \mathsf{k}^{i,j'}\}$,

and of the ordered combinations of theses cases: for $1 \leq \ell < k < j \leq i$, $\{\text{L-Fol}^k\}$ can be only preceded by $\{\text{Ev}_5^\ell\}$, and $\{\text{Wrg}^j\}$ can be preceded by $\{\text{L-Fol}^k\}$ or $\{\text{Ev}_5^\ell\}$ or $\{\text{L-Fol}^k \text{ and } \text{Ev}_5^\ell\}$. Thus:

$$\begin{aligned}
\Pr(\text{Cont}^i) &= \prod_{j=1}^{i}\Big(\Pr(\text{Ev}_5^j) + \Pr(\text{L-Fol}^j) + \Pr(\text{Wrg}^j)\Big) \\
&+ \sum_{j=1}^{i-1}\Bigg(\prod_{k=1}^{j}\Pr(\text{L-Fol}^k) \times \prod_{\ell=j+1}^{i}\Pr(\text{Wrg}^\ell)\Bigg) \\
&+ \sum_{j=1}^{i-1}\Bigg(\prod_{k=1}^{j}\Pr(\text{Ev}_5^k) \times \prod_{\ell=j+1}^{i}\Big(\Pr(\text{L-Fol}^\ell) + \Pr(\text{Wrg}^\ell)\Big)\Bigg) \\
&+ \sum_{j=1}^{i-2}\Bigg(\prod_{k=1}^{j}\Pr(\text{Ev}_5^k) \times \sum_{\ell=j+1}^{i-1}\Big(\prod_{m=j+1}^{\ell}\Pr(\text{L-Fol}^m) \times \prod_{p=\ell+1}^{i}\Pr(\text{Wrg}^p)\Big)\Bigg).
\end{aligned}$$

Table 2.4 gives numerical values of Eq. 2.3 when the security and noise parameters are fixed in order to illustrate that the adversary random answer (as ultimate choice) is also a non-negligible element of the attack success. Like with the table of P_{luck}, we decide to only give values for $\beta = \{2, 20, 100\}$ because they are not significant for a larger β.

α \ β	2	20	100
1	13.79%	45.17%	49.01%
20	0.16%	7.79%	33.45%
50	0.04%	2.06%	18.30%
100	0.02%	1.51%	6.70%
200	0.01%	0.94%	1.28%

Table 2.4: Numerical values of the probability P_{fail} of tracing a tag $\mathcal{T}$ according to branching factor β when the adversary tampers with α tags. The system contains 2^{20} tags, $\lambda = 86$ and $\epsilon = 0.125$.

2.4.6 Overall Adversary Advantage

When tampering with several tags, the adversary advantage to trace one tag is defined as:

$$Adv_{\mathcal{A}} = \big|2\Pr(Exp_{\mathcal{S},\mathcal{A}}^{\mathsf{HSH-UNT}}[\lambda] \text{ succeeds}) - 1\big| \tag{2.4}$$

where

$$\Pr(Exp^{\mathsf{HSH-UNT}}_{\mathcal{S},\mathcal{A}}[\lambda] \text{ succeeds}) = (\mathrm{P}_{\mathrm{succ}} + \mathrm{P}_{\mathrm{luck}} + \mathrm{P}_{\mathrm{fail}}).$$

Table 2.5 gives numerical values of Eq. 2.4 to illustrate our attack on HSH when the security and noise parameters are fixed. The advantage $Adv_{\mathcal{A}}$ is plotted in Figure 2.6.

α \ β	2	20	100	500	1000
1	0.2122	0.0434	0.0091	0.0018	0.0004
20	0.2716	0.8422	0.3274	0.0768	0.0392
50	0.2712	0.9571	0.6262	0.1810	0.0951
100	0.2711	0.9692	0.8536	0.3292	0.1811
200	0.2711	0.9811	0.9654	0.5497	0.3294

Table 2.5: Numerical values of the adversary advantage of tracing a tag $\mathcal{T}$ according to branching factor β when the adversary tampers with α tags. The system contains 2^{20} tags, $\lambda = 86$ and $\epsilon = 0.125$.

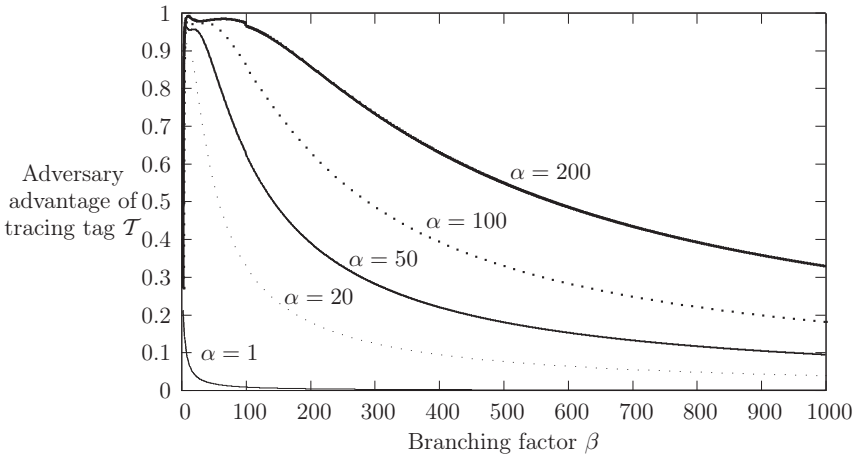


Figure 2.6: Adversary advantage of tracing a tag $\mathcal{T}$ when $\mathcal{A}$ tampers with α tags in a system of 2^{20} tags, where $\lambda = 86$ and $\epsilon = 0.125$.

For a given branching factor β, $Adv_{\mathcal{A}}$ increases when α increases. This outcome seams reasonable: clearly, the more tags an adversary $\mathcal{A}$

corrupts, the more secret keys she knows, which implies that the higher probability the attack success will be.

For a given number α of corrupted tags, $Adv_{\mathcal{A}}$ decreases when the branching factor β increases. This is a normal behavior of the advantage. For each level i of the key-tree composed of β branches, $\mathcal{A}$ knows a number α_i of branch keys over the β ones (where $\alpha_i \leq \beta$). When β augments, the ratio $\frac{\alpha_i}{\beta}$ of known keys at level i decreases, implying that the advantage of tracing a tag at level i (and thus the whole advantage) also decreases.

Yet, the following special case is worth to be distinguished from the previous explanations and clarified. Actually, when α is greater than 20 and the branching factor β is around 9, $Adv_{\mathcal{A}}$ reaches its maximum which is greater than 0.9. This result comes from two facts. Firstly, the numerical value of $\mathrm{P_{succ}}$ is the one that mainly determines the value of $Adv_{\mathcal{A}}$ for such a β, i.e., $\mathrm{P_{succ}}$ is greater than 0.9 while $\mathrm{P_{luck}}$ and $\mathrm{P_{fail}}$ are less than 0.1. Secondly, the value of $\mathrm{P_{succ}}$ mainly depends on the value of α. In fact, α is used to compute the value of the different α_i where $2 \leq i \leq d$ (see Section 2.4.3 for more details). When β is around 9, we can note that the values of α_i are almost identical for every α greater than 20, and so are the values of $\mathrm{P_{succ}}$. Consequently, when β is around 9, the adversary has nearly the same highest advantage to trace a tag whether she corrupts 20 or 200 tags.

Surprisingly, the results of our attack on HSH are quite similar to the ones of ADO performed on a classical tree-based protocol. Yet, one could have expected that the use of an HB-based protocol (i.e., built from the LPN problem) would strengthen the resistance of HSH to traceability attacks in comparison to a classical tree-based protocol. Our results clearly conjecture that this intuition is not accurate.

Chapter 3

RFID Privacy Models

One of the major concerns of cryptography and more generally information security is to establish proofs of security. Such proofs only make sense if they are built in a well-established model, and not in an ad-hoc one as performed in Chapter 2.

This chapter investigates the field of RFID privacy models, and chronologically presents eight well-known models designed to analyze systems based on identification/authentication protocols preserving privacy. Some of them are very popular like [9, 99, 162]. Other ones have interesting frameworks [44, 47, 107] or are valuable successors of [162], such as [34, 82].

3.1 Common Definitions

This section formally introduces the building blocks that are common to the presented privacy models.

3.1.1 The RFID System

All the privacy models presented in this chapter consider that an RFID system $\mathcal{S}$ is composed of a set of n tags and a single reader $\mathcal{R}$ that contains the system database DB. A tag is considered as *legitimate* when it is registered in DB as being an authorized entity of the system. The database DB stores, at least, the identifier $\mathsf{ID}_{\mathcal{T}}$ and potentially a secret $\mathsf{k}_{\mathcal{T}}$ of each legitimate tag $\mathcal{T}$ involved in the system.

3.1.2 Additional Entities

Other entities may play a role in the presented privacy models. An *adversary* $\mathcal{A}$ is a malicious entity whose aim is to perform attacks, either through the wireless communications between readers and tags (e.g., eavesdropping), or on the RFID devices themselves (e.g., corrupting a device and obtaining all the information stored on it). The adversary advantage is the success measure of an attack performed by $\mathcal{A}$. In some models, $\mathcal{A}$ is requested to answer to a kind of riddle, which is determined by an honest entity, called *challenger* $\mathcal{C}$. A *challenge tag* is a tag which is suffering from an attack performed by $\mathcal{A}$. It can be chosen either by $\mathcal{A}$ or by $\mathcal{C}$.

Generally, a modelization with oracles is used to represent the possible interactions between $\mathcal{A}$ and the system $\mathcal{S}$. Thus, $\mathcal{A}$ carries out her attack on the system, performing some queries to the oracles that simulate the system. The generic oracles used in the presented privacy models are detailed in Section 3.1.4.

It is generally considered that $\mathcal{A}$ is able to play/interact with a tag when the latter is in $\mathcal{A}$'s neighborhood. At that moment, the tag is called by its pseudonym $\mathcal{T}$ (not by its identifier $\mathsf{ID}_{\mathcal{T}}$). During an attack, if a tag goes out and comes back to $\mathcal{A}$'s neighborhood, then it is considered that its pseudonym has changed. This notion is detailed in the Vaudenay model [162] (see Section 3.4). The same case happens when a set of tags is given to the challenger $\mathcal{C}$: when $\mathcal{C}$ gives the tags back to $\mathcal{A}$, their pseudonyms are changed.

3.1.3 Procedures

Most of the models studied in this chapter focus on an RFID system $\mathcal{S}$ based on an anonymous identification protocol implying a single reader and several tags. The system is generally composed of several procedures, either defining how to set up the system, the reader and the tags, or defining the studied protocol. One way to define these procedures is detailed below. Note that this is just a generalization but it may be different in some models.

- SetupReader(1^λ) defines the reader parameters (e.g., generating a pair of public/private keys $(\mathsf{P}_{\mathcal{R}}, \mathsf{K}_{\mathcal{R}})$) depending on the security

parameter λ. It also creates an empty database DB which will later contain, at least, the identifiers and secrets of all tags.

- $\textsc{SetupTag}_{\mathsf{P}_\mathcal{R}}(\mathsf{ID}_\mathcal{T})$ returns $\mathsf{k}_\mathcal{T}$, i.e., the secret $\mathsf{k}_\mathcal{T}$ of the tag $\mathcal{T}$ with identifier $\mathsf{ID}_\mathcal{T}$. $(\mathsf{ID}_\mathcal{T}, \mathsf{k}_\mathcal{T})$ is stored in the database DB of the reader.
- $\textsc{Ident}$ is a polynomial-time interactive protocol between the reader $\mathcal{R}$ and a tag $\mathcal{T}$, where $\mathcal{R}$ ends with a private tape Output. At the end of the protocol, $\mathcal{R}$ either accepts the tag (if legitimate) and $\mathsf{Output} = \mathsf{ID}_\mathcal{T}$, or rejects it (if not) and $\mathsf{Output} = \perp$.

3.1.4 Generic Oracles

An adversary $\mathcal{A}$ is able to interact/play with the system with the following oracles. First, she can setup a new tag of identifier $\mathsf{ID}_\mathcal{T}$.

- $\mathcal{O}^{\textsc{CreateTag}}(\mathsf{ID}_\mathcal{T})$: creates a tag $\mathcal{T}$ with a unique identifier $\mathsf{ID}_\mathcal{T}$. It uses $\textsc{SetupTag}_{\mathsf{P}_\mathcal{R}}$ to initialize the tag and add it to DB.

$\mathcal{A}$ can ask for a full execution of the $\textsc{Ident}$ protocol on a tag $\mathcal{T}$.

- $\mathcal{O}^{\textsc{Execute}}(\mathcal{T}) \rightarrow (\pi, \mathsf{transcript})$: executes an $\textsc{Ident}$ protocol between $\mathcal{R}$ and $\mathcal{T}$, denoted π. It outputs the $\mathsf{transcript}$ of the execution π, i.e., its whole list of the successive messages.

She can decompose a protocol execution, combining the following oracles.

- $\mathcal{O}^{\textsc{Launch}}() \rightarrow \pi$: makes $\mathcal{R}$ start a new $\textsc{Ident}$ protocol execution π.
- $\mathcal{O}^{\textsc{SendReader}}(m, \pi) \rightarrow r$: sends a message m to $\mathcal{R}$ in the protocol execution π. It outputs the response r of the reader.
- $\mathcal{O}^{\textsc{SendTag}}(m, \mathcal{T}) \rightarrow r$: sends a message m to $\mathcal{T}$. It outputs the response r of the tag.

Then, $\mathcal{A}$ can obtain the reader result of a protocol execution π.

- $\mathcal{O}^{\textsc{Result}}(\pi) \rightarrow x$: when π is completed, it outputs $x = 1$ if $\mathsf{Output} \neq \perp$, and $x = 0$ otherwise.

Finally, she can corrupt a tag $\mathcal{T}$ in order to retrieve its secret.

- $\mathcal{O}^{\text{Corrupt}}(\mathcal{T}) \rightarrow \mathsf{k}_{\mathcal{T}}$: outputs the current secret $\mathsf{k}_{\mathcal{T}}$ of $\mathcal{T}$.

If the conditions of the oracle uses are not respected, then the oracles return $\perp$. Note that these definitions are generic ones. Some models do not exactly use the same generic oracles: in those cases, some refinements will be provided on their definitions.

3.2 Avoine [9], 2005

In 2005, Avoine proposed the first privacy model for RFID systems. The goal was to analyze the untraceability notion of 3-pass protocols[1] following the idea of communication intervals: the adversary $\mathcal{A}$ asks some oracle queries on specific intervals of the targeted tags lives. The privacy notion behind this model represents the unfeasibility to distinguish one tag among two.

3.2.1 Oracles

This model considers that each tag has a unique and independent secret, and that DB already stores all the tag secrets at the initialization of the system, i.e., a SetupTag has already been performed on every tag.

Then $\mathcal{A}$ has only access to the following modified generic oracles adapted for 3-pass protocols. Instead of using the names of the entities, Avoine uses the protocol executions names. Since $\mathcal{T}$ and $\mathcal{R}$ can run several protocol executions, $\pi^i_{\mathcal{T}}$ (resp. $\pi^j_{\mathcal{R}}$) denotes the i^{th} (resp. j^{th}) execution of $\mathcal{T}$ (resp. $\mathcal{R}$). These notations favor the precise description of $\mathcal{R}$'s and $\mathcal{T}$'s lifetimes.

- $\mathcal{O}^{\text{SendTag}}(m_1, m_3, \pi^i_{\mathcal{T}}) \rightarrow r$: sends a message m_1 to $\mathcal{T}$, and then sends a message m_3 after outputting $\mathcal{T}$'s answer r. This is done during the execution $\pi^i_{\mathcal{T}}$ of $\mathcal{T}$.

- $\mathcal{O}^{\text{SendReader}}(m_2, \pi^j_{\mathcal{R}}) \rightarrow r$: sends a message m_2 to $\mathcal{R}$ in the protocol execution $\pi^j_{\mathcal{R}}$. It outputs $\mathcal{R}$'s answer r.

[1] A i-pass RFID protocol is a protocol where i messages are exchanged between a reader and a tag.

- $\mathcal{O}^{\text{Execute}}(\pi^i_{\mathcal{T}}, \pi^j_{\mathcal{R}}) \rightarrow$ transcript: executes a whole protocol between $\mathcal{T}$ and $\mathcal{R}$. This is done during the execution $\pi^i_{\mathcal{T}}$ of $\mathcal{T}$ and the execution $\pi^j_{\mathcal{R}}$ of $\mathcal{R}$. It outputs the whole transcript.
- $\mathcal{O}^{\text{Execute}^*}(\pi^i_{\mathcal{T}}, \pi^j_{\mathcal{R}}) \rightarrow \mathcal{R}$-transcript: this is the same as the normal $\mathcal{O}^{\text{Execute}}$. But it only outputs the $\mathcal{R}$-transcript, i.e., the messages sent by $\mathcal{R}$.
- $\mathcal{O}^{\text{Corrupt}}(\pi^i_{\mathcal{T}}) \rightarrow \mathsf{k}_{\mathcal{T}}$: outputs the current secret $\mathsf{k}_{\mathcal{T}}$ of $\mathcal{T}$ when the tag is in its i^{th} execution.

The goal of the $\mathcal{O}^{\text{Execute}^*}$ oracle is to simulate the fact that the forward channel (from reader to tag) has a longer communication range than the backward channel (from tag to reader), and therefore can be easily eavesdropped. It formalizes the asymmetry regarding the channels.

Two remarks are of interest for the $\mathcal{O}^{\text{Corrupt}}$ oracle. First, $\mathcal{O}^{\text{Corrupt}}$ can be used only once by $\mathcal{A}$. After this oracle query, $\mathcal{A}$ cannot use the other oracles anymore. Second, $\mathcal{O}^{\text{Corrupt}}$ is queried on the tag execution number, and not the tag itself. This allows $\mathcal{A}$ to exactly specify the targeted moment of the tag life.

During her attack, $\mathcal{A}$ has access to the oracles $\mathcal{O} \subset \{\text{T, R, E, E}^*, \text{C}\} = \{\mathcal{O}^{\text{SendTag}}, \mathcal{O}^{\text{SendReader}}, \mathcal{O}^{\text{Execute}}, \mathcal{O}^{\text{Execute}^*}, \mathcal{O}^{\text{Corrupt}}\}$.

Avoine denotes $\omega_i(\mathcal{T})$ as being the result of any oracle query on $\mathcal{T}$. He further designates an *interaction* $\Omega_I(\mathcal{T})$ as being a set of executions on the same tag $\mathcal{T}$ during an interval I when $\mathcal{A}$ can play with $\mathcal{T}$. Formally, $\Omega_I(\mathcal{T}) = \{\omega_i(\mathcal{T}) |\ i \in I\} \cup \{\mathcal{O}^{\text{SendReader}}(*, \pi^j_*) |\ j \in J\}$, where $I, J \subset \mathbb{N}$. By this definition, the length of $\Omega_I(\mathcal{T})$ is $|I|$.

Avoine also defines a function *Oracle* which takes as parameters a tag $\mathcal{T}$, an interval I and the oracles $\mathcal{O}$, and which outputs the interaction $\hat{\Omega}_I(\mathcal{T})$ that maximizes $\mathcal{A}$'s advantage.

3.2.2 Untraceability Experiments

Avoine defines two experiments to represent two untraceability (UNT) notions. They depend on λ_{ref} and λ_{chal}, which represent respectively a reference length and a challenge length, and which are function of the security parameter λ.

The first experiment detailed in Figure 3.1 works as follows. First, $\mathcal{A}$ receives the interactions of a tag $\mathcal{T}$ during an interval I that she chooses.

Then, she receives the interactions of the challenge tags $\mathcal{T}_0$ and $\mathcal{T}_1$, also during the intervals I_0 and I_1 that she chooses, such that $\mathcal{T} = \mathcal{T}_0$ or $\mathcal{T}_1$. This last information is unknown to $\mathcal{A}$. Additionally here, none of these two intervals I_0 and I_1 crosses the interval I of $\mathcal{T}$. At the end, $\mathcal{A}$ has to decide which one of the challenge tags is the tag $\mathcal{T}$.

Experiment $Exp^{\mathsf{Existential\text{-}UNT}}_{\mathcal{S},\mathcal{A}}[\lambda_{\text{ref}}, \lambda_{\text{chal}}, \mathcal{O}]$

1. $\mathcal{C}$ initializes the system $\mathcal{S}$.
2. $\mathcal{A}$ requests $\mathcal{C}$ to receive a tag $\mathcal{T}$.
3. $\mathcal{A}$ chooses I, queries $Oracle(\mathcal{T}, I, \mathcal{O})$ where $|I| \leq \lambda_{\text{ref}}$, and then receives $\hat{\Omega}_I(\mathcal{T})$.
4. $\mathcal{A}$ requests $\mathcal{C}$ to receive two challenge tags $\mathcal{T}_0$ and $\mathcal{T}_1$, such that $\mathcal{T} = \mathcal{T}_0$ or $\mathcal{T}_1$.
5. $\mathcal{A}$ chooses I_0 and I_1 such that $|I_0| \leq \lambda_{\text{chal}}$, $|I_1| \leq \lambda_{\text{chal}}$, and $(I_0 \cup I_1) \cap I = \emptyset$
6. $\mathcal{A}$ queries $Oracle(\mathcal{T}_0, I_0, \mathcal{O})$ and $Oracle(\mathcal{T}_1, I_1, \mathcal{O})$, and then receives $\hat{\Omega}_{I_0}(\mathcal{T}_0)$ and $\hat{\Omega}_{I_1}(\mathcal{T}_1)$.
7. $\mathcal{A}$ decides which of $\mathcal{T}_0$ or $\mathcal{T}_1$ is $\mathcal{T}$, and outputs a guess bit b.

$Exp^{\mathsf{Existential\text{-}UNT}}_{\mathcal{S},\mathcal{A}}$ succeeds if $\mathcal{T} = \mathcal{T}_b$.

Figure 3.1: Existential untraceability experiment of the Avoine model.

The second experiment detailed in Figure 3.2 follows the same reasoning. The only difference is that, now, $\mathcal{C}$ is the one who chooses the intervals I_0 and I_1 of the challenge tags, and not $\mathcal{A}$ anymore.

3.2.3 Untraceability Notions

From the experiments defined above, the notions of Existential-UNT and Universal-UNT are extended in this model, depending on restrictions about the choices of I_0 and I_1. Existential-UNT is when $\mathcal{A}$ chooses I_0 and I_1, whereas Universal-UNT is when $\mathcal{C}$ chooses them. Then, if $I < I_0, I_1$

Experiment $Exp^{\mathsf{Universal\text{-}UNT}}_{\mathcal{S},\mathcal{A}}[\lambda_{\text{ref}}, \lambda_{\text{chal}}, \mathcal{O}]$

1. $\mathcal{C}$ initializes the system $\mathcal{S}$.
2. $\mathcal{A}$ requests $\mathcal{C}$ to receive a tag $\mathcal{T}$.
3. $\mathcal{A}$ chooses I, queries $Oracle(\mathcal{T}, I, \mathcal{O})$ where $|I| \leq \lambda_{\text{ref}}$, and then receives $\hat{\Omega}_I(\mathcal{T})$. Here I is known by $\mathcal{C}$.
4. $\mathcal{A}$ requests $\mathcal{C}$ to receive two challenges $\mathcal{T}_0$, $\mathcal{T}_1$, I_0 and I_1, such that $\mathcal{T} = \mathcal{T}_0$ or $\mathcal{T}_1$.
5. $\mathcal{A}$ queries $Oracle(\mathcal{T}_0, I_0, \mathcal{O})$ and $Oracle(\mathcal{T}_1, I_1, \mathcal{O})$, and then receives $\hat{\Omega}_{I_0}(\mathcal{T}_0)$ and $\hat{\Omega}_{I_1}(\mathcal{T}_1)$.
6. $\mathcal{A}$ decides which of $\mathcal{T}_0$ or $\mathcal{T}_1$ is $\mathcal{T}$, and outputs a guess bit b.

$Exp^{\mathsf{Universal\text{-}UNT}}_{\mathcal{S},\mathcal{A}}$ succeeds if $\mathcal{T} = \mathcal{T}_b$.

Figure 3.2: Universal untraceability experiment of the Avoine model.

(resp. $I > I_0, I_1$), that means I_0 and I_1 take place after (resp. before) I, with respect to the lifetime of the system.

- If $\mathcal{A}$ (resp. $\mathcal{C}$) chooses I_0 and I_1 such that $I < I_0, I_1$, then it is denoted $\mathsf{Existential}^+$ (resp. $\mathsf{Universal}^+$).
- If $\mathcal{A}$ (resp. $\mathcal{C}$) chooses I_0 and I_1 such that $I > I_0, I_1$, then it is denoted $\mathsf{Existential}^-$ (resp. $\mathsf{Universal}^-$).

The notion of $\mathsf{Universal}^-$ when the $\mathcal{O}^{\text{Corrupt}}$ oracle is used is called Forward-UNT.

Definition 3.1 (Untraceability [9]). *An RFID system $\mathcal{S}$ is said P-UNT-$\mathcal{O}$ (for $P \in$ {Existential, Forward, Universal}) if, for every adversary $\mathcal{A}$:*

$$\left|\Pr(Exp^{P\text{-}UNT}_{\mathcal{S},\mathcal{A}}[\lambda_{\text{ref}}, \lambda_{\text{chal}}, \mathcal{O}] \text{ succeeds}) - \frac{1}{2}\right| \leq \epsilon(\lambda_{\text{ref}}, \lambda_{\text{chal}}).$$

Direct implications are made from these notions:

Existential-UNT-$\mathcal{O}$ $\Rightarrow$ Forward-UNT-$\mathcal{O}$ $\Rightarrow$ Universal-UNT-$\mathcal{O}$

3.3 Juels and Weis [99], 2007

Two years after Avoine's publication, Juels and Weis proposed a new privacy model, referred in the sequel as JW, based on indistinguishability of tags. It intended to analyze classical challenge/response protocols based on symmetric-key cryptography (with possible additional messages in order to update the tag keys).

In their article, the authors highlighted that the Avoine model lacks of two important features. Firstly, they proved that it is unable to catch an important attack on systems where tags have correlated secrets, because the Avoine adversary can only play with two tags. Secondly, they showed that Avoine did not have hindsight regarding all the possible attacks that can be performed on a protocol. His model does not capture all the relevant information that can be extracted from a protocol execution. For instance, it does not consider that $\mathcal{A}$ has access to the result of a protocol execution. However, this simple "side information bit" allows the formalization of a special kind of attack on desynchronizable protocols like the OSK-Prot one (we will explain this issue in Section 4.1.3). Therefore, the JW model aimed to fill that gap.

3.3.1 Oracles

At the initialization of the system, DB already stores the contents of all the tags, i.e., a SetupTag has already been performed on every tag. Then $\mathcal{A}$ has access to the generic oracles $\mathcal{O}^{\text{Launch}}$, $\mathcal{O}^{\text{SendTag}}$, and $\mathcal{O}^{\text{SendReader}}$, with the difference that the output of $\mathcal{O}^{\text{Launch}}$ includes a first message m answered by $\mathcal{R}$, and that the output of $\mathcal{O}^{\text{SendReader}}$ includes the output of $\mathcal{O}^{\text{Result}}$. $\mathcal{A}$ has furthermore access to the following oracles.

- $\mathcal{O}^{\text{TagInit}}(\mathcal{T}, \pi)$: makes $\mathcal{T}$ start a new protocol execution π and delete the information related to any existing execution.
- $\mathcal{O}^{\text{SetKey}}(\mathcal{T}, \mathsf{k}_{\mathcal{T}}^{\text{new}}) \rightarrow \mathsf{k}_{\mathcal{T}}$: outputs the current key $\mathsf{k}_{\mathcal{T}}$ of $\mathcal{T}$, and replaces it in $\mathcal{T}$ by the new one $\mathsf{k}_{\mathcal{T}}^{\text{new}}$.

$\mathcal{O}^{\text{SetKey}}$ is equivalent to the $\mathcal{O}^{\text{Corrupt}}$ oracle given in Section 3.1.4 in the sense that it reveals the tag current key to $\mathcal{A}$. Note that its use and its result have an interesting feature: $\mathcal{A}$ is able to put any new key

in the targeted tag, either the revealed one or a random one (that can be illegitimate).

3.3.2 Privacy Experiment

Let ρ, σ and τ be respectively the numbers of $\mathcal{O}^{\text{Launch}}$, computation steps (represented by the $\mathcal{O}^{\text{SendReader}}$ and $\mathcal{O}^{\text{SendTag}}$ queries) and $\mathcal{O}^{\text{TagInit}}$ that are allowed to $\mathcal{A}$. Let n be the total number of tags involved in the system $\mathcal{S}$. The privacy experiment is detailed in Figure 3.3.

3.3.3 Privacy Notions

From the previous experiment, the JW model defines the following privacy property, where ρ, σ and τ can be function of the system security parameter λ.

Definition 3.2 ((ρ, σ, τ)-Privacy [99]). *A protocol initiated by $\mathcal{R}$ in an RFID system $\mathcal{S}$ with security parameter λ is* (ρ, σ, τ)-private *if, for every adversary $\mathcal{A}$:*

$$\left|\Pr(Exp^{\textsf{JW-priv}}_{\mathcal{S},\mathcal{A}}[\lambda, n, \rho, \sigma, \tau] \text{ succeeds}) - \frac{1}{2}\right| \leq \epsilon(\lambda).$$

Considering a variant of experiment $Exp^{\textsf{JW-priv}}_{\mathcal{S},\mathcal{A}}$ where the "*except* $\mathcal{T}_b^*$" is removed from step (6.b), then forward-(ρ, σ, τ)-privacy can be defined in the same way as the previous definition.

Note that, if $\mathcal{A}$ uses $\mathcal{O}^{\text{SetKey}}$ to put an illegitimate key in a tag, then this tag will possibly no longer be authenticated successfully by the reader. Nevertheless, whether this is performed on the non-challenge tags or on $\mathcal{T}_b^*$ (only for the forward-(ρ, σ, τ)-privacy experiment), this does not help $\mathcal{A}$ to find more easily the bit b, and thus does not influence her success to win the experiment.

3.4 Vaudenay [162], 2007

Later the same year, Vaudenay proposed formal definitions for RFID systems and adversaries, and considered that a system $\mathcal{S}$ can be characterized by two notions: security and privacy. In this chapter, we only present the privacy notion. Vaudenay's article followed some joint work

Experiment $Exp^{\mathsf{JW\text{-}priv}}_{\mathcal{S},\mathcal{A}}[\lambda, n, \rho, \sigma, \tau]$

Setup:

1. $\mathcal{C}$ initializes the system $\mathcal{S}$.

Phase 1 (Learning):

2. $\mathcal{A}$ may do the following in any interleaved order:
   a. Make $\mathcal{O}^{\textsc{Launch}}$ and $\mathcal{O}^{\textsc{TagInit}}$ queries, without exceeding ρ and τ overall queries respectively.
   b. Make arbitrary $\mathcal{O}^{\textsc{SetKey}}$ queries to any $(n-2)$ tags.
   c. Make $\mathcal{O}^{\textsc{SendReader}}$ and $\mathcal{O}^{\textsc{SendTag}}$ queries, without exceeding σ overall queries.

Phase 2 (Challenge):

3. $\mathcal{A}$ selects two challenge tags $\mathcal{T}_i$ and $\mathcal{T}_j$ to which she did not send $\mathcal{O}^{\textsc{SetKey}}$ queries.
4. Let $\mathcal{T}_0^* = \mathcal{T}_i$ and $\mathcal{T}_1^* = \mathcal{T}_j$, and remove both from the current tag set.
5. $\mathcal{C}$ chooses a bit b at random, and provides $\mathcal{A}$ access to $\mathcal{T}_b^*$.
6. $\mathcal{A}$ may do the following in any interleaved order:
   a. Make $\mathcal{O}^{\textsc{Launch}}$ and $\mathcal{O}^{\textsc{TagInit}}$ queries, without exceeding ρ and τ overall queries respectively.
   b. Make arbitrary $\mathcal{O}^{\textsc{SetKey}}$ queries to any tag in the current tag set, *except* $\mathcal{T}_b^*$.
   c. Make $\mathcal{O}^{\textsc{SendReader}}$ and $\mathcal{O}^{\textsc{SendTag}}$ queries, without exceeding σ overall queries.
7. $\mathcal{A}$ outputs a guess bit b'.

$Exp^{\mathsf{JW\text{-}priv}}_{\mathcal{S},\mathcal{A}}$ succeeds if $b = b'$.

Figure 3.3: Privacy experiment of the JW model.

done with Bocchetti [24], and its goal was to propose a comprehensive model that can formalize a wide range of adversaries. This characteristic is missing in the previous models, and turns to be an asset of the Vaudenay model.

This model defines tags with respect to the adversary possibility to interact with them, as explained in Section 3.1.2. Clearly, when a tag is within $\mathcal{A}$'s neighborhood, it is said drawn and has a pseudonym so that $\mathcal{A}$ is able to communicate with the tag. In the opposite situation, a tag is said free (i.e., not drawn), and $\mathcal{A}$ cannot communicate with it. Consequently, the model considers that, at any given time, a tag can be either free or drawn. For example, the same tag with identifier $\mathsf{ID}_{\mathcal{T}}$ which is drawn, freed and drawn again has two pseudonyms: $\mathcal{A}$ sees two different tags. Additionally, all the tags may not be accessible to $\mathcal{A}$ during all the attack, e.g., $\mathcal{A}$ may only play with two (drawn) tags.

3.4.1 Oracles

Contrary to the other previous models, DB is empty at the initialization of the system. Then $\mathcal{A}$ has access to all the generic oracles defined in Section 3.1.4. The only modification done on these ones is that $\mathcal{A}$ can create a fake tag with $\mathcal{O}^{\text{CreateTag}}$. In that case, no information related to this tag is stored in DB. She can also query the following oracles.

- $\mathcal{O}^{\text{DrawTag}}(\mathsf{distr}) \rightarrow (\mathcal{T}_1, b_1, \ldots, \mathcal{T}_k, b_k)$: following the distribution probability distr (which is specified by a polynomially bounded sampling algorithm), it randomly selects k tags between all the existing (not already drawn) ones. For each chosen tag, the oracle assigns a new pseudonym denoted $\mathcal{T}_i$ to it and changes its status from free to drawn. Finally, the oracle outputs all the generated temporary tags $(\mathcal{T}_1, \cdots, \mathcal{T}_k)$ in any random order. If there is not enough free tags (i.e., less than k), or tags already drawn, then the oracle outputs $\perp$. It is further assumed that this oracle returns bits $(b_1, \ldots, b_k)$ telling if each of the drawn tags are legitimate or not. All relations $(\mathcal{T}_i, \mathsf{ID}_{\mathcal{T}_i})$ are kept in an *a priori* secret table denoted Tab.

- $\mathcal{O}^{\text{Free}}(\mathcal{T})$: moves the tag $\mathcal{T}$ from the status drawn to the status free. $\mathcal{T}$ is unavailable from now on.

3.4.2 Privacy Experiment

From the oracles given above, Vaudenay defines five classes of polynomial-time adversary, characterized by $\mathcal{A}$'s ability to use the oracles.

Definition 3.3 (Adversary Class [162]). *An adversary class is said:*

- STRONG *if $\mathcal{A}$ has access to all the oracles;*
- DESTRUCTIVE *if $\mathcal{A}$ cannot use anymore a "corrupted" tag (i.e., the tag has been destroyed);*
- FORWARD *if $\mathcal{A}$ can only use the $\mathcal{O}^{\text{Corrupt}}$ oracle after her first query to the $\mathcal{O}^{\text{Corrupt}}$ oracle;*
- WEAK *if $\mathcal{A}$ has no access to the $\mathcal{O}^{\text{Corrupt}}$ oracle;*
- NARROW *if $\mathcal{A}$ has no access to the $\mathcal{O}^{\text{Result}}$ oracle.*

Remark. The following relation is clearly verified:

$$\mathsf{WEAK} \subseteq \mathsf{FORWARD} \subseteq \mathsf{DESTRUCTIVE} \subseteq \mathsf{STRONG}.$$

Note that the WIDE notion is the contrary to the NARROW one. If an adversary $\mathcal{A}$ is not said NARROW, then nothing is said, but the term WIDE is implicitly meant.

Experiment $Exp^{\mathsf{Vaud\text{-}priv}}_{\mathcal{S},\mathcal{A}}[\lambda]$

1. $\mathcal{C}$ initializes the system $\mathcal{S}$ and sends 1^λ, and $\mathsf{P}_\mathcal{R}$ to $\mathcal{A}$.
2. $\mathcal{A}$ interacts with the whole system, limited by her class P.
3. $\mathcal{A}$ analyzes the system without oracle queries.
4. $\mathcal{A}$ receives the hidden table Tab of the $\mathcal{O}^{\text{DrawTag}}$ oracle.
5. $\mathcal{A}$ returns *true* or *false*.

$Exp^{\mathsf{Vaud\text{-}priv}}_{\mathcal{S},\mathcal{A}}$ succeeds if $\mathcal{A}$ returns *true*.

Figure 3.4: Privacy experiment of the Vaudenay model.

The Vaudenay privacy experiment is detailed in Figure 3.4, where P is the adversary class such that:

$$P \in \{\emptyset, \mathsf{NARROW}\} \times \{\mathsf{WEAK}, \mathsf{FORWARD}, \mathsf{DESTRUCTIVE}, \mathsf{STRONG}\}.$$

3.4.3 Privacy Notions

To define the privacy property of Vaudenay, it is first needed to define the notions of *blinder* (i.e., an algorithm able to simulate the answers of some specific oracles) and *trivial adversary* (i.e., an adversary who learns nothing about the system).

Definition 3.4 (Blinder, Trivial Adversary [162]). *A* blinder $\mathcal{B}$ *for an adversary* $\mathcal{A}$ *is a polynomial-time algorithm which sees the same messages as* $\mathcal{A}$ *and simulates the* $\mathcal{O}^{\textsc{Launch}}$, $\mathcal{O}^{\textsc{SendReader}}$, $\mathcal{O}^{\textsc{SendTag}}$ *and* $\mathcal{O}^{\textsc{Result}}$ *oracles to* $\mathcal{A}$. $\mathcal{B}$ *does not have access to the reader tapes, so does not know the secret key nor the database.*

A blinded adversary $\mathcal{A}^{\mathcal{B}}$ *is itself an adversary who does not use the* $\mathcal{O}^{\textsc{Launch}}$, $\mathcal{O}^{\textsc{SendReader}}$, $\mathcal{O}^{\textsc{SendTag}}$ *and* $\mathcal{O}^{\textsc{Result}}$ *oracles.*

An adversary $\mathcal{A}$ *is* trivial *if there exists a blinder* $\mathcal{B}$ *such that:*

$$|\Pr(Exp^{\mathsf{Vaud\text{-}priv}}_{\mathcal{S},\mathcal{A}}[\lambda] \text{ succeeds}) - \Pr(Exp^{\mathsf{Vaud\text{-}priv}}_{\mathcal{S},\mathcal{A}^{\mathcal{B}}}[\lambda] \text{ succeeds})| \leq \epsilon(\lambda).$$

Definition 3.5 (Privacy [162]). *The RFID system* $\mathcal{S}$ *is said* P-private *if all the adversaries which belong to class* P *are trivial following Definition 3.4.*

The implications between the Vaudenay privacy notions are:

STRONG	⇒	DESTRUCTIVE	⇒	FORWARD	⇒	WEAK
⇓		⇓		⇓		⇓
NARROW-STRONG	⇒	NARROW-DESTRUCTIVE	⇒	NARROW-FORWARD	⇒	NARROW-WEAK

The main result of Vaudenay is that STRONG-privacy is impossible, by proving that a DESTRUCTIVE-private protocol is not NARROW-STRONG-private. However, Vaudenay does not define which privacy level should be targeted by a protocol: it is never specified if NARROW-STRONG-privacy is better or not than DESTRUCTIVE-privacy.

Also, it is not explicit how the blinded adversary $\mathcal{A}^{\mathcal{B}}$ operates. Basically, there are two options: (i) $\mathcal{A}^{\mathcal{B}}$ aims the same probability as $\mathcal{A}$, or (ii) $\mathcal{A}^{\mathcal{B}}$ aims the same behavior as $\mathcal{A}$. It is obvious that the first option allows proving the privacy of some protocols which are actually not private, but this should be correctly formalized.

3.4.4 Extensions of the Model

Model [139], 2008. Paise and Vaudenay extended the Vaudenay model to analyze mutual authentication protocols. Actually, they enriched the definition of the RFID system $\mathcal{S}$ by introducing an output on the tag side: either the tag accepts the reader (if legitimate) and outputs OK, or rejects it (if not) and outputs $\perp$. This formalizes the concept of *reader authentication*. Nevertheless, their extension does not modify the core of the Vaudenay model.

They also showed an important impossibility result: if the corruption of a tag reveals its entire state (and not only its secret $\mathsf{k}_{\mathcal{T}}$), then no RFID scheme providing reader authentication is NARROW-FORWARD-private. To counter this issue, they claimed that the temporary memory of a tag should be automatically erased as soon as the tag is put back as free. However, this idea is not formalized in the paper.

This division between the persistent and the temporary memory of a tag has also been investigated by Armknecht, Sadeghi, Scafuro, Visconti, and Wachsmann [5]. Based on the work of Paise and Vaudenay, they showed several impossibility results in attack scenarios with special uses of tag corruption.

Model [136], 2011. Ouafi presented in his thesis an adaptation of the Vaudenay model in order to counter the Vaudenay impossibility result of STRONG-privacy. Concretely, the author proposed to incorporate the blinder with the adversary, so that the blinder has the knowledge of all the random choices and incoming messages made by the adversary. With this new definition of the blinder, Ouafi proved that STRONG-privacy can be ensured. This result is demonstrated with a public-key-based authentication protocol where the cryptosystem is IND-CCA2 secure and PA1+ plaintext-aware[2].

Other Extensions. The Vaudenay model has also been broadened in different works. In a nutshell, this is generally performed via the addition of a new oracle to the adversary capabilities (e.g., $\mathcal{O}^{\textsc{Timer}}$ that we will detail in Chapter 5, $\mathcal{O}^{\textsc{MakeInactive}}$ in [53], or $\mathcal{O}^{\textsc{DestroyReader}}$

[2]More details about these security notions can be found in [23].

in [68]), and the corresponding new adversary class (e.g., the TIMEFUL class when $\mathcal{A}$ is allowed to use $\mathcal{O}^{\text{TIMER}}$).

3.5 Le, Burmester and de Medeiros [31, 107], 2007

Also in 2007, Le, Burmester and de Medeiros introduced a privacy model in [107] (and an extended version in [31]) that is derived from the universal composability (UC) framework [36, 37][3]. Their aim was to provide security proofs of protocols under concurrent and modular composition, such that protocols can be easily incorporate in more complex systems without reanalyses. The model, denoted LBM in the following, is based on the indistinguishability between two worlds: the real world and the ideal one.

The transposition of RFID privacy into such a framework is a great contribution since universal composability is considered as one of the most powerful tools for security, especially when composition among several functionalities is required.

3.5.1 General Statements About the UC Framework

The Environment $\mathcal{Z}$. In the UC framework, $\mathcal{Z}$'s purpose is to manage the evolution of the system $\mathcal{S}$. In other words, this entity is in charge of the activation of all the parties, including the adversary $\mathcal{A}$. $\mathcal{Z}$ is the only entity able to request a party $\mathcal{P}$ to initiate a new execution of the studied IDENT protocol. It is also able to read the output tapes of the system parties and of $\mathcal{A}$. On the other hand, $\mathcal{Z}$ is not assumed to read the incoming/outgoing messages of the parties during a protocol execution.

While this new entity is quite unusual compared to the other privacy models, it permits formalizing systems where there is an underlying communication structure which may be unknown to the adversary. In the other models, $\mathcal{A}$ is in charge of the activation of the parties. As a consequence, if there exists an underlying activation sequence that is unknown to the adversary, she cannot respect it and thus may loose information that would help her to perform her attack. The potential activation scheduling performed by $\mathcal{Z}$ thus strengthens the power of the adversary.

[3] And not on the oracle-based framework.

The Real World. The system $\mathcal{S}$ is composed of several *honest parties* that interact together through an IDENT protocol in order to achieve a well-defined objective.

An adversary $\mathcal{A}$ is in charge of the communication channels: she can eavesdrop, modify and schedule all the channels between the honest parties in an arbitrary way. $\mathcal{A}$ may also be able to corrupt parties and obtain the full knowledge of their state. Corrupted parties are assumed to be totally controlled by $\mathcal{A}$ afterwards.

$\mathcal{Z}$ and $\mathcal{A}$ can discuss in an arbitrary way. Consequently, if $\mathcal{A}$ wants to, she can forward all the communications to $\mathcal{Z}$. She can also ask $\mathcal{Z}$ to launch new executions of IDENT. At the end of the experiment, $\mathcal{A}$ may send her final output to $\mathcal{Z}$ which is the last activated entity of the system. Then, $\mathcal{Z}$ outputs an arbitrary string, denoted $\text{EXEC}_{\text{IDENT},\mathcal{A},\mathcal{Z}}$, which can be reduced to one bit as proved by Canetti in [36, 37].

The Ideal World. Here, all the parties have access to the *ideal functionality* $\mathcal{F}$, that is a trusted and uncorrupted party. $\mathcal{F}$ must trivially ensure the desired security objectives of the IDENT protocol, and does not depend on any cryptographic mechanism.

Equivalently to the adversary $\mathcal{A}$ in the real world, a simulated adversary *Sim* is defined such that *Sim* can arbitrarily discuss with $\mathcal{Z}$. However, *Sim* can no longer directly interact with parties: it can only communicate with $\mathcal{F}$ which manages all the entities communications. The main goal of *Sim* is to reproduce the behavior of $\mathcal{A}$ in the real world as faithfully as possible. Since (i) $\mathcal{A}$ may transfer messages of the IDENT protocol to $\mathcal{Z}$, (ii) *Sim* does not have access to IDENT, and (iii) $\mathcal{F}$ does not produce such messages, then *Sim* should simulate these messages to $\mathcal{Z}$. The final output of $\mathcal{Z}$ is denoted $\text{EXEC}_{\Phi,Sim,\mathcal{Z}}$, where the protocol Φ UC-realizes the ideal functionality $\mathcal{F}$ (as defined in [36]).

3.5.2 UC Security

To prove that an IDENT protocol is as secure as the corresponding ideal functionality $\mathcal{F}$, no environment $\mathcal{Z}$ should distinguish if it is interacting with the real adversary $\mathcal{A}$ and IDENT (i.e., the real world), or with the simulated adversary *Sim* and $\mathcal{F}$ (i.e., the ideal world). Consequently, $\mathcal{F}$ must be well-defined such that all the targeted security properties are

trivially ensured. Canetti formally defines this concept in [36] as follows, where PPT denotes a probabilistic polynomial time Turing machine.

Definition 3.6 (UC-Emulation [36]). *A protocol* IDENT UC-emulates *a protocol* Φ *if for all PPT adversary* $\mathcal{A}$*, there exists a PPT simulated adversary Sim such that, for all PPT environment* $\mathcal{Z}$*, the distributions* $EXEC_{IDENT,\mathcal{A},\mathcal{Z}}$ *and* $EXEC_{\Phi,Sim,\mathcal{Z}}$ *are indistinguishable.*

Based on this security framework, Le, Burmester and de Medeiros designed in [31, 107] several ideal functionalities to formalize anonymous authentication as well as anonymous authenticated key exchange.

3.5.3 Description of the LBM Model

The advantage of using this UC-based model is that all the possible adversaries and environments are considered during the security proof that can be carried out with LBM. In this thesis, we only focus on the forward-security objective led by anonymous authentication.

Assumptions of an RFID System $\mathcal{S}$. First, the LBM model establishes that the reader $\mathcal{R}$ is the only entity that can start a protocol execution. Then, it considers that only tags can be corrupted by an adversary $\mathcal{A}$. Upon corruption of a tag, $\mathcal{A}$ obtains its keys and all its persistent memory values.

The LBM Ideal Functionality $\mathcal{F}_{aauth}$. This ideal functionality represents the *anonymous authentication* security objective of a given protocol. To do so, several parties (at least $\mathcal{R}$ and one tag) may be involved in a protocol execution. Two parties $\mathcal{P}$ and $\mathcal{P}'$ are said *feasible partners* if and only if they are respectively $\mathcal{R}$ and a tag. In the ideal world, communication channels between $\mathcal{R}$ and tags are assumed to be anonymous (meaning that they only reveal the type $type(\mathcal{P})$ of a party, either tag or reader), and a sent message is necessarily delivered to the recipient. Finally, $state(\mathcal{P})$ is the list of all the execution records, and $active(\mathcal{P})$ is the list of all the preceding incomplete executions.

Ideal Functionality $\mathcal{F}_{aauth}$

- **Upon receiving** INITIATE **from** $\mathcal{P}$: if $\mathcal{P}$ is corrupted then ignore this message. Else generate a unique execution identification sid, record $\texttt{init}(sid, \mathcal{P})$ and send $\texttt{init}(sid, type(\mathcal{P}), active(\mathcal{P}))$ to the adversary.
- **Upon receiving** ACCEPT(sid, sid') **from** the adversary: if there are two records $\texttt{init}(sid, \mathcal{P})$ and $\texttt{init}(sid', \mathcal{P}')$ where $\mathcal{P}$ and $\mathcal{P}'$ are feasible partners, then remove them, record $\texttt{partner}(sid', \mathcal{P}', sid, \mathcal{P})$ and write output $\textbf{ACCEPT}(\mathcal{P}')$ to $\mathcal{P}$. Else if there is a record $\texttt{partner}(sid, \mathcal{P}, sid', \mathcal{P}')$, then remove it and write output $\textbf{ACCEPT}(\mathcal{P}')$ to $\mathcal{P}$.
- **Upon receiving** IMPERSONATE$(sid, \mathcal{P}')$ **from** the adversary: if there is a record $\texttt{init}(sid, \mathcal{P})$ and party $\mathcal{P}'$ is corrupted, then remove this record and write output $\textbf{ACCEPT}(\mathcal{P}')$ to $\mathcal{P}$.
- **Upon receiving** CORRUPT(sid) **from** the adversary: if there is a record $\texttt{init}(sid, \mathcal{P})$ or $\texttt{partner}(sid, \mathcal{P}, sid', \mathcal{P}')$ such that $\mathcal{P}$ is corruptible, then mark $\mathcal{P}$ as corrupted and remove $state(\mathcal{P})$.

Forward-security. When the adversary corrupts a tag $\mathcal{T}$, she gets its identifier $\mathsf{ID}_{\mathcal{T}}$, and is then able to impersonate this tag using the IMPERSONATE command. A corrupted tag is thereafter considered as totally controlled by the adversary. Consequently, $\mathcal{F}_{aauth}$ will no longer manage the behavior of this corrupted tag and thus will reject every INITIATE command from this tag. As $state(\mathcal{T})$ is removed after a corruption, the adversary is not able to link the related tag to its previous authentication.

However, the adversary is able to link all the incomplete protocol executions of a corrupted tag $\mathcal{T}$ up to the last successfully completed one, based on the knowledge of $active(\mathcal{T})$. Thus, $\mathcal{F}_{aauth}$ obviously provides forward-security for all previous completed protocol executions.

3.6 Van Deursen, Mauw and Radomirovic [47], 2008

The model of van Deursen, Mauw and Radomirovic, published in 2008, defines untraceability in the standard Dolev-Yao intruder model [51]. The untraceability notion is inspired by the anonymity theory given in [67, 113], and is used as a formal verification of RFID protocols. Such a technique is based on a *symbolic protocol analysis* approach[4]. This model will be called DMR in what follows.

3.6.1 Definition of the System

We remind below the basic definitions given in DMR.

First, the system is composed of some *agents* (e.g., Alice or Bob) that execute a *security protocol* which is described by a set of *traces*. A security protocol represents the behavior of a set of *roles* (i.e., initiator, responder, server), each one specifying a set of *actions*. These actions depict the role specifications with a sequence of *events* (e.g., sending or reception of a message). A *role term* is a message contained in an event and is built from *basic role terms* (e.g., nonces, role names, keys). A *complex term* is built with functions (e.g., encryption, hashing, XOR).

Each trace t is composed of interleaved runs and run prefixes, denoted *subtraces*. A *run* of a role R is a protocol execution from R's point of view, denoted $R\#sid$, where *sid* is a (possibly unique) run identifier. Thus, a run is an instantiation of a role. A *run event* is an instantiation of a role event, i.e., an instantiation of an event role terms. A *run term* denotes an instantiated role term. A *run prefix* is an unfinished run.

An adversary $\mathcal{A}$ is defined in the Dolev-Yao model, and is characterized by her *knowledge*. This knowledge is composed of a set of run terms known at the beginning, and the set of run terms that she will observe during her attack. The adversary is allowed to manipulate the information of her knowledge to understand terms or build new ones. However, perfect cryptography is assumed (i.e., cryptographic primitives are assumed unbreakable and considered as black boxes). The inference of term a from term set K is denoted $K \vdash a$.

[4] And not on the oracle-based framework.

Corrupted agents are modeled[5]: $\mathcal{A}$ is given all the secrets of a corrupted agent in her initial knowledge. When an agent is corrupted, it is said "destroyed", i.e., it cannot be used during $\mathcal{A}$'s attack. Yet, the security evaluation of a system is done on non-corrupted agents, i.e., $\mathcal{A}$ cannot have access to the secret of an agent after she starts her attack.

3.6.2 Untraceability Notion

The model firstly defines several notions of *linkability*, *reinterpretation*, and *indistinguishability*, before giving the *untraceability* one.

Definition 3.7 (Linkability [47]). *Two subtraces* $\mathfrak{t}_i^R$ *and* $\mathfrak{t}_j^R$ *are linked, denoted* $L(\mathfrak{t}_i^R, \mathfrak{t}_j^R)$*, if they are instantiated by the same agent:*

$$L(\mathfrak{t}_i^R, \mathfrak{t}_j^R) \equiv (agent(\mathfrak{t}_i^R) = agent(\mathfrak{t}_j^R)).$$

The notion of *reinterpretation* has been introduced in [67] in order to show that subterms of a message can be replaced by other subterms if the adversary $\mathcal{A}$ is not able to understand these subterms. Note that, when $\mathcal{A}$ is able to understand a subterm, it remains unchanged.

Definition 3.8 (Reinterpretation [47]). *A map* μ *from run terms to run terms is called a reinterpretation under knowledge set* K *if it and its inverse* μ^{-1} *satisfy the following conditions:*

- $\mu(a) = a$ — *if* a *is a basic run term,*
- $\mu(a) = (\mu(a_1), \ldots, \mu(a_n))$ — *if* $a = (a_1, \ldots, a_n)$ *is n-tuple,*
- $\mu(\{a\}_k) = \{\mu(a)\}_k$ — *if* $K \vdash k^{-1}$ *or* $(K \vdash a \wedge K \vdash k)$*, and* $\{.\}_k$ *is an encryption under key* k*,*
- $\mu(f(a)) = f(\mu(a))$ — *if* $K \vdash a$ *or* f *is not a hash function.*

Reinterpretations are used to define *indistinguishability* of traces.

Definition 3.9 (Indistinguishability of Traces [47]). *Let* K *be the adversary knowledge at the end of trace* $\mathfrak{t}$*. The trace* $\mathfrak{t}$ *is* indistinguishable *from a trace* $\mathfrak{t}'$*, denoted* $\mathfrak{t} \sim \mathfrak{t}'$*, if there is a reinterpretation* μ *under* K*, such that* $\mu(\mathfrak{t}_i^R) = \mathfrak{t}'^R_i$ *for all roles* R *and subtraces* $\mathfrak{t}_i^R$*.*

[5]Note that, regarding corruption, there is no restriction about the role of such an agent: it can be either a tag or a reader.

From all the above notions, the untraceability notion of a role is defined as follows.

Definition 3.10 (Untraceability [47]). *An* IDENT *protocol is said* untraceable *with respect to role R if:*
$\big(\forall t \in \mathit{Traces}(\textsc{Ident})\big)$
$\big(\forall i \neq j\big)\Big(L(\mathfrak{t}_i^R, \mathfrak{t}_j^R) \Rightarrow \big(\exists \mathfrak{t}' \in \mathit{Traces}(\textsc{Ident})\big)\big((\mathfrak{t} \sim \mathfrak{t}') \wedge \neg L(\mathfrak{t}'^R_i, \mathfrak{t}'^R_j)\big)\Big)$.

In this thesis, if no role is specified, we consider that "untraceability" means "untraceability for role $\mathcal{T}$".

3.7 Canard, Coisel, Etrog and Girault [34, 35], 2010

In the same vein as the Vaudenay model, Canard, Coisel, Etrog and Girault proposed in 2010 a security model that comprises the properties of (strong) correctness, soundness and untraceability. We only present the last notion. Contrary to Vaudenay, the authors only defined untraceability (and not privacy in general) and their main goal was to use the strongest adversary of the Vaudenay model. During the following, this model will be denoted CCEG.

3.7.1 Oracles

As for Vaudenay, DB is empty after the setup of the system, and a tag can be either free or drawn. Then $\mathcal{A}$ has access to all the generic oracles. She may also use the following ones.

- $\mathcal{O}^{\textsc{DrawTag}}(k) \rightarrow (\mathcal{T}_1, \cdots, \mathcal{T}_k)$: works similarly as the one of Vaudenay. It first randomly and uniformly selects k tags between all existing (not already drawn) ones. For each chosen tag, the oracle gives a new pseudonym denoted $\mathcal{T}_i$ to it and changes its status from free to drawn. Finally, since $\mathcal{A}$ cannot create fake tags, then the oracle only outputs all the generated pseudonyms $(\mathcal{T}_1, \cdots, \mathcal{T}_k)$ in any order. If there is not enough free tags (i.e., less than k), then the oracle outputs $\perp$. All relations $(\mathcal{T}_i, \mathsf{ID}_{\mathcal{T}_i})$ are kept in a *a priori* secret table denoted Tab.
- $\mathcal{O}^{\textsc{Free}}(\mathcal{T})$: works exactly as the one of Vaudenay.

3.7.2 Untraceability Experiment

From the oracles given above, CCEG defines three classes of polynomial-time adversaries for the untraceability experiment.

Definition 3.11 (Adversary Class [34]). *An adversary class is said:*

- STRONG *if* $\mathcal{A}$ *has access to all the oracles;*
- DESTRUCTIVE *if* $\mathcal{A}$ *cannot use anymore a "corrupted" tag (i.e., the tag has been destroyed);*
- WEAK *if* $\mathcal{A}$ *has no access to the* $\mathcal{O}^{\textsc{Corrupt}}$ *oracle.*

The authors do not define the NARROW adversary class introduced in the Vaudenay model (see Section 3.4 for more details). They consider that the model aims to be as powerful as possible: the NARROW notion weakens the adversary.

A *link* is a couple of pseudonyms $(\mathcal{T}_i, \mathcal{T}_j)$ associated to the same identifier in Tab. Some links are considered obvious (e.g., both $\mathcal{T}_i$ and $\mathcal{T}_j$ have been corrupted). Therefore, the authors define the notion of *non-obvious link*. As remark, links are chronologically ordered, i.e., $(\mathcal{T}_i, \mathcal{T}_j)$ means that $\mathcal{T}_i$ has been freed before that $\mathcal{T}_j$ has been drawn.

Definition 3.12 (Non-Obvious Link (NOL) [34]). $(\mathcal{T}_i, \mathcal{T}_j)$ *is a* non-obvious link *if* $\mathcal{T}_i$ *and* $\mathcal{T}_j$ *refer to the same* $\mathsf{ID}_{\mathcal{T}}$ *in* Tab *and if a "dummy" adversary* $\mathcal{A}_d$*, who only has access to* $\mathcal{O}^{\textsc{CreateTag}}$*,* $\mathcal{O}^{\textsc{DrawTag}}$*,* $\mathcal{O}^{\textsc{Free}}$*, and* $\mathcal{O}^{\textsc{Corrupt}}$*, is not able to output this link with a probability better than* $\frac{1}{2}$*. Moreover, a non-obvious link is said:*

- standard *if* $\mathcal{A}$ *has not corrupted* $\mathcal{T}_i$ *or* $\mathcal{T}_j$*;*
- past *if* $\mathcal{A}$ *has corrupted* $\mathcal{T}_j$*;*
- future *if* $\mathcal{A}$ *has corrupted* $\mathcal{T}_i$*.*

Note that this model uses a "dummy" adversary $\mathcal{A}_d$, instead of a blinded adversary $\mathcal{A}^{\mathcal{B}}$ as in the Vaudenay model. Both adversaries are equivalent but not identical. Indeed, the main difference is that the Vaudenay blinder $\mathcal{B}$ is an entity clearly separated from $\mathcal{A}^{\mathcal{B}}$. Therefore, $\mathcal{B}$ does not know the random choices done by $\mathcal{A}^{\mathcal{B}}$ during the experiment.

On the opposite in CCEG, $\mathcal{A}_d$ is a single entity: she is obviously aware of her random choices.

A WEAK adversary is only able to output a *standard* NOL as she cannot query the $\mathcal{O}^{\text{Corrupt}}$ oracle. A DESTRUCTIVE adversary is not able to output a *future* NOL as a tag corruption destroys the tag (and thus prevents the tag from being drawn again). However, this adversary can output a *standard* or *past* NOL. Then, a STRONG adversary is able to output every NOL.

The CCEG untraceability experiment is detailed in Figure 3.5, where P is the adversary class such that:

$$P \in \{\mathsf{STRONG}, \mathsf{DESTRUCTIVE}, \mathsf{WEAK}\}.$$

Experiment $Exp^{\mathsf{CCEG\text{-}UNT}}_{\mathcal{S},\mathcal{A}}[\lambda]$

1. $\mathcal{C}$ initializes the system $\mathcal{S}$ and sends 1^λ, $\mathcal{S}$'s public parameters param (including $\mathsf{P}_{\mathcal{R}}$) to $\mathcal{A}$.
2. $\mathcal{A}$ interacts with the whole system, limited by her class P.
3. $\mathcal{A}$ returns one link $(\mathcal{T}_i, \mathcal{T}_j)$.

$Exp^{\mathsf{CCEG\text{-}UNT}}_{\mathcal{S},\mathcal{A}}$ succeeds if $(\mathcal{T}_i, \mathcal{T}_j)$ is a NOL.

Figure 3.5: Untraceability experiment of the CCEG model.

3.7.3 Untraceability Notions

With the previous experiment, the CCEG untraceability of a system $\mathcal{S}$ is proved if no adversary is able to output a NOL with a probability better than the one of the dummy adversary $\mathcal{A}_d$.

Definition 3.13 (Untraceability [34]). *An RFID system $\mathcal{S}$ is said* standard-untraceable *(resp.* past-untraceable */* future-untraceable*) if, for every* WEAK *(resp.* DESTRUCTIVE */* STRONG*) adversary $\mathcal{A}$ running in polynomial-time, it is possible to define a "dummy" adversary $\mathcal{A}_d$ who only has access to oracles $\mathcal{O}^{\text{CreateTag}}$, $\mathcal{O}^{\text{DrawTag}}$, $\mathcal{O}^{\text{Free}}$ and $\mathcal{O}^{\text{Corrupt}}$*

such that:

$$|\Pr(Exp^{\mathsf{CCEG\text{-}UNT}}_{\mathcal{S},\mathcal{A}}[\lambda] \ succeeds) - \Pr(Exp^{\mathsf{CCEG\text{-}UNT}}_{\mathcal{S},\mathcal{A}_d}[\lambda] \ succeeds)| \leq \epsilon(\lambda).$$

Direct implications are made from these notions:

Future-untraceability ⇒ Past-untraceability ⇒ Standard-untraceability

The main result of this paper is that future-untraceability (the strongest privacy property) is achievable.

3.8 Deng, Li, Yung and Zhao [44], 2010

Also in 2010, Deng, Li, Yung and Zhao proposed a new framework based on zero-knowledge formulation to define the security and privacy of RFID systems. Here, we only present the zero-knowledge privacy (denoted ZK-privacy), which is a new way of thinking in privacy for RFID. This model, denoted DLYZ in the sequel, is part of the *unpredictability models* family [44, 77, 106, 111]. They all rely on the unpredictability of the output returned by a tag or a reader in a protocol execution. This thesis only presents DLYZ since it is the most achieved model of this family.

3.8.1 Analyzed Protocol

This model considers that an RFID protocol execution π is, w.l.o.g., always initialized by $\mathcal{R}$, and π consists of $2\gamma + 1$ rounds of exchanged messages for some $\gamma \geq 1$. Each protocol execution π is associated to a unique identifier *sid*. At each execution, a tag may update its internal state and secret key, and $\mathcal{R}$ may update its internal state and database. The update process (of the secret key or the internal state) on a tag always erases the old values. The outputs bits $o^{sid}_{\mathcal{R}}$ and $o^{sid}_{\mathcal{T}}$ (equal to 1 if $\mathcal{R}$ and $\mathcal{T}$ accept the protocol execution with identifier *sid*, or 0 otherwise) are publicly known. The authors further claim that each tag $\mathcal{T}$ has its output bit $o^{sid}_{\mathcal{T}} = 0$ if the authentication protocol is not mutual. However, we consider this fact too limiting since $\mathcal{T}$ can have an output (possibly known by $\mathcal{A}$), even if it may not authenticate the reader. For instance, $\mathcal{T}$ can output "I arrived correctly at the end of the protocol on my side".

DLYZ assumes that $\mathcal{T}$ may participate to at most s executions in its life with $\mathcal{R}$, thus $\mathcal{R}$ is involved in at most sn executions, where s is polynomial in λ, and n is the total number of tags of the system.

3.8.2 Oracles

In a nutshell, DLYZ aims to analyze protocols where the secrets of the entities may potentially be updated at every protocol execution. Therefore, the model automatically enumerates the internal information of each entity. At the initialization of the system, the database is in an initial state, called DB^0, and already stores the secrets of all the tags, i.e., a SetupTag has already been performed on every tag. The only differences with the generic initialization procedures are the following:

- SetupReader additionally generates $\mathcal{R}$'s initial internal state $\mathsf{s}_{\mathcal{R}}^0$;
- SetupTag associates to every tag $\mathcal{T}$ a triplet $(\xi_{\mathcal{T}}, \mathsf{k}_{\mathcal{T}}^0, \mathsf{s}_{\mathcal{T}}^0)$, which is respectively $\mathcal{T}$'s public parameter, initial secret key, and initial internal state.

This information is stored in DB^0. Finally, let $\mathsf{param} = (\mathsf{P}_{\mathcal{R}}, \{\xi_{\mathcal{T}}\}_{\forall \mathcal{T}})$ denote the public parameters of the system $\mathcal{S}$. At the end of the system initialization, all the tags are accessible to the adversary.

Then, $\mathcal{A}$ has access to the following modified generic oracles.

- $\mathcal{O}^{\text{Launch}}() \rightarrow (\pi, m)$: makes $\mathcal{R}$ launch a new protocol execution π, and generates the 1^{st}-round message m which is also used as the execution identifier *sid*. If this is the j^{th} new execution run by $\mathcal{R}$, then $\mathcal{R}$ stores $sid = m$ into its internal state $\mathsf{s}_{\mathcal{R}}^j$.
- $\mathcal{O}^{\text{SendTag}}(m, \mathcal{T}) \rightarrow r$: sends m to $\mathcal{T}$. The output response r of $\mathcal{T}$ is as follows.
  1. If $\mathcal{T}$ currently does not run any execution, then $\mathcal{T}$:
     - initiates a new execution with identifier $sid = m$,
     - treats m as the 1^{st}-round message of the new execution,
     - and returns the 2^{nd}-round message r.
  2. If $\mathcal{T}$ is currently running an incomplete execution with identifier *sid*, and is waiting for the u^{th} message from $\mathcal{R}$ ($u \geq 2$), then $\mathcal{T}$ works as follows:
     - if $2 \leq u \leq \gamma$, $\mathcal{T}$ treats m as the u^{th} message from $\mathcal{R}$, and returns the next round message r;

  - if $u = \gamma+1$ (i.e., the last-round message of the execution), $\mathcal{T}$ returns its output $o_{\mathcal{T}}^{sid}$, and updates its internal state to $\mathsf{s}_{\mathcal{T}}^{v+1}$ (where *sid* corresponds to the v^{th} execution run by $\mathcal{T}$, where $1 \leq v \leq s$).

- $\mathcal{O}^{\text{SendReader}}(m, sid) \rightarrow r$: sends m to $\mathcal{R}$ for the execution with identifier *sid*. After receiving m, $\mathcal{R}$ checks from its internal state whether it is running such an execution, and responds as follows.

  1. If $\mathcal{R}$ is currently running an incomplete execution with identifier *sid*, and is waiting for the u^{th} message from a tag ($1 \leq u \leq \gamma$), then $\mathcal{R}$ works as follows:
     - if $u \leq \gamma$, $\mathcal{R}$ treats m as the u^{th} message from the tag, and returns the next round message r;
     - if $u = \gamma$, $\mathcal{R}$ returns the last-round message r and its output $o_{\mathcal{R}}^{sid}$, and updates its internal state to $\mathsf{s}_{\mathcal{R}}^{j+1}$ and the database DB^{j+1} (where *sid* corresponds to the j^{th} execution run by $\mathcal{R}$).
  2. In all the other cases, $\mathcal{R}$ returns $\perp$ (for invalid queries).

- $\mathcal{O}^{\text{Corrupt}}(\mathcal{T}) \rightarrow (\mathsf{k}_{\mathcal{T}}^{v}, \mathsf{s}_{\mathcal{T}}^{v})$: returns the secret key $\mathsf{k}_{\mathcal{T}}^{v}$ and the internal state $\mathsf{s}_{\mathcal{T}}^{v}$ currently held by $\mathcal{T}$. Once $\mathcal{T}$ is corrupted, all its actions are controlled and performed by $\mathcal{A}$.

Let $\mathcal{O}$ denote the set of these four oracles. $\mathcal{A}^{\mathcal{O}}(\mathcal{R}, T, \mathsf{param})$ denotes a PPT adversary $\mathcal{A}$ that takes on input the system public parameters param, the reader $\mathcal{R}$ and the tag set T of the already initialized system. Then $\mathcal{A}$ interacts with $\mathcal{R}$ and the tags of T via the four oracles. $\mathcal{A}'^{\mathcal{O}}(\mathcal{R}, \widehat{T}, \mathcal{I}(\mathcal{T}_c), \mathsf{aux})$ denotes a PPT adversary $\mathcal{A}'$ equivalent to $\mathcal{A}$, where $\mathsf{aux} \in \{0,1\}^*$ generally includes param or some historical state information of $\mathcal{A}'$. Then $\mathcal{A}'$ interacts with $\mathcal{R}$ and the tag set $\widehat{T}$ via the four oracles. $\mathcal{A}'$ is said to have a *blinded access* to a *challenge* tag $\mathcal{T}_c \notin \widehat{T}$ if she interacts with $\mathcal{T}_c$ via a special interface $\mathcal{I}$ (i.e., a PPT algorithm which runs $\mathcal{T}_c$ internally, and interacts with $\mathcal{A}'$ externally). To send a message m to $\mathcal{T}_c$, $\mathcal{A}'$ sends a $\mathcal{O}^{\text{SendTag}}(m, challenge)$ to $\mathcal{I}$; then $\mathcal{I}$ invokes $\mathcal{T}_c$ with $\mathcal{O}^{\text{SendTag}}(m, \mathcal{T}_c)$, and answers $\mathcal{T}_c$'s output to $\mathcal{A}'$. $\mathcal{A}'$ does not know which tag is interacting with her. $\mathcal{A}'$ interacts with $\mathcal{T}_c$ via $\mathcal{O}^{\text{SendTag}}$ queries only.

Definition 3.14 (Clean Tag [44]). *A tag $\mathcal{T}$ is said* clean *if it is not corrupted (i.e., no query to $\mathcal{O}^{\text{Corrupt}}$ on $\mathcal{T}$) and is not currently running an incomplete execution with $\mathcal{R}$ (i.e., $\mathcal{T}$'s last execution is either finished or aborted).*

The main goal of this definition is to force the adversary to use some uncorrupted and non running tags to proceed the ZK-privacy experiment (see next section). This notion of non running tags is very similar to the $\mathcal{O}^{\text{TagInit}}$ oracle of JW.

3.8.3 Privacy Experiments

In the experiments, a PPT CMIM[6] adversary $\mathcal{A}$ (resp. PPT simulator Sim) is composed of a pair of adversaries $(\mathcal{A}_1, \mathcal{A}_2)$ (resp. (Sim_1, Sim_2)), and runs in two stages. Note that, if $\delta = 0$, then no challenge tag is selected, and $\mathcal{A}$ is reduced to $\mathcal{A}_1$ in the experiment.

The first experiment detailed in Figure 3.6 is the one performed by the real adversary $\mathcal{A}$. After the system initialization, $\mathcal{A}_1$ plays with all the entities and returns a set of clean tags C. From this set C, a challenge tag $\mathcal{T}_c$ is chosen at random. Then, $\mathcal{A}_2$ plays with all the entities, including the challenge tag via the interface $\mathcal{I}$, except the set of clean tags. At the end, $\mathcal{A}$ outputs a view of the system.

Experiment $Exp^{\mathsf{ZK\text{-}priv}}_{\mathcal{S},\mathcal{A}}[\lambda, n]$ (real world)

1. $\mathcal{C}$ initializes the system $\mathcal{S}$ and sends 1^λ, param to $\mathcal{A}$.
2. $\{C, \mathsf{info}\} \leftarrow \mathcal{A}_1^{\mathcal{O}}(\mathcal{R}, T, \mathsf{param})$, where $C = \{\mathcal{T}_{i_1}, \mathcal{T}_{i_2}, \ldots, \mathcal{T}_{i_\delta}\} \subseteq T$ is a set of clean tags ($0 \leq \delta \leq n$), and info is a state information.
3. $c \in_R \{1, \ldots, \delta\}$, set $\mathcal{T}_c = \mathcal{T}_{i_c}$ and $\widehat{T} = T - C$.
4. $view_{\mathcal{A}} \leftarrow \mathcal{A}_2^{\mathcal{O}}(\mathcal{R}, \widehat{T}, \mathcal{I}(\mathcal{T}_c), \mathsf{info})$.
5. Output $(c, view_{\mathcal{A}}(\lambda, n))$.

Figure 3.6: Adversary ZK-privacy experiment of the DLYZ model.

[6] Concurrent Man-In-The-Middle.

Experiment $Exp^{\mathsf{ZK\text{-}priv}}_{\mathcal{S},Sim}[\lambda, n]$ (simulated world)

1. $\mathcal{C}$ initializes the system $\mathcal{S}$ and sends 1^λ, param to $\mathcal{A}$.
2. $\{C, \mathsf{info}\} \leftarrow Sim_1^{\mathcal{O}}(\mathcal{R}, T, \mathsf{param})$, where $C = \{\mathcal{T}_{i_1}, \mathcal{T}_{i_2}, \ldots, \mathcal{T}_{i_\delta}\} \subseteq T$ is a set of clean tags ($0 \leq \delta \leq n$), and info is a state information.
3. $c \in_R \{1, \ldots, \delta\}$ unknown to *Sim*, and set $\widehat{T} = T - C$.
4. $sview \leftarrow Sim_2^{\mathcal{O}}(\mathcal{R}, \widehat{T}, \mathsf{info})$, where *sview* includes all oracle answers to queries made by *Sim*.
5. Output $(c, sview(\lambda, n))$.

Figure 3.7: Simulator ZK-privacy experiment of the DLYZ model.

Then, the second experiment detailed in Figure 3.7 is the one performed by the simulator *Sim*. As in the previous experiment, Sim_1 plays with all the entities and returns a set of clean tags C. From this set C, a challenge tag $\mathcal{T}_c$ is chosen at random, but *Sim* is not informed about its identity and cannot play anymore with this tag. Then, Sim_2 plays with all the entities, except the set of clean tags. At the end, *Sim* outputs a simulated view of the system.

3.8.4 Privacy Notions

From the previous experiments, the ZK-privacy of a system $\mathcal{S}$ is proved when no one is able to distinguish if he is interacting with the real world or with the simulated one.

Definition 3.15 (ZK-Privacy [44]). *An RFID system $\mathcal{S}$ satisfies computational (resp. statistical)* ZK-privacy *if, for any PPT CMIM adversary $\mathcal{A}$, there exists a polynomial-time simulator Sim such that, for all sufficiently large λ and any n which is polynomial in λ, the following ensembles are computationally (resp. statistically) indistinguishable.*

- $\{c, view_{\mathcal{A}}(\lambda, n)\}_{\lambda \in \mathbb{N}, n \in poly(\lambda)}$
- $\{c, sview(\lambda, n)\}_{\lambda \in \mathbb{N}, n \in poly(\lambda)}$

That is, for any polynomial-time (resp. any computationally power unlimited) algorithm $\mathcal{D}$, it holds that:

$$|\Pr[\mathcal{D}(\lambda, n, c, view_{\mathcal{A}}(\lambda, n)) = 1] - \Pr[\mathcal{D}(\lambda, n, c, sview(\lambda, n)) = 1]| = \epsilon(\lambda).$$

The probability is taken over the random coins used during the system initialization, the random coins used by $\mathcal{A}$, Sim, $\mathcal{R}$ and all (uncorrupted) tags, the choice of c, and the coins used by the distinguisher algorithm $\mathcal{D}$.

Definition 3.16 (Forward/Backward-ZK-Privacy [44]). *Let the pair $(k_{\mathcal{T}_c}^{final}, s_{\mathcal{T}_c}^{final})$ (resp. $(k_{\mathcal{T}_c}^{0}, s_{\mathcal{T}_c}^{0})$) denote the final (resp. initial) secret key and internal state of the challenge tag $\mathcal{T}_c$ at the end (resp. beginning) of $Exp_{\mathcal{S},\mathcal{A}}^{ZK\text{-}priv}$. An RFID system $\mathcal{S}$ is* forward *(resp.* backward*)*-ZK-private *if, for any PPT CMIM adversary $\mathcal{A}$, there exists a polynomial-time simulator Sim such that, for all sufficiently large λ and any n which is polynomial in λ, the following distributions are indistinguishable.*

- $\{k_{\mathcal{T}_c}^{final}, s_{\mathcal{T}_c}^{final}(resp., k_{\mathcal{T}_c}^{0}, s_{\mathcal{T}_c}^{0}), c, view_{\mathcal{A}}(\lambda, n)\}$
- $\{k_{\mathcal{T}_c}^{final}, s_{\mathcal{T}_c}^{final}(resp., k_{\mathcal{T}_c}^{0}, s_{\mathcal{T}_c}^{0}), c, sview(\lambda, n)\}$

It is required that $\mathcal{T}_c$ should remain clean at the end of $Exp_{\mathcal{S},\mathcal{A}}^{ZK\text{-}priv}$. Note that $\mathcal{A}$ is allowed to corrupt it after the end of $Exp_{\mathcal{S},\mathcal{A}}^{ZK\text{-}priv}$.

One justification of the authors on the way of corrupting $\mathcal{T}_c$ is that it is enough to give its secrets to $\mathcal{A}$ at the end. Another reason pointed out by the authors is that forward-ZK or backward-ZK-privacy cannot be achieved if $\mathcal{A}$ corrupts $\mathcal{T}_c$ before the end of the experiment.

3.9 Hermans, Pashalidis, Vercauteren and Preneel [82], 2011

Following the path opened by Vaudenay with his privacy model, Hermans, Pashalidis, Vercauteren and Preneel presented in 2011 a new model, denoted here HPVP, based on indistinguishability between two "worlds": it is most commonly called the "left-or-right" paradigm.

The main goal of the authors was to propose a model with a clear defined purpose, that is straightforward to use for proving privacy. Also as CCEG, HPVP aimed to use the Vaudenay strongest adversary.

3.9.1 Oracles

As for Vaudenay and CCEG, DB is empty after the initialization of the system, and a tag can be either free or drawn. Then $\mathcal{A}$ has access to the generic oracles $\mathcal{O}^{\text{CreateTag}}$ (here it additionally returns a reference $\mathcal{T}$ to the new created tag), $\mathcal{O}^{\text{SendReader}}$, $\mathcal{O}^{\text{Result}}$. Then, $\mathcal{A}$ has also access to these other following oracles.

- $\mathcal{O}^{\text{DrawTag}}(\mathcal{T}_i, \mathcal{T}_j) \rightarrow \mathcal{T}_{\mathsf{drawn}}$: generates a drawn tag $\mathcal{T}_{\mathsf{drawn}}$ and stores $(\mathcal{T}_{\mathsf{drawn}}, \mathcal{T}_i, \mathcal{T}_j)$ in a table Tab. Depending on the bit b chosen at the start of the privacy experiment (see next section), $\mathcal{T}_{\mathsf{drawn}}$ will either reference $\mathcal{T}_i$ or $\mathcal{T}_j$. If one of the two tags $(\mathcal{T}_i, \mathcal{T}_j)$ is already referenced in Tab, then it outputs $\perp$.
- $\mathcal{O}_b^{\text{Free}}(\mathcal{T}_{\mathsf{drawn}})$: recovers the tuple $(\mathcal{T}_{\mathsf{drawn}}, \mathcal{T}_i, \mathcal{T}_j)$ in Tab. If $b = 0$ then it resets $\mathcal{T}_i$, otherwise it resets $\mathcal{T}_j$. Then it removes the tuple from Tab. When a tag is reset, its volatile memory is erased, not its non-volatile memory (which contains its secret $\mathsf{k}_{\mathcal{T}}$).

This specific definition of the $\mathcal{O}^{\text{Free}}$ oracle comes from one important statement highlighted by Paise and Vaudenay in their model (see Section 3.4.4 for more details).

Finally $\mathcal{A}$ has access to the following modified generic oracles.

- $\mathcal{O}^{\text{Launch}}() \rightarrow (\pi, m)$: makes $\mathcal{R}$ launch a new Ident protocol execution π, together with $\mathcal{R}$'s first message m.
- $\mathcal{O}^{\text{SendTag}}(m, \mathcal{T}) \rightarrow r$: retrieves the tuple $(\mathcal{T}, \mathcal{T}_i, \mathcal{T}_j)$ in Tab. It sends a message m to the corresponding tag ($\mathcal{T}_i$ if $b = 0$, $\mathcal{T}_j$ otherwise). It outputs the response r of the tag. If $\mathcal{T}$ is not found in Tab, it returns $\perp$.
- $\mathcal{O}^{\text{Corrupt}}(\mathcal{T}) \rightarrow \mathsf{k}_{\mathcal{T}}$: returns the whole memory (including the current secret $\mathsf{k}_{\mathcal{T}}$) of $\mathcal{T}$. If $\mathcal{T}$ is drawn, it returns $\perp$.

All these oracles are very similar to the ones of Vaudenay, but with important differences. First, $\mathcal{O}^{\text{DrawTag}}$ is only applied on two tags chosen by the adversary when she queries this oracle. Then, $\mathcal{O}^{\text{Free}}$ clearly specifies that it erases the volatile memory of the chosen tag. Lastly, $\mathcal{O}^{\text{Corrupt}}$ is only authorized on a free tag. However, the intrinsic definition of a free tag (given in the Vaudenay model [162]) is that it is not

accessible to $\mathcal{A}$, since it is not in her neighborhood. Thus, it seems impossible for $\mathcal{A}$ to query a $\mathcal{O}^{\text{CORRUPT}}$ on a tag that she cannot manipulate (i.e., not drawn).

3.9.2 Privacy Experiment

The authors keep the same adversary classes as the ones given by Vaudenay: STRONG, DESTRUCTIVE, FORWARD, WEAK, and NARROW.

Their privacy experiment is detailed in Figure 3.8, where P represents the adversary class such that:

$$P \in \{\emptyset, \text{NARROW}\} \times \{\text{WEAK}, \text{FORWARD}, \text{DESTRUCTIVE}, \text{STRONG}\}.$$

Experiment $Exp^{\text{HPVP-priv}}_{\mathcal{S},\mathcal{A}}[\lambda, b]$

1. $\mathcal{C}$ initializes the system $\mathcal{S}$, chooses a random bit b, and sends 1^λ and $\mathcal{S}$'s public parameters param to $\mathcal{A}$.
2. $\mathcal{A}$ interacts with the whole system, limited by her class P.
3. $\mathcal{A}$ outputs a guess bit b'.

$Exp^{\text{HPVP-priv}}_{\mathcal{S},\mathcal{A}}$ succeeds if $b = b'$.

Figure 3.8: Privacy experiment of the HPVP model.

3.9.3 Privacy Notions

From the previous experiment, the HPVP privacy property is based on the adversary advantage to distinguish the two worlds.

Definition 3.17 (Privacy [82]). *The RFID system $\mathcal{S}$ is said to unconditionally (resp. computationally) provide P-*privacy *if and only if, for all the adversaries (resp. polynomial time adversaries) which belong to class P, it holds that:*

$$|\Pr(Exp^{\textit{HPVP-priv}}_{\mathcal{S},\mathcal{A}}[\lambda, 0] \textit{ succeeds}) + \Pr(Exp^{\textit{HPVP-priv}}_{\mathcal{S},\mathcal{A}}[\lambda, 1] \textit{ succeeds}) - 1| = 0$$
$$\textit{(resp. } \leq \epsilon(\lambda)\textit{)}.$$

Note that the authors claim that the already existing models do not take care about some privacy leakage information such as the cardinality of the tag set. Yet, they never prove nor explain in their paper how their model can handle this issue, nor why this is indeed a privacy issue.

Chapter 4

Untangling RFID Privacy Models

When an RFID system is created, its designer can formally assess the privacy level of the system within a cryptographic model. Yet, a system designer unfamiliar with privacy can get confused with so many existing models, and may not use the most adapted model to analyze his system. Consequently, providing an analysis and a comparison of the eight most well-known RFID privacy models presented in Chapter 3 is meaningful to help a designer in his choice. A similar work has already been independently achieved in [157], but it mainly focuses on the relations among the privacy notions of the models. It does not fully review the strengths and weaknesses of each model, and thus considers some models as weak, even though they offer interesting properties.

In this chapter, we provide a thorough study of RFID privacy models. We first present five different authentication protocols (namely SK-Prot, MW-Prot, OSK-Prot, O-FRAP, PK-Prot). We analyze these protocols with the Avoine, JW, Vaudenay, LBM, DMR, CCEG, DLYZ, and HPVP models. This study exhibits the lack of granularity of these models, meaning that no model can fairly analyze and compare protocols that are designed with different security levels. Then, we thoroughly compare the eight models regarding their different features and privacy notions. We show that no model encompasses all the others. We however point out the most appropriate model(s) to use for analyzing a protocol in specific scenarios.

4.1 Privacy Analysis of Different Protocols

To investigate more deeply the differences between the presented models, the privacy level of five different protocols is studied in each model where the RFID system is assumed to be composed of one reader and n tags.

These protocols are (variant of) the ones published in [89, 107, 117, 162], and differ according to their building blocks and their underlying key infrastructure. The four first ones are based on symmetric-key cryptography. In the first protocol, each tag is attached to a unique long-term secret key, i.e., a key that never changes during the tag life-time. On the contrary in the second protocol, each tag is attached to a set of unique long-term secret keys, but each key is potentially shared between some tags to speed up tag authentication. The third and fourth protocols use key-update mechanisms to increase the security level in case of tag corruption. In particular, the fourth one provides mutual authentication in order to be undesynchronizable. The last analyzed protocol is based on public-key cryptography.

Due to their disparities, they may thus ensure different privacy levels. Yet, this section shows that some models assign the same privacy level to some protocols while other models clearly differentiate them, e.g., taking into account an attack which cannot be modeled in other models.

4.1.1 SK-Prot Authentication Protocol

Note on the ISO/IEC 9798 [89]. This standard is one of the current international ones for authentication.

Parts 2 to 4 of the standard provide four authentication protocols that are respectively based on symmetric encryption functions (ISO/IEC 9798-2), digital signature schemes (ISO/IEC 9798-3), and cryptographic hash functions (ISO/IEC 9798-4). Each part describes three mechanisms for achieving authentication: unilateral authentication with timestamps, unilateral authentication with random numbers, and mutual authentication with random numbers.

Part 5 of the standard provides several mechanisms using zero-knowledge techniques that are based on integer factorization, discrete logarithms with respect to prime or composite numbers, and asymmetric encryption. Such mechanisms can be zero-knowledge proofs, such as the Fiat-Shamir [65] protocol or GPS [22].

Description of the Protocol. This protocol has been proposed in [162]. It is equivalent to the ISO/IEC 9798-2 Mechanism 2 [89] with an additional random number chosen by the tag and where the original symmetric-key cryptosystem has been replaced by a PRF[1]. A tag $\mathcal{T}$ has a unique secret key $\mathsf{k}_\mathcal{T}$ shared with $\mathcal{R}$ used for the challenge/response authentication as depicted in Figure 4.1.

Reader $\mathcal{R}$		**Tag** $\mathcal{T}$
$\mathsf{ID}_\mathcal{T}, \mathsf{k}_\mathcal{T}$		$\mathsf{ID}_\mathcal{T}, \mathsf{k}_\mathcal{T}$
• $n_\mathcal{R} \in_R \{0,1\}^\lambda$		
	$\xrightarrow{n_\mathcal{R}}$	
		• $n_\mathcal{T} \in_R \{0,1\}^\lambda$ • $E = F_{\mathsf{k}_\mathcal{T}}(n_\mathcal{R}, n_\mathcal{T})$
	$\xleftarrow{n_\mathcal{T}, E}$	
• Find $(\mathsf{ID}_\mathcal{T}, \mathsf{k}_\mathcal{T}) \in \mathsf{DB}$ s.t. $E = F_{\mathsf{k}_\mathcal{T}}(n_\mathcal{R}, n_\mathcal{T})$		

Figure 4.1: SK-Prot authentication protocol.

Reader Complexity. To authenticate a tag, the reader carries out a linear exhaustive search on its database to find the correct pair $(\mathsf{ID}_\mathcal{T}, \mathsf{k}_\mathcal{T})$ among the pairs of all the tags of the system. The reader complexity is thus in $O(n)$.

Privacy Attack. When an adversary $\mathcal{A}$ corrupts a tag $\mathcal{T}$, she recovers its secret key $\mathsf{k}_\mathcal{T}$. Let assume that $\mathcal{A}$ is able to eavesdrop some protocol executions. Since $\mathsf{k}_\mathcal{T}$ is a fixed value and the nonces used in the PRF are sent in the clear during each protocol execution, $\mathcal{A}$ is able to recompute $F_{\mathsf{k}_\mathcal{T}}(n_\mathcal{R}, n_\mathcal{T})$ and compare it with the value E sent by the tag performing one authentication. If these values are equal, then $\mathcal{A}$ is convinced that the corrupted tag $\mathcal{T}$ performed this authentication[2]. So clearly, one single corruption of $\mathcal{T}$ allows an adversary $\mathcal{A}$ to trace it unconditionally. Nevertheless, the corruption of another tag $\mathcal{T}'$ does not necessary help $\mathcal{A}$

[1] Pseudo-Random Function.

[2] Note that this equality can be due to a collision, but this happens with a negligible probability.

to trace the tag $\mathcal{T}$, since all the secret keys are independent. It only aids $\mathcal{A}$ in knowing whether or not a protocol execution has been executed by $\mathcal{T}'$. Consequently, this protocol can only reach privacy properties when $\mathcal{A}$ is not allowed to corrupt the tags she wants to trace.

Privacy Levels. This protocol is Existential-UNT-RTE in the Avoine model (proved for this kind of protocol in [9]), and (ρ, σ, τ)-private in the JW model (proved in [99]). It is untraceable for DMR (proved in [47]), and ZK-private in the DLYZ model (the proof of a similar protocol in [44] can be trivially adapted).

This protocol is WEAK-private for Vaudenay (proved in [162]) and for HPVP. It is standard-untraceable for CCEG. The proofs for HPVP and CCEG are very similar to the one of Vaudenay.

Finally, this protocol cannot UC-emulate the ideal functionality in the LBM model as the attack presented here permits an adversary to link several executions while this is not possible for the simulator (as $state(\mathcal{T})$ is removed after a corruption).

Table 4.1 sums up the privacy analysis of SK-Prot.

Model	SK-Prot
Avoine	Existential-UNT-RTE
JW	(ρ, σ, τ)-privacy
Vaudenay	WEAK-privacy
LBM	✗
DMR	Untraceability
CCEG	Standard-untraceability
DLYZ	ZK-privacy
HPVP	WEAK-privacy

Table 4.1: Summary of the SK-Prot analysis. "✗" means no privacy.

4.1.2 MW-Prot Authentication Protocol

This protocol is also based on symmetric-key cryptography. It is a variant of the protocol [117] which is the original one that introduced the MW key infrastructure presented in Section 2.1.2.

Description of the Protocol. As detailed in Section 2.1.2, $\mathcal{T}$ is initialized with a set of keys $\{\mathsf{k}_{p_1}, \mathsf{k}_{p_2}, \ldots, \mathsf{k}_{p_d}\}$, where each k_{p_i} is the secret key attached to its path node p_i (except the root). At the setup of the system, $\mathcal{R}$ knows the entire tree arrangement (of depth d and branching factor β) and all the keys associated to each node.

The protocol is carried out in d rounds. For each round, $\mathcal{R}$ and $\mathcal{T}$ perform a challenge/response authentication as described in Figure 4.2. As caption, DB_i corresponds to the set of the β possible keys at tree level i ($1 \leq i \leq d$). If $\mathcal{T}$ correctly answers at each round, then $\mathcal{R}$ successfully authenticates $\mathcal{T}$ at the end of the last round.

Reader $\mathcal{R}$		**Tag** $\mathcal{T}$
$\mathsf{ID}_\mathcal{T}, \mathsf{k}_{p_1}, \ldots, \mathsf{k}_{p_d}$		$\mathsf{ID}_\mathcal{T}, \mathsf{k}_{p_1}, \ldots, \mathsf{k}_{p_d}$
• $n_\mathcal{R} \in_R \{0,1\}^\lambda$		
	$\xrightarrow{n_\mathcal{R}}$	
		• $n_\mathcal{T} \in_R \{0,1\}^\lambda$ • $E = F_{\mathsf{k}_{p_i}}(n_\mathcal{R}, n_\mathcal{T})$
	$\xleftarrow{n_\mathcal{T}, E}$	
• Find $\mathsf{k}_{p_i} \in \mathsf{DB}_i$ s.t. $E = F_{\mathsf{k}_{p_i}}(n_\mathcal{R}, n_\mathcal{T})$		

Figure 4.2: i^{th} round of the MW-Prot authentication protocol.

Reader Complexity This protocol inherits from the reader complexity provided by the MW key infrastructure, that is $O(\log(n))$.

Privacy Attack. The main drawback in this protocol is that some keys are shared by several tags. As example, let assume that an adversary $\mathcal{A}$ chooses and corrupts a random tag $\mathcal{T}$: its secret keys $(\mathsf{k}_{1,1}, \mathsf{k}_{2,1}, \mathsf{k}_{3,1}, \ldots)$ are revealed, where $\mathsf{k}_{i,j}$ denotes the j^{th} key of the tree level i. Then, let assume that $\mathcal{A}$ wants to trace the tags $\mathcal{T}_0$ and $\mathcal{T}_1$ which are initialized as follows: $\mathcal{T}_0$'s keys are $(\mathsf{k}_{1,1}, \mathsf{k}_{2,1}, \mathsf{k}_{3,2}, \ldots)$, and $\mathcal{T}_1$'s keys are $(\mathsf{k}_{1,1}, \mathsf{k}_{2,2}, \mathsf{k}_{3,3}, \ldots)$. Clearly, $\mathcal{T}_0$ and $\mathcal{T}_1$ share the same path for the first node, since they have the same key for p_1. But they have different keys for p_2, and consequently for p_3 as well. From the keys revealed during $\mathcal{T}$'s corruption, it is therefore possible to differentiate $\mathcal{T}_0$ and $\mathcal{T}_1$: $\mathcal{T}_0$'s

answers will always be verifiable with $(\mathsf{k}_{1,1}, \mathsf{k}_{2,1})$, but this is not the case for $\mathcal{T}_1$ since it does not use the revealed key $\mathsf{k}_{2,1}$. Note that, in the example, the tags that $\mathcal{A}$ wants to trace are not corrupted: only one other tag is corrupted.

Privacy Levels. This protocol faces the same problem as SK-Prot: the corruption of a tag allows an adversary $\mathcal{A}$ to trace it unconditionally. As a consequence for all the models, we consider that $\mathcal{A}$ is not allowed to corrupt (at least) the challenge tags. Note that this option is not available in LBM, and this protocol is consequently not forward-secure.

It should not be possible to study this kind of protocol in the Avoine model because of the correlated secrets, but the analysis is given here to show the contrasts between the different models. Thus in the Avoine model, since $\mathcal{A}$ only plays with the two challenge tags, the protocol does not suffer from the previous attack, and the protocol is Existential-UNTRTE (same proof as for SK-Prot). For Vaudenay and HPVP, the protocol is WEAK-private, and standard-untraceable for CCEG: clearly, since no secret is revealed, the proof is similar to the one for SK-Prot.

Then $\mathcal{A}$ is able to corrupt the non-challenge tags in JW, and the revealed keys are part of the adversary knowledge in DMR. Thus, the attack presented above can be formalized in these two models. Consequently, the protocol is not (ρ, σ, τ)-private for JW (explained in [99], proved in Chapter 2 and in [15, 17]), and not untraceable for DMR.

For DLYZ, we use the method provided in [44] to show that the protocol is not ZK-private. We consider that *Sim* runs as subroutine the underlying adversary $\mathcal{A}$. Sim_1 just runs basically $\mathcal{A}_1$, and both adversaries obtain several keys from the corruption of non clean tags in the first phase. Let us also consider that $\mathcal{A}_1$ and Sim_1 return a set C of clean tags where (i) $|C| \geq 2$ and (ii) each tag in C can be easily recognizable, thanks to the revealed keys. Then $\mathcal{A}_2$ will be able the recognize the challenge tag chosen in C. But Sim_2 is forbidden to interact with the chosen challenge tag, and does not know additional information about it. Thus Sim_2 has to choose at random a tag to simulate the challenge one whose interactions will be released in its view. At the end of the experiment, $\mathcal{A}$ will always retrieve the correct challenge tag, contrary to *Sim*: the views of $\mathcal{A}$ and *Sim* will be distinguishable. Therefore, the protocol is not ZK-private.

Table 4.2 sums up the privacy analysis of MW-Prot.

Model	MW-Prot
Avoine	Existential-UNT-RTE
JW	✗
Vaudenay	WEAK-privacy
LBM	✗
DMR	✗
CCEG	Standard-untraceability
DLYZ	✗
HPVP	WEAK-privacy

Table 4.2: Summary of the MW-Prot analysis. "✗" means no privacy.

4.1.3 OSK-Prot Authentication Protocol

The Original OSK Protocol [134]. OSK is an identification protocol, where there is no proof of the tag identity. At the setup, $\mathcal{T}$ is initialized with a unique secret key $\mathsf{k}_\mathcal{T}$ shared with $\mathcal{R}$. $\mathcal{T}$ just sends the result of a one-way function performed on its key.

OSK is one of the first synchronized protocols proposed in the RFID literature. This comes from its main feature where both $\mathcal{T}$ and $\mathcal{R}$ update the shared key after each protocol execution. $\mathcal{T}$ and $\mathcal{R}$ thus need to be in possession of the same key to correctly operate an identification.

The OSK protocol has been introduced to ensure the *forward security* property, i.e., data sent by a given tag $\mathcal{T}$ today will still be secure even if $\mathcal{T}$'s secret key is disclosed by tampering with this tag in the future, contrary to SK-Prot.

Description of the Protocol. The protocol presented here is the one proposed in [162]. It is slightly different from OSK for two reasons. First, $\mathcal{R}$ additionally sends a nonce to $\mathcal{T}$ in order to prevent replay attacks, as described in [15]. It thus ensures tag authentication rather than elementary tag identification. Secondly, $\mathcal{R}$ is also initialized with a value δ playing the role of a threshold. When $\mathcal{T}$ sends its answer E to $\mathcal{R}$, the latter verifies E for each pair $(\mathsf{ID}_\mathcal{T}, \mathsf{k}_\mathcal{T})$ of the database. If no pair corresponds to this result, $\mathcal{R}$ verifies E considering that $\mathsf{k}_\mathcal{T}$ might

have been updated once, then twice, and so on. This procedure may be repeated for at most δ key updates. At the end, if no pair corresponds to this result for δ key updates, then $\mathcal{R}$ rejects the tag $\mathcal{T}$.

Reader $\mathcal{R}$ $\mathsf{ID}_\mathcal{T}, \mathsf{k}_\mathcal{T}, \delta$		**Tag $\mathcal{T}$** $\mathsf{ID}_\mathcal{T}, \mathsf{k}_\mathcal{T}$
• $n_\mathcal{R} \in_R \{0,1\}^\lambda$		
	$\xrightarrow{n_\mathcal{R}}$	
		• $E = f(\mathsf{k}_\mathcal{T}, n_\mathcal{R})$ • $\mathsf{k}_\mathcal{T} = g(\mathsf{k}_\mathcal{T})$
	$\xleftarrow{E}$	
• Find $(\mathsf{ID}_\mathcal{T}, \mathsf{k}_\mathcal{T}) \in \mathsf{DB}$ s.t. $E = f(g^i(\mathsf{k}_\mathcal{T}), n_\mathcal{R})$ and $i \leq \delta$ • $\mathsf{k}_\mathcal{T} = g^i(\mathsf{k}_\mathcal{T})$		

Figure 4.3: OSK-Prot authentication protocol.

Reader Complexity. As for SK-Prot, the reader carries out a linear exhaustive search on its database to authenticate a tag. This exhaustive search may further be repeated at most δ times. The reader complexity of this protocol is thus in $O(n.\delta)$.

Privacy Attack. A significant attack on this family of protocols has been defined by Juels and Weis in their privacy model [99] based on the fact that the key of a tag can be updated while the equivalent one stored by the reader is not. Note that upon receipt of a message E, $\mathcal{R}$ tries to find a match with one of the tag keys or their δ first updates. Thus, if the adversary $\mathcal{A}$ sends more than δ consecutive authentication requests to a tag without transferring the answers to $\mathcal{R}$, the shared secrets stored in $\mathcal{T}$ and $\mathcal{R}$ are consequently desynchronized. Therefore, if $\mathcal{A}$ has access to the authentication result on the reader side, she is able to recognize a desynchronized tag $\mathcal{T}$ from another random tag as $\mathcal{T}$ will be rejected. This attack is generally called a *desynchronization attack*.

Privacy Levels. Recall that a NARROW adversary does not have access to the authentication result on the reader side, while a WIDE one does have this access (e.g., through a $\mathcal{O}^{\text{Result}}$ query).

Considering a NARROW adversary, under the one-wayness assumption of g, it is obviously infeasible to link a secret key to a previous authentication transcript as this is equivalent to invert g. Furthermore, since all the secrets of the system are independent, then corrupting one tag does not allow tracing the other ones. Since $\mathcal{A}$ is restricted to be NARROW in the Avoine and DMR models, the desynchronization attack does not work and thus the privacy level is equivalent to the one of SK-Prot, namely the protocol is respectively Existential-UNT-RTE (proved in [9]) and untraceable (proof similar to the one in [47]). Considering tag corruption, it is furthermore Forward-UNT-RTEC in the Avoine model (proved in [9]). Regarding the Vaudenay and HPVP models, the protocol is NARROW-DESTRUCTIVE-private (proved in [162, 82]).

When $\mathcal{A}$ is WIDE, the protocol is vulnerable to the desynchronization attack explained above. Therefore, the protocol is not (ρ, σ, τ)-private for JW when $(\rho \geq 1, \sigma > \delta, \tau > \delta)$ (proved in [99]), and not standard-untraceable for CCEG. In the LBM model, a legitimate tag cannot be rejected in the ideal world as the ideal functionality will always accept it, while the desynchronization attack works in the real world.

For DLYZ, the same problem as for MW-Prot appears. If $|C| = 2$ and one of the two tags has been desynchronized by $\mathcal{A}_1$, then $\mathcal{A}_2$ can distinguish these tags depending on the result of an execution in the second phase. But Sim_2 is forbidden to interact with the chosen challenge tag, and does not know additional information about it. Thus Sim_2 has to choose at random a tag to simulate the challenge one (as victim or not of the desynchronization attack) . At the end of the experiment, $\mathcal{A}$ is always able to retrieve the correct challenge tag, which is not the case of *Sim*. This implies that the views of $\mathcal{A}$ and *Sim* will be distinguishable. Therefore, the protocol is not ZK-private.

Table 4.3 sums up the privacy analysis of OSK-Prot.

4.1.4 O-FRAP Authentication Protocol

Undesynchronizable Protocols. Several protocols [30, 33, 49, 107] have been proposed to counter the desynchronization drawback of the OSK-Prot protocol[3]. They are based on mutual authentication, where

[3] A larger survey about the existing improvements of the original OSK protocol has been published by Avoine, Bingol, Carpent, and Ors in [11].

Model	OSK-Prot
Avoine	Existential-UNT-RTE + Forward-UNT-RTEC
JW	✗
Vaudenay	NARROW-DESTRUCTIVE-privacy
LBM	✗
DMR	Untraceability
CCEG	✗
DLYZ	✗
HPVP	NARROW-DESTRUCTIVE-privacy

Table 4.3: Summary of the OSK-Prot analysis. "✗" means no privacy.

tags only update their key after authenticating the reader. This characteristic ensures that the key inside the tag and the one inside the reader database can be desynchronized at most one time.

Description of the Protocol. O-FRAP is an undesynchronizable protocol introduced by Le, Burmester and de Medeiros in [107].

At the setup, $\mathcal{T}$ is initialized with a pair $(\mathsf{k}_\mathcal{T}, n_\mathcal{T})$ containing a secret key and a nonce. $(\mathsf{k}_\mathcal{T}, n_\mathcal{T})$ is stored by $\mathcal{R}$ as the current secret $cur_\mathcal{T}$ of $\mathcal{T}$. Then, a mutual authentication between $\mathcal{R}$ and $\mathcal{T}$ is performed, as depicted in Figure 4.4, where $\mathcal{T}$'s key and/or nonce are updated by both entities after each protocol execution. The main difference with OSK is that the tag updates its key $\mathsf{k}_\mathcal{T}$ only after having successful authenticated $\mathcal{R}$.

The SearchID procedure is detailed in Algorithm 1, where the internal Update($\mathcal{T}$) procedure works as follows. Firstly, if $\mathcal{R}$ uses $cur_\mathcal{T}$ to identify $\mathcal{T}$, then $\mathcal{R}$ replaces the content of $old_\mathcal{T}$ with the one of $cur_\mathcal{T}$. Secondly, $\mathcal{R}$ refreshes $cur_\mathcal{T} = (\mathsf{k}_\mathcal{T}^{cur}, n_\mathcal{T}^{cur})$ by (ν'_4, ν'_1).

Reader Complexity. The SearchID procedure clearly states that the reader complexity of the O-FRAP protocol is in $O(n)$.

Privacy Attack. Avoine, Coisel and myself described in [13] an attack which can be applied to the undesynchronizable protocols (e.g., [33, 49]), and which works when the adversary $\mathcal{A}$ is able to corrupt a challenge

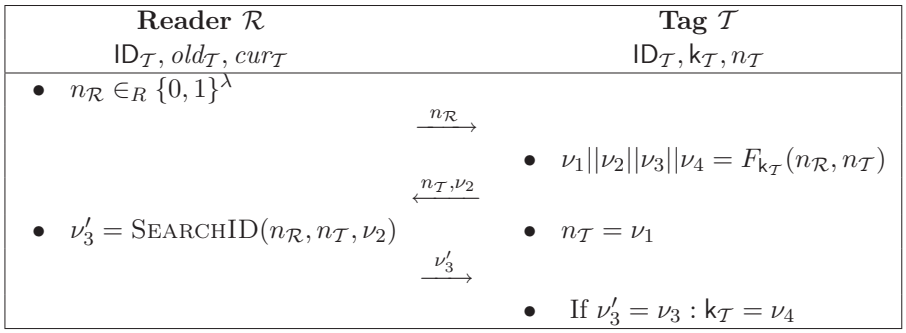


Figure 4.4: O-FRAP authentication protocol.

Algorithm 1 : The SEARCHID procedure

Input: $n_\mathcal{R}, n_\mathcal{T}, \nu_2$
Output: ν'_3

1: $Out \leftarrow \perp$
2: **if** $\exists(\mathsf{ID}_\mathcal{T}, old_\mathcal{T}, cur_\mathcal{T}) \in \mathsf{DB}$ s.t. $n_\mathcal{T} = n_\mathcal{T}^{old}$ (resp. $n_\mathcal{T} = n_\mathcal{T}^{cur}$) **then**
3: $\nu'_1||\nu'_2||\nu'_3||\nu'_4 \leftarrow F_{\mathsf{k}_\mathcal{T}^{old}}(n_\mathcal{R}, n_\mathcal{T}^{old})$
(resp. $\nu'_1||\nu'_2||\nu'_3||\nu'_4 \leftarrow F_{\mathsf{k}_\mathcal{T}^{cur}}(n_\mathcal{R}, n_\mathcal{T}^{cur})$)
4: **if** $\nu'_2 = \nu_2$ **then**
5: $\mathcal{T}$ is correctly authenticated
6: $Out \leftarrow \nu'_3$
7: UPDATE($\mathcal{T}$)
8: **end if**
9: **end if**
10: **for all** $(\mathsf{ID}_\mathcal{T}, old_\mathcal{T}, cur_\mathcal{T}) \in \mathsf{DB}$ **and** $i \in \{old_\mathcal{T}, cur_\mathcal{T}\}$ **do**
11: $\nu'_1||\nu'_2||\nu'_3||\nu'_4 \leftarrow F_{\mathsf{k}_\mathcal{T}^i}(n_\mathcal{R}, n_\mathcal{T})$
12: **if** $\nu'_2 = \nu_2$ **then**
13: $\mathcal{T}$ is correctly authenticated
14: $Out \leftarrow \nu'_3$
15: UPDATE($\mathcal{T}$)
16: **end if**
17: **end for**
18: **return** Out

tag. We adapt this attack to the O-FRAP protocol. First, $\mathcal{A}$ makes $\mathcal{T}$ and $\mathcal{R}$ start a new protocol execution, but $\mathcal{A}$ blocks the last message sent from $\mathcal{R}$ to $\mathcal{T}$. Then, if $\mathcal{A}$ corrupts $\mathcal{T}$ directly after this incomplete execution, she is able to recognize $\mathcal{T}$ by recomputing ν_2 as $\mathsf{k}_{\mathcal{T}}$ has not been updated and the nonces $(n_{\mathcal{R}}, n_{\mathcal{T}})$ have been sent in the clear. Note that the traceability attack of O-FRAP proposed in [138] is specific to the way they define Algorithm 1, and does not apply here.

Privacy Levels. According to the previous attack, no $\mathcal{O}^{\text{Corrupt}}$ query should be allowed to an adversary of this protocol. In that case, the desynchronization attack of OSK does not work here either. As a consequence, for JW, Vaudenay, CCEG and HPVP, the privacy level of O-FRAP is the same as the one of SK-Prot (proofs are equivalent): it is respectively (ρ, σ, τ)-private, WEAK-private, standard-untraceable, and WEAK-private.

In the Avoine and DMR models, the protocol is Existential-UNT-RTE and untraceable: the attack presented above without corruption does not work since the tag keys are needed. The proofs are thus similar to the ones of SK-Prot. The protocol is furthermore Forward-UNT-RTEC for Avoine, because in that case, $\mathcal{C}$ can give $\mathcal{A}$ non-consecutive intervals (contrary to the ones needed for the above attack): thus corrupting a tag does not help $\mathcal{A}$ to trace a tag.

Since the analysis for LBM is only related to completed protocol executions, this attack can be perfectly simulated in the ideal world using the knowledge of $active(\mathcal{T})$ as proved in [107]. The protocol is thus forward-secure.

For DLYZ, the protocol is ZK-private: the proof is similar to the one of SK-Prot when no corruption is allowed. Regarding the forward-ZK-privacy, it is possible to define an adversary $\mathcal{A}$ who has a distinguishable view than the simulator's one. Let us consider that $|C| \geq 2$. Sim_1 just runs $\mathcal{A}_1$ as subroutine. Then $\mathcal{A}_2$ forces an interaction between $\mathcal{R}$ and $\mathcal{T}_c$, and blocks the last message. Sim_2 has to provide a simulated incomplete interaction of $\mathcal{R}$ with $\mathcal{T}_c$: since Sim_2 cannot interact with $\mathcal{T}_c$ and does not have any information about it, this interaction can only be composed of random messages. At the end, $\mathcal{T}_c$'s secrets are revealed to a distinguisher $\mathcal{D}$. Thus $\mathcal{D}$ is able to recognize if $\mathcal{A}_2$'s interaction corresponds to a real incomplete interaction with $\mathcal{T}_c$, or a simulated one.

The protocol is therefore not forward-ZK-private.

Table 4.4 sums up the privacy analysis of O-FRAP.

Model	O-FRAP
Avoine	Existential-UNT-RTE + Forward-UNT-RTEC
JW	(ρ, σ, τ)-privacy
Vaudenay	WEAK-privacy
LBM	Forward-security
DMR	Untraceability
CCEG	Standard-untraceability
DLYZ	ZK-privacy
HPVP	WEAK-privacy

Table 4.4: Summary of the O-FRAP analysis. "✗" means no privacy.

4.1.5 PK-Prot Authentication Protocol

In [162], Vaudenay defines a generic solution to use public-key cryptography for RFID authentication. The following protocol is the one proposed by Vaudenay without the method for shrinking the database.

Description of the Protocol. At the system setup, $\mathcal{R}$ is initialized with a pair of public/private keys $(\mathsf{P}_{\mathcal{R}}, \mathsf{K}_{\mathcal{R}})$, and $\mathcal{T}$ is attached to a unique secret key $\mathsf{k}_{\mathcal{T}}$ known by $\mathcal{R}$. Then, $\mathcal{R}$ and $\mathcal{T}$ engage in a challenge/response authentication as depicted in Figure 4.5. The cryptosystem (Enc/Dec) is considered to be either IND-CPA[4] or IND-CCA[5] secure [23, 52, 73, 121, 141].

Reader Complexity. Contrary to the other protocols, the reader directly deciphers the tag answer with its private key $\mathsf{K}_{\mathcal{R}}$, and obtains the data to authenticate the tag communicating with it. $\mathcal{R}$ then checks if these data belong to its database in constant time. Therefore, the reader complexity of PK-Prot is in $O(1)$.

[4]INDistinguishable under Chosen-Plaintext Attack.
[5]INDistinguishable under Chosen-Ciphertext Attack.

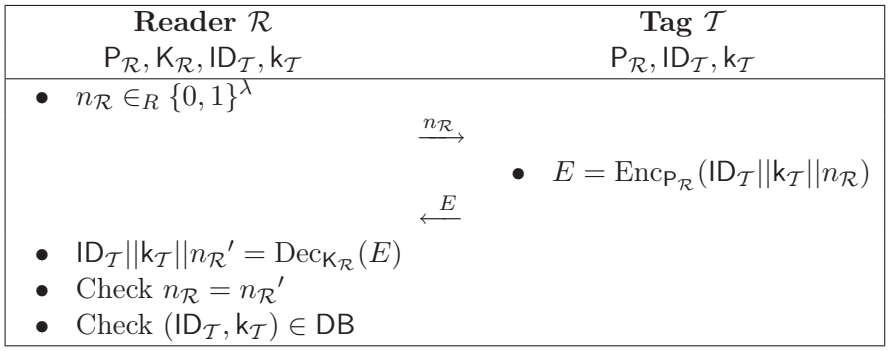


Figure 4.5: PK-Prot authentication protocol.

Privacy Levels. First, it is important to note that this protocol may not be easily proved private for WIDE adversaries in any model under IND-CPA security. The main reason is that the simulator/blinder in the proof does not have access to a decryption oracle in the IND-CPA experiment. Therefore, this simulator/blinder is unable to correctly simulate the $\mathcal{O}^{\text{Result}}$ oracle, and thus has to answer at random 0 or 1 in some cases. Here, an adversary $\mathcal{A}$ may be able to detect if it is interacting with the real world or with a simulated one. The authors of CCEG proved in [34] that standard-untraceability can nevertheless be reached by public-key-based protocols using an IND-CPA cryptosystem, but by adding other security mechanisms to these protocols (namely a MAC[6]).

For Avoine and DMR, since $\mathcal{A}$ is NARROW, this problem does not appear (i.e., no query to $\mathcal{O}^{\text{Result}}$). When the cryptosystem is IND-CPA secure, the protocol is thus Existential-UNT-RTE and Forward-UNT-RTEC for Avoine, and untraceable for DMR. The proof is as follows in the Avoine model, but can be easily adapted for DMR.

Proof (in the Avoine model). The goal is to show that, if there exists an adversary $\mathcal{A}$ who wins $Exp^{P-\mathsf{UNT}}_{\mathcal{S},\mathcal{A}}$ (with $P \in \{\mathsf{Existential}, \mathsf{Forward}\}$), then it is possible to construct an adversary $\mathcal{A}'$ who wins the IND-CPA game. To do so, $\mathcal{A}'$ runs $\mathcal{A}$ as subroutine, simulating the system $\mathcal{S}$ to $\mathcal{A}$ by answering all oracles queries made by $\mathcal{A}$. At the end of the IND-CPA game, $\mathcal{A}'$ answers what $\mathcal{A}$ answers for $Exp^{P-\mathsf{UNT}}_{\mathcal{S},\mathcal{A}}$. Here, $\mathcal{A}'$ knows the secrets of $\mathcal{T}_0$ and $\mathcal{T}_1$ at the beginning of the IND-CPA game,

[6]Message Authentication Code.

in order to perform it. When $\mathcal{A}$ asks the interactions for $\mathcal{T}_0$ and $\mathcal{T}_1$, $\mathcal{A}'$ answers the corresponding ciphertexts for these interactions using the correct plaintext. When $\mathcal{A}$ asks the interactions for $\mathcal{T}$, then $\mathcal{A}'$ submits the plaintexts for both $\mathcal{T}_0$ and $\mathcal{T}_1$ for these interactions to the IND-CPA challenger $\mathcal{C}'$. $\mathcal{A}'$ receives the ciphertexts answered by $\mathcal{C}'$ for $\mathcal{T}_b$, where b is the unknown bit of the IND-CPA experiment, and transfers them to $\mathcal{A}$. So far, the simulation done by $\mathcal{A}'$ to $\mathcal{A}$ is perfect. Then, two cases can occur.

1. $\mathcal{A}$ does not need $\mathcal{T}$'s secrets (i.e., $\mathcal{A}$ is playing the Existential experiment). $\mathcal{A}$ wins $Exp_{\mathcal{S},\mathcal{A}}^{\mathsf{Existential-UNT}}$, thus her advantage is non negligible, so is the advantage of $\mathcal{A}'$.

2. $\mathcal{A}$ asks $\mathcal{T}$'s secrets (i.e., $\mathcal{A}$ is playing the Forward experiment). $\mathcal{A}'$ does not know b, thus she sends at random $\mathcal{T}_0$'s or $\mathcal{T}_1$'s secrets. If $\mathcal{A}'$ sends the expected ones, then $\mathcal{A}$ wins $Exp_{\mathcal{S},\mathcal{A}}^{\mathsf{Forward-UNT}}$, thus her advantage is non negligible, so is the advantage of $\mathcal{A}'$. If not, at worst $\mathcal{A}$ answers at random 0 or 1. Therefore, the whole advantage of $\mathcal{A}$ is non negligible, so is the advantage of $\mathcal{A}'$.

Consequently, $\mathcal{A}'$ is an adversary who wins the IND-CPA game with non negligible advantage, which concludes the proof. □

Vaudenay proves in [162] that the protocol is NARROW-STRONG-private with IND-CPA security, and that it is furthermore FORWARD-private with IND-CCA. Since the privacy notions of JW are included in Vaudenay (as explained in Section 4.3), the protocol is thus forward-(ρ, σ, τ)-private for JW. HPVP proves in [82] that the protocol is also NARROW-STRONG-private with IND-CPA security, but that it is STRONG-private with IND-CCA.

In the LBM model, if an environment is able to distinguish the real world from the ideal one, it can easily be transformed into a distinguisher of the IND-CCA property of the underlying cryptosystem. Thus it is obvious that this protocol is forward-secure.

In the CCEG model, the protocol is future-untraceable with IND-CCA security (proved in [34]). In the DLYZ model, the protocol is also backward-ZK-private with IND-CCA security: the proof follows the same reasoning as the one of CCEG.

Table 4.5 sums up the privacy analysis of PK-Prot.

Model	PK-Prot
Avoine	Existential-UNT-RTE* + Forward-UNT-RTEC*
JW	Forward-(ρ, σ, τ)-privacy
Vaudenay	NARROW-STRONG-privacy* + FORWARD-privacy
LBM	Forward-security
DMR	Untraceability*
CCEG	Future-untraceability
DLYZ	Backward-ZK-privacy
HPVP	NARROW-STRONG-privacy* + STRONG-privacy

Table 4.5: Summary of the PK-Prot analysis. "✗" means no privacy. A property followed by "*" means that it is at least achieved with IND-CPA security

4.1.6 Analysis Comparison

Table 4.6 sums up the security analysis of the studied protocols regarding each privacy model.

The Lack of Comprehensiveness. In many models, several protocols are proved to ensure the same privacy level. The most glaring examples in our study are (i) SK-Prot, MW-Prot, and O-FRAP in the Vaudenay, CCEG, and HPVP models, (ii) SK-Prot, OSK-Prot, O-FRAP in the DMR model, and (iii) OSK-Prot, O-FRAP and PK-Prot in the Avoine model. However, if we explore the last example of the Avoine model, OSK-Prot, O-FRAP and PK-Prot are not threatened by the same kind of attack. As detailed in Section 4.1.3, OSK-Prot can be desynchronized contrary to the two others, and O-FRAP is subject to a specific attack based on tag corruption (see Section 4.1.4), while PK-Prot is not vulnerable to such attacks.

This is due to the evolution of the privacy concept: at their conception, not all the models were designed to formalize all the potential privacy attacks. This fact may affect the implementation choices made by a system designer unfamiliar with privacy. The latter will probably choose the cheapest protocol (regarding computing complexity) among a set of protocols proved with the same privacy level within a given model. Yet, such protocols may be built with important characteristics that

Table 4.6: Analysis summary of the protocols. "✗" means no privacy. For PK-Prot, a property followed by "*" means that it is at least achieved with IND-CPA security. For the sake of clarity, "N" and "DESTR" respectively denote "NARROW" and "DESTRUCTIVE".

Model \ Protocol	SK-Prot	MW-Prot	OSK-Prot	O-FRAP	PK-Prot
Avoine	Existential-UNT-RTE	Existential-UNT-RTE	Existential-UNT-RTE Forward-UNT-RTEC	Existential-UNT-RTE Forward-UNT-RTEC	Existential-UNT-RTE* Forward-UNT-RTEC*
JW	(ρ, σ, τ)-privacy	✗	✗	(ρ, σ, τ)-privacy	Forward-(ρ, σ, τ)-privacy
Vaudenay	WEAK-privacy	WEAK-privacy	N-DESTR-privacy	WEAK-privacy	N-STRONG-privacy* FORWARD-privacy
LBM	✗	✗	✗	Forward-security	Forward-security
DMR	Untraceability	✗	Untraceability	Untraceability	Untraceability*
CCEG	Standard-untraceability	Standard-untraceability	✗	Standard-untraceability	Future-untraceability
DLYZ	ZK-privacy	✗	✗	ZK-privacy	Backward-ZK-privacy
HPVP	WEAK-privacy	WEAK-privacy	N-DESTR-privacy	WEAK-privacy	N-STRONG-privacy* STRONG-privacy

clearly distinguish them (e.g., different cryptographic building blocks) and that are not taken into account by the model.

The Case of Correlated Secrets. The JW, DMR and DLYZ models are able to point out the weaknesses of protocols based on correlated secrets by proving that MW-Prot is not secure, while SK-Prot is. This comes from the fact that an adversary may know some secrets without being authorized to corrupt the challenge tags (as explained in Section 4.1.2). For instance, this adversary could be a tag owner who only knows the secrets of her tag, and who is not able to corrupt other tags that she wants to trace. It is consequently normal that SK-Prot ensures a higher privacy than MW-Prot. Note that this differentiation cannot be established in the Avoine, Vaudenay, CCEG and HPVP models because their adversary does not have the modularity to only corrupt some particular tags. As a result, these models classify SK-Prot and MW-Prot with the same privacy level.

The Key-update Mechanism Dilemma. All the models (except Avoine and LBM) give the same privacy level for SK-Prot and for O-FRAP. This is another example about the evolution of the privacy concept in the models. These two protocols do not manage the tag secrets in the same way: a tag updates one of its secrets each time it starts an execution of O-FRAP, while a tag always keeps the same secret when it runs SK-Prot. For O-FRAP, the attack presented in Section 4.1.4 only permits linking a freshly corrupted tag to its last previous incomplete protocol execution. But all the previous completed ones are unlinkable. This is not the case with SK-Prot, where a tag corruption allows tracing the tag unconditionally. This distinction of the two protocols cannot however be performed in most of the models.

Accuracy Refinement of the NARROW Adversary. The NARROW nuance provided in some models permits granting some protocols with a reasonable privacy level. For instance, Vaudenay and HPVP confer NARROW-DESTRUCTIVE-privacy on OSK-Prot and NARROW-STRONG-privacy on PK-Prot with IND-CPA security, while some other models argue that OSK-Prot ensures no privacy at all, or that PK-Prot with IND-CPA security cannot be proved private. These last claims are

highly restrictive since these two protocols are clearly more private than the dummy protocol where tags send their identifier in clear.

4.2 Classification of the Models

This section compares the different features of all the privacy models presented in Chapter 3 of this thesis. It classifies the models according to the expected features by pointing out the most appropriate one(s) to use in each specific attack scenario.

Table 4.7 sums up the features that are achieved by each model. Note that "protocols" (resp. "tag-init protocols") refer to authentication/identification protocols where the reader (resp. tag) is the only entity that can start a protocol execution.

4.2.1 Adversary Experiment

Privacy models can be compared according to the similarities and differences of their experiment. To do so, it is first needed to define the notion of *challenge tags* in some models. Indeed, Vaudenay, LBM, DMR, CCEG and HPVP do not stipulate this specific notion in their experiment. However, since their adversary must use some tags for her attack, all the created tags of the system are considered as challenge ones. Note that the agents that can be corrupted before $\mathcal{A}$'s attack in the DMR model are considered as *non-challenge tags*.

Number of Tags Allowed in the Experiment. Vaudenay, LBM and CCEG are the only models where the adversary $\mathcal{A}$ is free to play with all the tags of the system at the same time during her attack.

At one moment of their experiment, the adversaries of JW and HPVP can only play with at most $(n-1)$ tags (where n is the total number of tags of the studied system). For the DLYZ model, the adversary cannot play with the set of clean tags she chose, except with the challenge tag $\mathcal{T}_c$ picked at random in this set. If this set contains only two tags, she can however play with at most $(n-1)$ tags. Then, the DMR adversary cannot play with the agents that were corrupted before the beginning of her attack. Finally, the Avoine model is the most limiting one, since $\mathcal{A}$ can only play with two tags. This fact prevents the Avoine model from

Table 4.7: Comparison of the presented privacy models. "✓" (resp. "✗") means that the feature is (resp. is not) given to the adversary $\mathcal{A}$. "N/A" means that the feature is not applicable in the model.

Feature \ Model	Avoine	JW	Vaudenay	LBM	DMR	CCEG	DLYZ	HPVP
Interaction with all the tags	only 2 tags	not $\mathcal{T}^*_{b\oplus 1}$	✓	✓	not corrupted agents	✓	not all clean tags	all-but-one
Choice of the challenge tags	✗	✓	✓	✓	✓	✓	✓	✓
Attack on incomplete executions	both	✓	✓	✗	✓	✓	✓	✓
$\mathcal{O}^{\text{Corrupt}}$ challenge tags	only $\mathcal{T}$	only $\mathcal{T}^*_b$	✓	✓	✗	✓	✗	✓
$\mathcal{O}^{\text{Corrupt}}$ non-challenge tags	N/A	✓	N/A	N/A	✓	N/A	✓	N/A
$\mathcal{O}^{\text{Corrupt}}$ any tag	✗	✗	✓	✓	✗	✓	✗	✓
NARROW/WIDE	NARROW	WIDE	both	WIDE	NARROW	WIDE	WIDE	both
Channels asymmetry	✓	✗	✗	✗	✗	✗	✗	✗
Protocols analyzable	3-pass with independent secrets	symmetric-key based	all	all	all	all	$(2\gamma+1)$-pass	all
Tag-init protocols analyzable	✗	✓	✗	✗	✓	✗	✗	✗

analyzing protocols with correlated secrets, which is not the case for all the other models.

Therefore, if $\mathcal{A}$ is allowed to play with all the tags of the system, then it is preferable to use the Vaudenay, LBM and CCEG models for the privacy analysis.

Choice of the Challenge Tags. All the models (except Avoine) allow $\mathcal{A}$ to choose the challenge tags of her attack. In the Avoine model, the challenger $\mathcal{C}$ is the entity that performs this task, choosing $\mathcal{T}$, $\mathcal{T}_0$ and $\mathcal{T}_1$ (such that $\mathcal{T} = \mathcal{T}_0$ or $\mathcal{T}_1$). $\mathcal{A}$ has no option on the tags used for her attack: she is weaker than the adversaries of the other models. Thus, if it is considered that $\mathcal{A}$ has the possibility to choose the challenge tags, protocols should be analyzed with all the models except the Avoine one.

Attack on Incomplete Protocol Executions. In the JW, Vaudenay, DMR, CCEG, DLYZ and HPVP models, $\mathcal{A}$ is allowed to perform her attack on incomplete protocol executions. As illustrated in Section 4.1.4, she can start an execution with a tag and not finish it. Afterward, she can use this tag during her game to break its privacy. If $\mathcal{A}$ succeeds to do so, then the protocol is not considered as private.

For LBM, such an attack is not taken into account. $\mathcal{F}_{aauth}$ is designed such that all the successfully completed protocol executions of a tag are protected against corruption. In other words, $\mathcal{A}$ cannot learn any information about these previous executions, and thus the privacy of a tag is ensured. However, she is authorized to link the previous incomplete executions of a corrupted tag up to the last completed one without compromising the security.

For the Avoine model, both scenarios are allowed. During the Existential game, $\mathcal{A}$ chooses the intervals I_0 and I_1 of the challenge tags that help her the most to perform her attack. She can choose I_0 and I_1 such that these intervals are directly consecutive to I (the interval of the targeted tag $\mathcal{T}$). In that case, nothing prevents $\mathcal{A}$ from using incomplete protocol executions during the experiment. For the Universal game, the challenger $\mathcal{C}$ is the one who chooses I_0 and I_1 that help $\mathcal{A}$ the less, contrary to the Existential game. If $\mathcal{A}$ uses incomplete protocol executions, then $\mathcal{C}$ can choose non consecutive intervals such that the incomplete executions remain meaningless to $\mathcal{A}$ (as for LBM). For instance, some completed

executions may separate the executions (completed or not) performed within the intervals.

Therefore, if a protocol must be protected against this attack, then Avoine, JW, Vaudenay, DMR, CCEG, DLYZ, HPVP are the most appropriate models to study its privacy. If such a feature is not wished, then it can be analyzed with the Avoine and LBM models. Note that the Avoine model is the most flexible one since it can handle both scenarios.

4.2.2 Tag Corruption

The tamper-resistance of RFID tags is a highly questionable assumption. Fortunately, all the models are flexible regarding the capacity of an adversary to corrupt tags. The two extreme cases are the impossibility to corrupt tags or the possibility to perform this action without restrictions. Yet, as detailed in the previous sections, intermediate levels of corruption have been introduced. To have an overall view of these levels, the models are gathered below based on their similarities from the weakest corruption level to the strongest one.

Weak Adversary. Obviously the weakest corruption level is when $\mathcal{A}$ is not allowed to corrupt tags. This feature is present in the Avoine, Vaudenay, LBM, CCEG and HPVP models. It permits formalizing the assumption of tags tamper-resistance.

Although the JW, DMR and DLYZ models consider that it is always possible to corrupt non-challenge tags, they also define a weak level of corruption where $\mathcal{A}$ is not able to corrupt the challenge tags. This adversary, called *insider adversary* in [46], may be a tag owner who only knows the secrets of her tag, and who wants to break the privacy of other tags. As explained in Section 4.1.2 and in Section 4.1.6, this subtle adversary can be used to perform a dedicated attack on a system with correlated secrets. However, even if this attack can be caught in other models by an over-powerful adversary (e.g., the Vaudenay FORWARD adversary), the Vaudenay, LBM, CCEG and HPVP models are unable to precisely formalize such an intermediate adversary, since these models allow $\mathcal{A}$ to corrupt either every tag or no tag at all.

Consequently on the one hand, if it is assumed that $\mathcal{A}$ can never corrupt a tag, then the Avoine, Vaudenay, LBM, CCEG and HPVP

models should be chosen for a protocol analysis. On the other hand, if it is assumed that only the non-challenge tags can be corrupted, then the most appropriate and fair models to use are JW, DMR and DLYZ.

Non-adaptive Adversary. A higher level of corruption consists in authorizing $\mathcal{A}$ to only corrupt tags at the end of the experiment. It corresponds to the FORWARD adversary of Vaudenay and HPVP, and to the Forward-UNT property of Avoine. It can be viewed as a non-adaptive corruption ability as $\mathcal{A}$ cannot adapt her attack according to the result of a corruption.

The forward-ZK-privacy of DLYZ is close to this property since the last key of the challenge tag is given to the distinguisher at the end of the experiment. Yet in this case, $\mathcal{A}$ is still allowed to adaptively corrupt the non-challenge tags during the experiment without stopping it. This fact slightly increases the strength of the DLYZ adversary.

Destructive Adversary. To increase the adversary power, some models give $\mathcal{A}$ the ability to pursue her attack after a corruption, leading to adaptive attacks regarding corruption. However, some constraints are still put into place in some models. In fact, the JW model considers that the challenge tags may be corrupted in the forward-(ρ, σ, τ)-privacy, but only during the challenge phase. In other words, a tag corruption can only be used to trace its previous interactions. It is thus possible to establish a parallel between this constraint and the destructive corruption ability defined in other models (namely the DESTRUCTIVE adversary of Vaudenay, CCEG and HPVP, and the forward-security of LBM). Indeed, the key material obtained through a tag corruption may allow tracing its previous interactions but not the future ones as the tag is destroyed.

Strong Adversary. The strongest level that can be defined is obviously when $\mathcal{A}$ has no restriction regarding tag corruption. This corresponds to the STRONG adversary defined in the Vaudenay, CCEG and HPVP models. A relatively similar notion is also defined by DLYZ, namely the backward-ZK-privacy. However, as for the forward-ZK-privacy, while every non-challenge tag may be corrupted during the experiment, the challenge tag cannot, and its initial key is only revealed at the end of the experiment. It may still help to distinguish the following interactions

of this tag, but $\mathcal{A}$ cannot adapt her attack to this result. This consequently leads to a non-adaptive adversary that may be useful in some cases. Yet, one may prefer the Vaudenay, CCEG and HPVP models to catch the strongest adversary definition regarding corruption ability.

As a conclusion, the Vaudenay, CCEG and HPVP models offer a wider adversary granularity regarding tag corruption[7]. Only these three models take into account the strongest adversary which can corrupt with no restriction. Still, they do not consider the insider adversary who represents a relevant assumption and affords, to our mind, an interesting granularity for some analyses. In this case, protocols may thus be studied with a more appropriate model, namely either JW, or DMR, or DLYZ.

4.2.3 Other Features

The remaining features of Table 4.7 are discussed in the following.

NARROW/WIDE Adversaries. As previously said, an adversary $\mathcal{A}$ is said to be NARROW (resp. WIDE) when she does not (resp. does) receive the result of a protocol execution. Several models restrict their adversary with one of these features.

Avoine does not define a $\mathcal{O}^{\text{Result}}$ oracle, and there is no equivalence of such an oracle in DMR (since $\mathcal{A}$ does not know if a protocol between two agents succeeds). Both models only consider NARROW adversaries.

On the contrary, the adversaries of JW, LBM, CCEG and DLYZ are only WIDE ones. For JW, there is no $\mathcal{O}^{\text{Result}}$ oracle defined in the model, but the adversary is forced to obtain the result of a protocol execution via the output of each $\mathcal{O}^{\text{SendReader}}$. The DLYZ adversary has the same behavior: she is forced to know this result information since $o_{\mathcal{R}}^{sid}$ and $o_{\mathcal{T}}^{sid}$ are public. In the LBM model, the output tape of each party is always available to $\mathcal{Z}$. Additionally, the adversary may also learn it as $\mathcal{Z}$ can communicate arbitrarily with her. Thus, it is impossible to model a NARROW adversary since the distinguisher may always know the result of a protocol execution. For CCEG, no NARROW adversary can be used

[7]Note that the CCEG authors consider that FORWARD and DESTRUCTIVE adversaries (in Vaudenay's sense) are equivalent in their experiment: both are able to output a *standard* or *past* NOL, but not a *future* NOL. Therefore, a FORWARD adversary is useless in their model.

in the experiment. Yet, as stressed in the analysis of OSK-Prot given in Section 4.1.3, this voluntary restriction implies that this family of protocols with decent security features are not considered private.

The Vaudenay and HPVP models are the most flexible ones since it is possible to choose either a NARROW or a WIDE adversary. Note that the other models can however be (more or less easily) adapted to provide both adversary classes.

Channels Asymmetry. As already explained in Section 3.2, the forward channel (reader to tag) has a longer communication range than the backward channel (tag to reader). This characteristic is of interest as it has been shown in [80, 160] that the former can be more easily eavesdropped than the latter in practice. This asymmetry is exploited for example by Molnar and Wagner in [117] in order to design an authentication protocol suitable to libraries. Yet, the Avoine model is the only one that formalizes this feature through the $\mathcal{O}^{\text{Execute}^*}$ oracle: $\mathcal{A}$ may only obtain the messages sent by $\mathcal{R}$ on the forward channel.

All the other models (as a matter of fact, created after the Avoine one) lost this feature, and cannot represent this kind of weaker but realistic adversary. The analysis must thus be performed with the Avoine model when it is assumed that $\mathcal{A}$ is only able to get the messages sent from $\mathcal{R}$.

Analyzable Protocols. Some models are designed "by default" to analyze specific identification/authentication protocols. In the Avoine model, the oracles to interact with the system can only be used for 3-pass protocols. Then, the authors of JW only aim to analyze protocols based on symmetric-key cryptography. Finally, DLYZ can only analyze $(2\gamma + 1)$-pass protocols with $\gamma \geq 1$.

On the contrary, Vaudenay, LBM, DMR, CCEG and HPVP can analyze any identification/authentication protocol. Some of the restrictive models can nevertheless be adapted to analyze most existing protocols. For instance, the Avoine model can be slightly modified to analyze 2-pass classical challenge/response protocols, and the JW model does not forbid the analysis of protocols with public-key cryptography.

Finally, JW and DMR are the only models that are not restricted by default to analyze protocols where the reader starts an execution.

4.3 Privacy Properties

The previous section discussed the features that are present (or not) in each of the studied models. This section goes a step further in the investigation and compares their privacy properties.

This task is not an easy one as the different features of each model make it tough to compare them in some cases. Indeed in the sequel, we highlight the fact that, when a privacy property of a given model is said "stronger" than the one of another model, the "weaker" model may present some additional features that are not present in the "stronger" one. We assume that a system designer is aware of this fact and that, in this special case, he may thus prefer to use the weaker model for his privacy analysis. Except when this fact must be highlighted, we will not detail it in each comparison.

4.3.1 Indistinguishability of Tags

Regarding only the privacy notions, the Avoine and JW models are really close. Indeed, they both define privacy as the unfeasibility for an adversary to recognize one tag among two. The JW model has been designed after the Avoine one, as an improved model since it takes into account several flaws of the Avoine model. It can be easily proved that the JW (ρ, σ, τ)-privacy (resp. forward-(ρ, σ, τ)-privacy) implies the Avoine Existential-UNT (resp. Forward-UNT): the goal is the same and any request of an Avoine adversary can be performed by a JW adversary.

In the DMR model, the privacy property corresponds to the unfeasibility to link two *traces* that are produced by the same agent (in our case, a tag). This notion is also really close to the one defined in the JW model. Clearly for JW, the adversary capacity to retrieve the tag associated to the bit b permits her to link two traces, and reciprocally. However, as the DMR model only defines a non-adaptive adversary regarding corruption, the JW (ρ, σ, τ)-privacy is obviously stronger than the DMR untraceability.

Largely inspired by the design of the Vaudenay model (on which we will come back later), the CCEG and HPVP models offer a comprehensive list of oracles that permit any JW adversary to be represented in their models. Regarding the privacy definition, it is obvious that the output of a JW adversary is exactly a CCEG non-obvious link (*stan-*

dard or *past*) and can thus be directly exploited by a CCEG adversary. As a consequence, the CCEG standard-untraceability (resp. past-untraceability) property obviously implies the JW (ρ, σ, τ)-privacy (resp. forward-(ρ, σ, τ)-privacy). The reciprocal does not lead to a tight reduction. Indeed, a CCEG adversary may shuffle the pseudonyms of tags several times (by performing successive $\mathcal{O}^{\text{DrawTag}}$ and $\mathcal{O}^{\text{Free}}$ queries), which is hard to simulate in the JW model.

The HPVP model defines privacy using the well-known "left-or-right" paradigm. As detailed in Section 3.9, it splits the tags space into two worlds. Nevertheless, a JW adversary can be simulated in this model. Firstly, the HPVP adversary draws each tag of the system[8]. Then, the two selected challenge tags of JW are freed, and given as input of the $\mathcal{O}^{\text{DrawTag}}$ oracle. If the JW adversary is able to recognize the outputted tag, then she may be used by an HPVP adversary to output the guessed bit. The reciprocal is not true for the same reasons as for CCEG.

As a conclusion, assuming that privacy is defined as indistinguishability of tags, the most comprehensive models are HPVP and CCEG. Intuitively, these two models have equivalent privacy notions. Indeed, an adversary who succeeds in the HPVP experiment can easily output a non-obvious link. On the opposite, a non-obvious link permits the distinction of one tag from the others and can thus be used in the "left-or-right" paradigm. However, it is not obvious to formally prove this equivalence result due to the following facts. Firstly, at one moment of the HPVP experiment, the adversary must use (at least once) the $\mathcal{O}^{\text{DrawTag}}$ oracle on two different tags in order to obtain information about the challenge bit. At that moment, this adversary can no longer interact with all the tags whereas a CCEG adversary can always interact with all the tags if she wants to. Secondly, a CCEG adversary may draw more than one tag in a $\mathcal{O}^{\text{DrawTag}}$ query (e.g., three tags out of four). If an HPVP adversary wants to use such an adversary as a subroutine to succeed in the HPVP experiment, the simulation of this fact entails that some choices (for the CCEG adversary) are mandatory and thus leads to a non-tight reduction.

[8] A single tag can be given as the two inputs of the $\mathcal{O}^{\text{DrawTag}}$ oracle.

4.3.2 Real World vs. Simulated World

The last three models (i.e., Vaudenay, LBM and DLYZ) define privacy as, in a nutshell, the unfeasibility to distinguish the interactions of an adversary against the real system from the interactions of a simulated adversary against a simulated world. In this second world, the simulator does not know the keys of the system to build some of the exchanged messages. Nevertheless, when a tag corruption is asked, its real secret key is returned. The idea behind this privacy notion is that, if there exists a distinction between these two worlds, then some information must leak from the messages of the real world (which are built with the real keys of the system).

The most adaptive and comprehensive model using this principle is clearly the Vaudenay model. First, this model offers the widest range of adversaries. Then, these adversaries can be adaptive, contrary to the ones of DLYZ. Finally, as explained in Section 4.2.1, the LBM model only ensures the privacy of authentications prior to the last complete one, while the Vaudenay model considers privacy of all the possible authentications. As a consequence, for equivalent adversary classes, the Vaudenay model is stronger than LBM and DLYZ.

From another point of view, the UC framework is generally used to analyze protocols that are not run alone, but in parallel/concurrency with other protocols. Here, the interesting feature is that the environment $\mathcal{Z}$ can interact with the system and thus may help $\mathcal{A}$ to perform her attack, while the Vaudenay adversary is on her own. This fact has been frequently used in the UC literature (e.g., [36, 37]) to prove that some "considered secure" constructions are indeed not. As a consequence, if the protocol to analyze is designed to belong to a complex system, its privacy may be studied in the LBM model. Nevertheless, if a strong privacy property is wished, the protocol should also be analyzed in the Vaudenay model.

4.3.3 Between the Two Families

Several points of comparison can be established between the most comprehensive models of these two families (namely CCEG and HPVP for the indistinguishability family, and Vaudenay for the 2-world family).

CCEG vs. Vaudenay. The oracles description of the CCEG model is really close to the one of Vaudenay. The authors of the former describe their model as a restriction of the Vaudenay one, mainly on the experiment. Indeed, the CCEG adversary is required to output a non-obvious link, while any adversary assumption can be outputted in the Vaudenay model. Consequently, the CCEG privacy notion is intuitively weaker than the Vaudenay one (for equivalent adversaries). Nevertheless, as proved in [34], the CCEG future-untraceability is a reachable property while the Vaudenay STRONG-privacy is impossible. Furthermore, to increase their result, the authors of CCEG also proved with a "toy scheme" that their future-untraceability considers attacks that are not taken into account in the two "highest" reachable privacy levels of Vaudenay (namely the NARROW-STRONG and DESTRUCTIVE-privacy). As a consequence, the CCEG model defines a potentially weaker privacy notion, but, under this framework, the privacy of a protocol can be studied against a stronger adversary than in the Vaudenay model.

HPVP vs. Vaudenay. Similar results may be proved for the HPVP model. First, its authors exhibited in their paper a protocol that ensures STRONG-privacy in their model. Then, using the "toy scheme" defined in [34], it can be proved that the same attacks (highlighted by CCEG) are also taken into account in the HPVP STRONG-privacy, which are not considered in the reachable privacy levels of Vaudenay. However, as for the CCEG model, it can be proved that the privacy of a slightly modified version of the Vaudenay model (called "modified Vaudenay model" in what follows) implies the HPVP one for equivalent adversary classes. As this final result is not intuitive, we prove it with the following theorem.

Theorem 4.1. *For any adversary class P such that*

$$P \in \{\emptyset, \mathit{NARROW}\} \times \{\mathit{WEAK}, \mathit{FORWARD}, \mathit{DESTRUCTIVE}, \mathit{STRONG}\},$$

the P-privacy property of the modified Vaudenay model implies the P-privacy property of the HPVP model.

Proof. Both models define the same adversary classes, but differ in their experiment. However, we show here that, for a given class P, the modified Vaudenay P-privacy implies the HPVP one. To do so, we exhibit an

adversary in the modified Vaudenay model, denoted $\mathcal{A}_{\mathsf{Vaud}}$, that emulates the system to an adversary playing the HPVP P-experiment, denoted $\mathcal{A}_{\mathsf{HPVP}}$, and uses the output of the latter to break the modified Vaudenay P-privacy.

First, $\mathcal{A}_{\mathsf{Vaud}}$ can answer all the possible queries performed by $\mathcal{A}_{\mathsf{HPVP}}$ during her experiment. The $\mathcal{O}^{\textsc{Launch}}$, $\mathcal{O}^{\textsc{SendTag}}$, $\mathcal{O}^{\textsc{SendReader}}$, $\mathcal{O}^{\textsc{Result}}$, and $\mathcal{O}^{\textsc{CreateTag}}$ queries can be easily emulated by $\mathcal{A}_{\mathsf{Vaud}}$ due to their large similarity. For the $\mathcal{O}^{\textsc{DrawTag}}$ oracle, the original Vaudenay model is slightly modified in order to emulate the one of HPVP. Indeed in HPVP, this oracle formalizes the "left-or-right" paradigm. To handle this issue, we assume that, when $\mathcal{A}_{\mathsf{Vaud}}$ gives as input of $\mathcal{O}^{\textsc{DrawTag}}$ a probability distribution with the form "$\Pr[\mathsf{ID}_i] = 1/2, \Pr[\mathsf{ID}_j] = 1/2$", then this also follows the "left-or-right" paradigm as well.

Also, $\mathcal{A}_{\mathsf{HPVP}}$ can only corrupt free tags while only drawn tags can be corrupted in the Vaudenay model. Nevertheless, $\mathcal{A}_{\mathsf{Vaud}}$ can correctly reply to these queries: upon a corruption query of the tag $\mathcal{T}$, $\mathcal{A}_{\mathsf{Vaud}}$ draws $\mathcal{T}$ using a special distribution probability which attribute a probability of 1 to $\mathcal{T}$ and 0 for all the other tags. Then, she can corrupt it, transmits the data to $\mathcal{A}_{\mathsf{HPVP}}$, and then frees $\mathcal{T}$. This method correctly works for DESTRUCTIVE and STRONG adversaries (and their NARROW variants). However, it must be adapted for a FORWARD adversary. Indeed, in both models, such an adversary can only perform corrupt queries after that the first one has been made, and $\mathcal{A}_{\mathsf{Vaud}}$ must anticipate all these possible queries of $\mathcal{A}_{\mathsf{HPVP}}$. Thus, upon the first corruption query, $\mathcal{A}_{\mathsf{Vaud}}$ first frees all tags, and then draws them one by one in order to know the correspondences between all the tags identifiers and their pseudonyms. Finally, $\mathcal{A}_{\mathsf{Vaud}}$ is able to reply to all the corruption queries correctly.

This simulation is perfect and cannot be detected by $\mathcal{A}_{\mathsf{HPVP}}$ who, as a consequence, will output her guessed bit b' with her habitual probability. Then, using this bit, $\mathcal{A}_{\mathsf{Vaud}}$ can decide which tag has been drawn by the $\mathcal{O}^{\textsc{DrawTag}}$ queries. Therefore, the success probability of $\mathcal{A}_{\mathsf{Vaud}}$ is exactly the one of $\mathcal{A}_{\mathsf{HPVP}}$. As the Vaudenay blinder cannot decide in advance which tag should be simulated after a $\mathcal{O}^{\textsc{DrawTag}}$, the success probability of this blinded adversary is necessary one half (random guess of the bit).

Thus, if there exists an attack for a given system against the P-privacy in HPVP, then there exists an attack against the P-privacy in the modified Vaudenay model that succeeds with the same probability.

Therefore, for any class P, the modified Vaudenay P-privacy implies the HPVP one. □

The reciprocal is hard to prove for two main reasons. Firstly, the Vaudenay experiment output is not specified and may thus be unexploitable by $\mathcal{A}_{\mathsf{HPVP}}$. Secondly, the $\mathcal{O}^{\text{DrawTag}}$ oracle may receive as input an arbitrary distribution that can be hard to simulate using the "left-or-right" $\mathcal{O}^{\text{DrawTag}}$ of HPVP.

Note on DLYZ and JW. To conclude this discussion, we highlight some existing results about the DLYZ model. The authors of this model argue that the JW (ρ, σ, τ)-privacy does not imply their ZK-privacy and used several schemes to illustrate their claim.

One example is a system composed of only one tag. Clearly, such a system cannot be analyzed in the JW model since it requires at least two tags in the experiment. Thus, their claim that the proposed single-tag system is (ρ, σ, τ)-private is doubtful and inadequate. Furthermore, in such a special case of single-tag systems, the authors of DLYZ say that ZK-privacy is reduced to the basic zero-knowledge definition which, according to them, provides a reasonable privacy. However in practice, each time this single tag is accepted by a reader, a WIDE adversary is obviously able to link this authentication to the previous ones. To our mind this is obviously a breach of privacy.

One other example is a system composed of one reader $\mathcal{R}$ and many tags where they all share a unique pair of public/private keys. In a nutshell, $\mathcal{R}$ and a tag $\mathcal{T}$ perform the following protocol to communicate with each other: $\mathcal{R}$ sends the cipher of a random value obtained with the public key; the tag deciphers the message and sends the recovered value back to $\mathcal{R}$. The argument claiming that this system is not ZK-private under the IND-CPA secure assumption is not considered as acceptable according to the authors of [118]. They show that there indeed exists a simulator *Sim* able to output an indistinguishable view from $\mathcal{A}$'s view at the end of the privacy analysis.

Finally, the authors of [118] go one step beyond and formally prove that the JW (ρ, σ, τ)-privacy is equivalent to the DLYZ ZK-privacy (Theorem 1 of [118]).

4.4 Summary of the Study

First of all, the study presented in this chapter reveals that none of the existing models encompasses all the others. The main reason is that no model offers enough granularity to provide all the features detailed previously. Even if it is sometime possible to extend an existing model to take into account a new property or a new assumption, it is not always a trivial task to add all of them.

Throughout the study, it appears that the Vaudenay model is the one that integrates the greatest number of features and defines the strongest privacy notion. As a default choice, the Vaudenay model is probably the best one. Nevertheless, some drawbacks have been highlighted. Firstly, the strongest privacy property of this model cannot be ensured by any protocol. To analyze the privacy of a protocol against the strongest (known) adversary, one may thus prefer the CCEG of the HPVP model. Secondly, the Vaudenay model (as other ones) considers that tracing a tag after an incomplete protocol execution compromises the privacy. On the one hand, this is a relevant consideration that ensures a strong privacy level. On the other hand, relaxing this constraint helps to design more efficient protocols with a still reasonable privacy level using the Avoine and LBM models. Finally, the lack of granularity of all the models implies difficulties to fairly distinguish, in a given model, protocols with different security levels.

If a system designer has precisely defined the requested properties of his application and the assumptions regarding potential adversaries, then he might use the results of this chapter to select the most appropriate model. Thereby, he can design or select the most adapted and efficient protocol for his needs. In Chapter 7, we will propose a new model that aims to unify and simplify the existing ones in order to help the community to compare protocols meaningfully.

Chapter 5

Time Attacks Threatens Privacy-Friendly Systems

In RFID systems where each tag is associated to a unique secret, the reader $\mathcal{R}$ generally has to retrieve the corresponding secret in its database to authenticate a given tag. To ensure a minimal level of privacy, tags should send a randomized value, and $\mathcal{R}$ should perform a SearchID procedure on this value to retrieve the corresponding secret. The simplest way to carry out this task is with a linear exhaustive search in the whole reader database. If the SearchID procedure is deterministic, the time spent by $\mathcal{R}$ to authenticate a given tag is always the same for every protocol execution. Consequently, $\mathcal{A}$ can deduce which tag is authenticated by simply measuring this time. This time information is clearly an important privacy issue, but it has not yet been included in any existing RFID privacy model.

In this chapter, we first add this *time attack* to the Vaudenay model through the formalization of a new privacy level called TIMEFUL. Then, we display the weaknesses of several existing protocols when facing this TIMEFUL adversary: we demonstrate that OSK-Prot is not TIMEFUL-NARROW-DESTRUCTIVE-private, and that neither SK-Prot nor undesynchronizable protocols like O-FRAP are TIMEFUL-WEAK-private. Finally, we propose various solutions to ensure TIMEFUL-privacy. They consist in combining an appropriate choice for the reader database structure with a pertinent SearchID procedure: our approaches are based on rainbow tables, hash tables, B-trees, and random search.

5.1 The Modified Vaudenay Privacy Model

In the previous chapter, the Vaudenay model [162] has been demonstrated to be the most comprehensive one so far. This section presents a revision of this model that allows the formalization of time attacks on identification/authentication protocols.

5.1.1 Definition of the Oracles

An adversary $\mathcal{A}$ is allowed to query all the oracles defined in the Vaudenay model as presented in Section 3.4.

Furthermore, she is allowed to ask the time spent by the reader to compute all the operations and to perform its SearchID procedure in order to identify and/or authenticate the tag linked to a particular protocol execution.

- $\mathcal{O}^{\text{Timer}}(\pi) \rightarrow \mathsf{t}$: outputs the time t taken by the reader for its overall computations during the protocol execution π.

5.1.2 Definition of the Adversary

From this oracle, a new class of adversaries can be defined as follows.

Definition 5.1 (Adversary Class). *An adversary class is said to be* TIMEFUL *if $\mathcal{A}$ has access to the $\mathcal{O}^{\text{Timer}}$ oracle.*

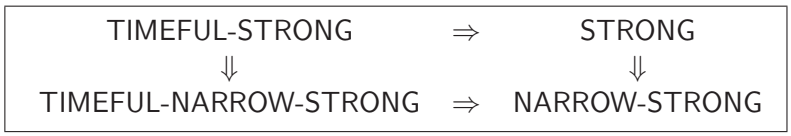
Now, an adversary playing the Vaudenay privacy experiment belongs to the class P such that:

$$\begin{aligned} P \quad \in \quad & \{\emptyset, \mathsf{TIMEFUL}\} \times \{\emptyset, \mathsf{NARROW}\} \\ & \times \{\mathsf{WEAK}, \mathsf{FORWARD}, \mathsf{DESTRUCTIVE}, \mathsf{STRONG}\}. \end{aligned}$$

In this modification of the Vaudenay model, the TIMEFUL adversary class formalizes the *time attack* introduced in this chapter. Concretely, if $\mathcal{A}$ has access to the $\mathcal{O}^{\text{Timer}}$ oracle, she knows the time spent by the reader to identify and/or authenticate a tag. If she cannot deduce anything about the tag identity with this information, then the protocol is said to be TIMEFUL-private.

With the introduction of the TIMEFUL adversary class, new connections at each level (STRONG, DESTRUCTIVE, FORWARD and WEAK)

of the Vaudenay privacy properties diagram (given in Section 3.4.3) can be established. For the sake of clarity, the new links are only given for the STRONG level:



5.2 Existing Protocols

This section analyzes the authentication time of four symmetric-key-based protocols presented in Chapter 4: SK-Prot, OSK-Prot, O-FRAP, and O-FRAPv2. They all differ according to their potential key-update mechanisms, and the last protocol is further defined with a constant-time SearchID procedure. In the classical Vaudenay model, all the following protocols ensure a certain level P of privacy. In the modified model, we will prove that a TIMEFUL adversary prevents these protocols from reaching the TIMEFUL-P-privacy.

For the proofs and analyses, we assume that pseudo-random functions have the same execution time as hash functions for any input.

5.2.1 A Trivial Example: Analysis of SK-Prot

The first study is performed on the SK-Prot authentication protocol presented in Section 4.1.1. As a reminder, the reader sends a nonce $n_{\mathcal{R}}$ to the tag $\mathcal{T}$, which answers the pair $(n_{\mathcal{T}}, F_{\mathsf{k}_{\mathcal{T}}}(n_{\mathcal{R}}, n_{\mathcal{T}}))$, where $n_{\mathcal{T}}$ is a nonce, $\mathsf{k}_{\mathcal{T}}$ is $\mathcal{T}$'s secret key, and F is a pseudo-random function. To authenticate $\mathcal{T}$, the reader performs a linear exhaustive SearchID procedure as follows: for each possible pair $(\mathsf{ID}_{\mathcal{T}}, \mathsf{k}_{\mathcal{T}})$ stored in its database DB, $\mathcal{R}$ computes the output of F using the received nonce. When $\mathcal{R}$ finds a match, it outputs the identifier associated to $\mathcal{T}$.

As proved in Section 4.1.1, this protocol is WEAK-private in the classical Vaudenay model. However, a TIMEFUL adversary can recognize a tag without corrupting it.

Theorem 5.2. *SK-Prot does not ensure TIMEFUL-WEAK-privacy.*

Proof. We exhibit an adversary who has a success probability different than the one of whichever blinded adversary. This adversary $\mathcal{A}$ is even

weaker than the TIMEFUL-WEAK one since she does not use the $\mathcal{O}^{\text{Result}}$ oracle to perform the following attack.

1. $\mathcal{A}$ creates two legitimate tags using twice $\mathcal{O}^{\text{CreateTag}}$ and affects them querying $\mathcal{O}^{\text{DrawTag}}$. She receives two pseudonyms $\mathcal{T}_1$ and $\mathcal{T}_2$.
2. $\mathcal{A}$ queries $\mathcal{O}^{\text{Execute}}(\mathcal{T}_1)$ and $\mathcal{O}^{\text{Execute}}(\mathcal{T}_2)$. She receives the responses $(\pi_1, \mathsf{transcript}_1)$ and $(\pi_2, \mathsf{transcript}_2)$. Then, she asks the time of each protocol execution: she obtains t_1 and t_2 from the queries $\mathcal{O}^{\text{Timer}}(\pi_1)$ and $\mathcal{O}^{\text{Timer}}(\pi_2)$.
3. $\mathcal{A}$ frees both tags with the oracle $\mathcal{O}^{\text{Free}}$, and only randomly reaffects one of them with $\mathcal{O}^{\text{DrawTag}}$. She thus obtains a new pseudonym $\mathcal{T}_3$.
4. $\mathcal{A}$ queries $\mathcal{O}^{\text{Execute}}(\mathcal{T}_3)$ and asks for the time t_3 of this protocol execution with $\mathcal{O}^{\text{Timer}}(\pi_3)$.
5. If $\mathsf{t}_3 = \mathsf{t}_1$, $\mathcal{A}$ claims that $\mathcal{T}_1 = \mathcal{T}_3$, otherwise she claims that $\mathcal{T}_2 = \mathcal{T}_3$.

It is obvious that the success probability of this adversary is 1. For the blinded adversary $\mathcal{A}^{\mathcal{B}}$, the blinder $\mathcal{B}$ does not know which tag has been drawn during the second query to the $\mathcal{O}^{\text{DrawTag}}$ oracle, but he can perfectly simulate the answers of this tag (as F is assumed to be pseudo-random). As $\mathcal{B}$ also simulates $\mathcal{O}^{\text{Timer}}$ to $\mathcal{A}^{\mathcal{B}}$, he has to choose between the two times t_1 and t_2: his only solution is to perform a random choice to keep the simulation perfect. But, $\mathcal{A}^{\mathcal{B}}$ will only have a success probability of $\frac{1}{2}$ as her success is based on the correctness of the blinder's choice. Consequently, the protocol is not TIMEFUL-WEAK-private. □

5.2.2 Analysis of OSK-Prot

The second analyzed protocol is OSK-Prot presented in Section 4.1.3. Since each tag updates its secret key at each protocol execution, an adversary is unable to recompute a previous tag answer after she had corrupted it (as the update is one-way). During the SearchID procedure, $\mathcal{R}$ computes the hash result of $\mathsf{k}_{\mathcal{T}}$ with the nonce $n_{\mathcal{R}}$ sent by $\mathcal{R}$ for each pair $(\mathsf{ID}_{\mathcal{T}}, \mathsf{k}_{\mathcal{T}})$ stored in its database DB. At any moment, if $\mathcal{R}$ finds a match with the received value, it stops the procedure and outputs the corresponding identifier and updates the key, using the g hash function.

After testing every pair $(\mathsf{ID}_{\mathcal{T}}, \mathsf{k}_{\mathcal{T}})$, if the reader did not find a match, it tries again with the updated key (computed on the fly), and so on.

This protocol is NARROW-DESTRUCTIVE-private in the classical Vaudenay model, as proved in Section 4.1.3. However, the time attack presented in the previous subsection is possible. Additionally, for any pair of tags, the time difference of their authentications can be increased. Indeed, as the adversary can desynchronize a tag, she increases the authentication time of a tag at every desynchronization. Consequently, she can easily distinguish one tag among two.

Theorem 5.3. *OSK-Prot does not ensure TIMEFUL-NARROW-DESTRUCTIVE-privacy.*

Proof. The demonstration is similar to the one of Theorem 5.2. □

Remark. In the original article of OSK [134], the SearchID procedure was not described as presented here. In fact, the reader first performs all computations before comparing all these values with the received one. Thus, the time attack does not work with the original procedure when tags are synchronized. Nevertheless, the authentication time of a desynchronized tag will still be longer than the one of a synchronized tag. We presented this SearchID procedure instead of the original one as it is the one generally presented in many contributions.

5.2.3 Analysis of O-FRAP

The last study is performed on the O-FRAP authentication protocol presented in Section 4.1.4. The particularity of O-FRAP is that the reader database DB stores two keys per tag to be able to authenticate each tag: the current one and the old one[1]. As a consequence of this feature, O-FRAP is not vulnerable to desynchronization attacks.

However, this protocol only ensures WEAK-privacy in the classical Vaudenay model, as explained in Section 4.1.4. Even if O-FRAP is an undesynchronizable protocol, an adversary is still capable to perform the previous time attack since the SearchID procedure is linear.

[1] This technique was also present in the Dimitriou protocol [49], even if this protocol does not ensure WEAK-privacy in the classical Vaudenay model as proved in [33].

Theorem 5.4. *The O-FRAP protocol does not ensure* TIMEFUL-WEAK-*privacy.*

Proof. The demonstration is similar to the one of Theorem 5.2. □

5.2.4 Analysis of O-FRAPv2

In the previous studies, the time differences appearing during protocol executions only result from the naive linearity of SearchID procedures. Yet, other protocols designed with improved SearchID complexities (e.g., in constant time) do not necessarily prevent time attacks as these last ones examine the whole protocol execution.

This section illustrates this statement with the analysis of the O-FRAPv2 authentication protocol depicted in Figure 5.1. This protocol has been proposed by Burmester, de Medeiros and Motta in [29] as an improved version of O-FRAP: its SearchID procedure is a constant-lookup instead of being linear.

Constant-lookup Feature of O-FRAPv2. To obtain this result, the authors introduced a new value for each tag that can be viewed as a pseudonym. During a protocol execution, the tag sends this value joined with an authentication value. As O-FRAPv2 has been proposed to ensure privacy (in the sense of unlinkability), this pseudonym must change between each (successful or not) protocol execution. It happens in two different ways, depending on whether the tag suspects an attack or not. To prevent entrapment attacks, if the tag $\mathcal{T}$ does not receive the confirmation that the reader authenticated it, $\mathcal{T}$ does not update its pseudonym but its counter $\mathsf{v}_\mathcal{T}$, and computes on-the-fly a pseudonym based on the counter (i.e., $F_{\mathsf{k}_\mathcal{T}}(\varsigma_\mathcal{T}, \mathrm{IV}, \mathsf{v}_\mathcal{T})$, where $\mathsf{k}_\mathcal{T}$ and $\varsigma_\mathcal{T}$ are $\mathcal{T}$'s secret key and seed, and IV is an initialization vector).

For each tag $\mathcal{T}$, the database DB contains its identifier $\mathsf{ID}_\mathcal{T}$, its secret key $\mathsf{k}_\mathcal{T}$, a counter $\mathsf{v}_\mathcal{T}$, a seed $\varsigma_\mathcal{T}$ used for entrapment pseudonyms, a one-time pseudonym $\psi^i_\mathcal{T}$ and all its possible pseudonyms $\psi^{i,j}_\mathcal{T}$ for each $i \in \{old_\mathcal{T}, cur_\mathcal{T}\}$. Therefore, DB must store $(2\ell + 3)$ values for each tag, where ℓ is the highest value of the counter $\mathsf{v}_\mathcal{T}$. By storing this huge amount of data, the reader is able to perform a really fast SearchID as all the possible tag answers are already precomputed in DB. On the one hand, the price to pay to obtain such a result is a large database where its

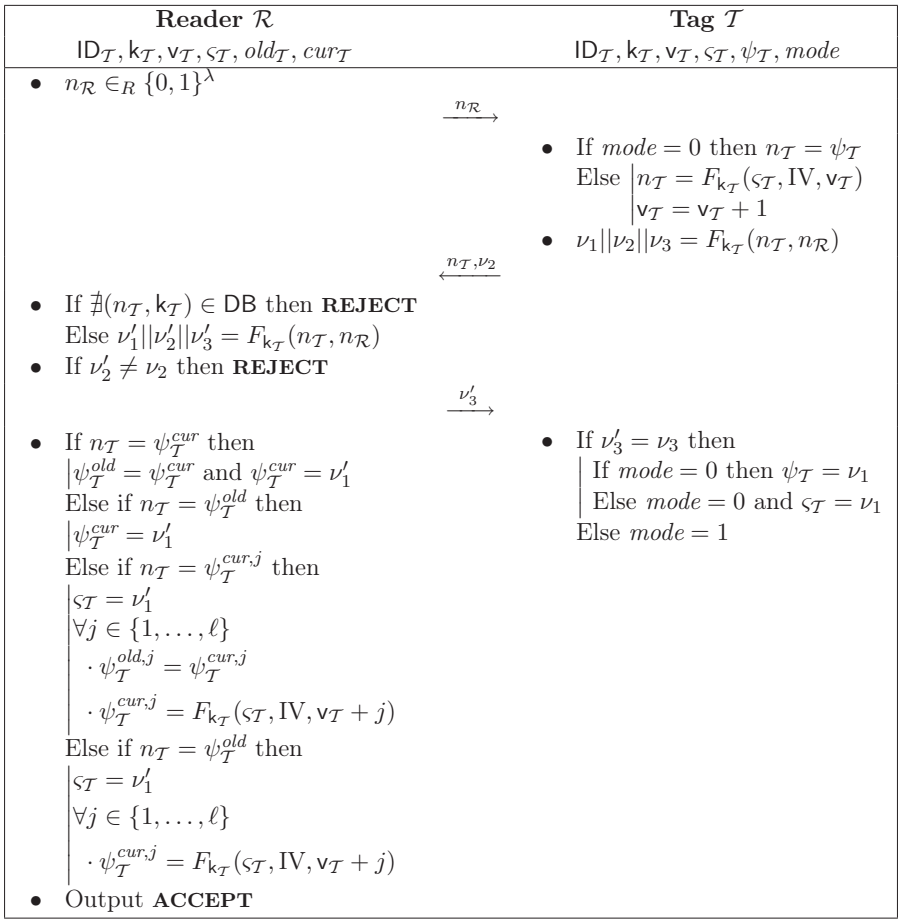


Figure 5.1: O-FRAPv2 authentication protocol.

size is parametrized by ℓ and the number n of tags in the system. On the other hand, when the reader receives an answer from a tag, SearchID only verifies that the received value belongs to DB. If so, the reader has authenticated the tag, and the end of the protocol consists of an update procedure of the values in DB and in the tag.

Time Attack on O-FRAPv2. As presented in [29], the SearchID procedure is in constant time, but the reader will spend more or less time to output its result under specific conditions. For example, if the tag

uses an entrapment value for $n_{\mathcal{T}}$ (i.e., $F_{\mathsf{k}_{\mathcal{T}}}(\varsigma_{\mathcal{T}}, \mathrm{IV}, \mathsf{v}_{\mathcal{T}})$), the reader has to computes ℓ values to replace each of the $\psi_{\mathcal{T}}^{cur,j}$. On the contrary, if $n_{\mathcal{T}} = \psi_{\mathcal{T}}$, the reader has no computation to realize. As a consequence, the time to output the result is different between these two cases: a TIMEFUL adversary $\mathcal{A}$ is thus able to distinguish if the reader has performed this computation or not. By stopping the protocol before the tag receives ν_3', $\mathcal{A}$ forces the tag to use an entrapment value and will consequently be able to distinguish this tag from a "synchronized" one (i.e., where $mode = 0$) during the next protocol execution. A trivial solution to this attack is to output the result before processing the update of the values.

Other Attacks on O-FRAPv2. This protocol further suffers from security flaws that allow (i) a FORWARD adversary to trace all the previous protocol executions of a tag, and (ii) a WEAK adversary to link two executions of a tag.

For the first attack, it is sufficient to notice that a FORWARD adversary who learns the secret key $\mathsf{k}_{\mathcal{T}}$ of a tag $\mathcal{T}$ is able to recompute all the previous values ν_2 of $\mathcal{T}$, as she can recompute $F_{\mathsf{k}_{\mathcal{T}}}(n_{\mathcal{T}}, n_{\mathcal{R}})$ (because $n_{\mathcal{T}}$ and $n_{\mathcal{R}}$ are sent in clear).

The second attack is based on the value $n_{\mathcal{T}}$ sent by the tag $\mathcal{T}$. During the i^{th} protocol execution, $\mathcal{T}$ sends $n_{\mathcal{T}} = \psi_{\mathcal{T}}$. If a WEAK adversary blocks the last message, the tag only updates its value *mode* to 1. During the $(i+1)^{\text{th}}$ protocol execution, $\mathcal{T}$ uses an entrapment value as $mode = 1$. When the protocol ends, $\mathcal{T}$ changes *mode* to 0, and updates $\varsigma_{\mathcal{T}}$ to ν_1. At that moment, $\mathcal{T}$ did not update $\psi_{\mathcal{T}}$ either in the i^{th} or in the $(i+1)^{\text{th}}$ protocol execution. Consequently, during the $(i+2)^{\text{th}}$ protocol execution, $\mathcal{T}$ sends $n_{\mathcal{T}} = \psi_{\mathcal{T}}$ where the value $\psi_{\mathcal{T}}$ is the same as in the i^{th} protocol execution. Thus, the adversary can trivially recognize $\mathcal{T}$ from any other tag.

5.2.5 Conclusion of Time Attacks

This section demonstrated that a TIMEFUL adversary is able to break the privacy of several protocols, while they were assumed to ensure a certain level of privacy in the classical Vaudenay model[2]. The analyzed protocols do not represent an exhaustive list of those threatened by these

[2]And in other privacy models as shown in Chapter 4.

new time attacks. For instance, the C^2 undesynchronizable protocol introduced by Canard and Coisel in [33] is also subject to such attacks as the SearchID procedure also implies a linear exhaustive search.

These attacks also work on protocols where the underlying key infrastructure is different. For example, protocols like MW-Prot (presented in Section 4.1.2) suffer from a time-flaw when the SearchID procedure is defined as a linear exhaustive search at each tree level.

Time attacks particularly affect protocols based on symmetric-key cryptography. Nevertheless, some authentication protocols using public-key techniques can be attacked by a TIMEFUL adversary. For instance, this is the case of the protocol introduced by Bringer, Chabanne and Icart in [28], where the SearchID procedure is also linear.

5.3 Solutions and Improvements

The attacks presented in the previous section always work in theory, but they require in practice a tight time measurement. For instance, let us consider SK-Prot with a linear SearchID procedure. If the data of two tags are very close in the reader database (e.g., one following the other), the time of the SearchID procedure will be almost the same for both: the time difference between the computation of i or $(i+1)$ functions F can be very small. Therefore, if the time measurement of the adversary is not precise enough, she will not be able to differentiate one of these two tags from the other one. Since such an adversarial issue mainly depends on the implementation of the function F, we consider that these practical concerns are out of the scope of this thesis.

This section proposes theoretical solutions to this time problem that can be applied to the analyzed protocols. The most obvious one is to compute the worst case time for any protocol execution, and the reader waits[3] until it reaches this time before outputting the result. This solution has been mentioned by Burmester, Le, and de Medeiros in [30] and clearly repairs all the previous protocols against the time attack. However, the protocol efficiency can be highly decreased depending on the number of tags involved in the system. Consequently, the goal of the

[3]During this time, it can also compute unused cryptographic functions to avoid power consumption attacks.

proposed solutions is to optimize the average protocol execution time while being resistant to a TIMEFUL adversary.

5.3.1 Constant-time Identification

Some protocols with a constant-time identification have been proposed (e.g., [1, 29]), not necessarily to solve the problem of time attack, but rather to reduce the complexity of the SEARCHID procedure. This section presents in detail the protocol proposed by Alomair, Clark, Cuellar and Poovendran in [1].

Description of the Protocol. The important step of the protocol, depicted in Figure 5.2, is the sent value $E_0 = h(\psi, \mathsf{v})$. Indeed, ψ represents a pseudonym associated to a tag $\mathcal{T}$, and v is a counter value which is incremented after each (successful or not) tag authentication. In the reader database, all the possible hash values for all the possible pseudonyms and all the counter values are precomputed and stored. Based on a special infrastructure (detailed in the following paragraph), the reader is able to quite instantaneously retrieve all the associated values to the corresponding pseudonym, i.e., the secret key $\mathsf{k}_{\mathcal{T}}$ of the tag $\mathcal{T}$ and its identifier $\mathsf{ID}_{\mathcal{T}}$. The reader is then able to compute the last message of the protocol which is composed of three parts (i.e., E_2, E_3, and E_4). The first one allows $\mathcal{T}$ to authenticate the reader. The second one securely transmits to $\mathcal{T}$ a new pseudonym ψ' which has been selected among the available ones in the database. The last one permits $\mathcal{T}$ to check the integrity of this new pseudonym ψ'.

Database Infrastructure. As explained in [1], the database can be decomposed in three logical parts. The first one, denoted M-I, can be viewed as a hash table which allows the definition of a direct addressing to the hash values $h(\psi, \mathsf{v})$. All these values are stored in the second part of the database, denoted M-II. Finally, each of these hash values points to one cell of the last part of the database, denoted M-III, which contains all the information related to the tag currently attached to the pseudonym ψ.

Remark. There is a mistake in the description of this database in [1]. Indeed in the M-II table, the authors said that each cell only contains

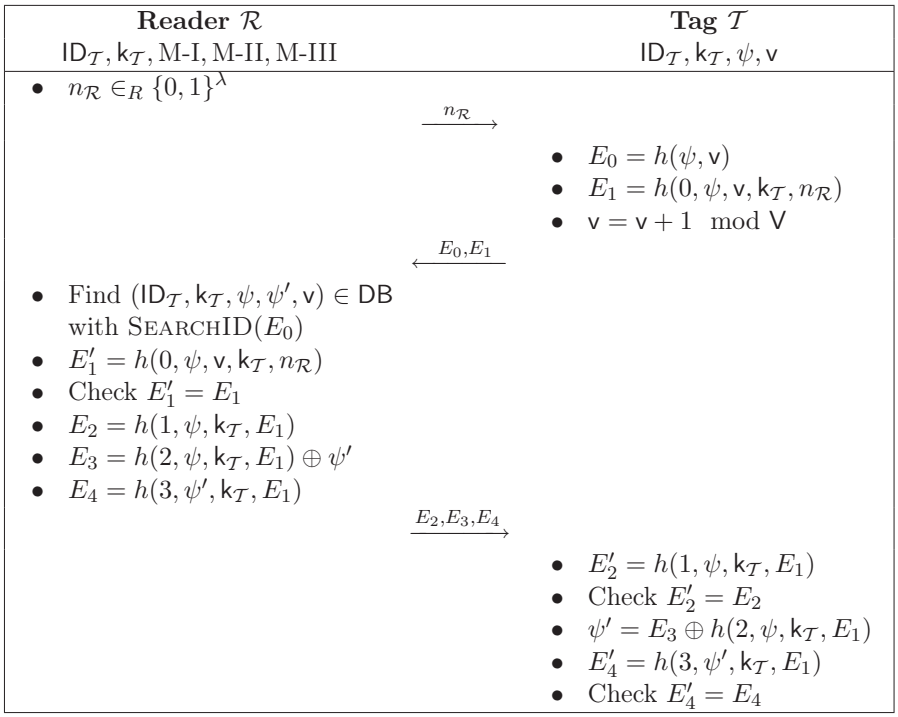


Figure 5.2: Constant-time identification protocol.

the hash value and a pointer to a tag data. However, each cell must contain the counter value v and the pseudonym which are used in the hash value. These values are essential to check the message E_1 sent by the tag by recomputing it.

Database Size. In their case study, the authors of [1] only consider the size of M-I as they claim that M-I is the unique concern for the total size of the database. However, the size of the M-II part is substantial as showed in the following.

Let us use the same parameters as those provided in [1]. Namely, the total number of pseudonyms is $\mathsf{N} = 2.10^9$ and the counter v is majored by $\mathsf{V} = 10^3$. Thus, the part M-II is composed of 2.10^{12} cells. Each one contains the hash value $h(\psi, \mathsf{v})$, the counter v and a pointer to the table M-III. Note that a pointer of 32 bits is enough for addressing the

2.10^9 cells of the part M-III (one for each pseudonym). In [1], the authors say that the output of the hash function must be at least of $\lfloor \log_2 \mathsf{NV} \rfloor \approx 41$ bits. Then, each cell of M-II contains at least 83 bits (the counter is approximately 10 bits long). The resulting database size is of at least 166.10^{12} bits, which is approximatively equal to 19 terabytes. This is obviously not negligible compared to the 12 terabytes of M-I.

Although this is still feasible in practice, it is not so practicable. Furthermore, the authors of [1] neglect another fact yet highlighted in the paper. As a tag can be desynchronized once, each tag should be associated to two pseudonyms. Thus, the total number of pseudonyms should not be twice the total number of tags, but more than this, for example three times this number[4]. Again, this fact increases the size of M-II (from 19 to 28 terabytes for this example), and therefore the whole database size (from 31 to 40 terabytes for the same example).

Modification of the Database. To decrease the size of the database, the part M-II could only store the counter v and the pointer. In average, this modification will not be a problem because, as presented in [1], pointers of M-I are attached to only one cell in most cases. As $\mathcal{R}$ must check the correctness of E_1, it will be able to differentiate two tags that have the same address in M-I. Of course, this optimization increases the number of computations performed by $\mathcal{R}$ in order to identify a tag. For example, if a hash value points to two different tags in M-II, $\mathcal{R}$ may compute two hash values to be sure of an authentication. However, this optimization decreases the size of M-II from 28 to 14 terabytes which is not negligible, and the collision event still occurs with a small probability.

Note that the authentication time is no longer constant with this modification. Nevertheless, an adversary is not able to predict if a tag, and more precisely a pseudonym with a given counter, will collide in the M-II table with another pseudonym and the same counter. Thus, she is not able to trace a tag with this difference of time.

Despite this huge amount of data, this protocol is however clearly efficient in terms of time. It further gives a solution to the time attacks presented in this chapter by providing a constant-time SearchID

[4]Remember that the reader must always be in possession of a set of available pseudonyms in order to randomly reaffect the current authenticated tag.

procedure while reaching the TIMEFUL-WEAK-privacy property.

5.3.2 Random Search

The second solution applies on protocols where the reader cannot predict the tag outputs (i.e., the tag inserts a nonce in its answer). Its core methodology is to randomize this search. The objective is to avoid the adversary to predict the time spent for a given tag authentication. To do so, the easiest way is to modify each time the "starting-point" of the linear search in the database. If the starting-point is randomly chosen in $\{1, \ldots, n\}$ (where n is the total number of tags), then the authentication time of a tag can never be guessed.

In Practice with O-FRAP. In this protocol, the only solution for the reader to authenticate a tag is to compute for each key stored in its database the theoretical output of the corresponding tag, and to compare it with the received value. Section 5.2.3 showed that a TIMEFUL adversary can trace a tag with a non-negligible probability when the SearchID procedure is performed linearly.

Furthermore, a tag can be desynchronized once when the system is based on O-FRAP. With the random search solution, the reader should first compute all the theoretical outputs using the current keys. Then, it should only compute these outputs with the old keys if it did not find a match with one of the current ones. Thus, a synchronized tag is authenticated in $\mathtt{t}_F.n/2$ time in average, whereas a desynchronized tag is authenticated in $3\mathtt{t}_F.n/2$ time in average, where $\mathtt{t}_F$ denotes the execution time of the F function. So, a TIMEFUL adversary is always able to trace a tag when she desynchronizes it. Consequently, to ensure the TIMEFUL-WEAK-privacy, the randomized SearchID procedure should indifferently test the current or the old keys of the tags. This procedure should hence consider a set of $2n$ keys and randomly tests one key after the other without considering if it is a current key or an old one.

Using this full randomized SearchID procedure, it is obvious that protocols like O-FRAP reach the TIMEFUL-WEAK-privacy property. Unfortunately, the price to pay is the decrease of the system efficiency. Indeed, the average time to authenticate a tag under a normal behavior (when the adversary does not desynchronize a tag) is $\mathtt{t}_F.n$ instead of

$\mathsf{t}_F.n/2$. Yet, such a random search still improves the "wait" solution as the latter requires a time of $2\mathsf{t}_F.n$ for O-FRAP.

In Practice with SK-Prot. This protocol does not use a key-update mechanism. Therefore, the random search solution does not modify the average authentication time of a tag (i.e., $\mathsf{t}_F.n/2$) while it allows the protocol to ensure the TIMEFUL-WEAK-privacy property.

5.3.3 Enhancing OSK-Prot

Another situation is when it is possible to precompute the whole set of possible tag answers (or a part of it). Then it is possible to use some time-memory trade-off to enhance the complexity of the SearchID procedure and, in some cases, to obtain a constant look-up in the database. To apply such a methodology, we consider a variant of OSK-Prot where the tag response E is the message $f(\mathsf{k}_\mathcal{T}, n_\mathcal{R})$ concatenated with $f(\mathsf{k}_\mathcal{T})$ (i.e., the tag response sent in the classical OSK protocol for identification).

In this variant, the second part $f(\mathsf{k}_\mathcal{T})$ of the tag response does not include any nonce and thus the database can store all the potential answers. As for OSK-Prot, this variant also is extremely desynchronizable. Consequently, instead of storing one answer per tag, the database should contain the δ successive answers of each tag. This highly increases the size of the database ($O(n.\delta)$ instead of $O(n)$). This section shortly presents three possible database infrastructures to enhance the efficiency of the SearchID procedure that allow this variant to ensure the TIMEFUL-NARROW-DESTRUCTIVE-privacy property.

Rainbow Tables. The first optimization is called OSK-AO. It has been proposed by Avoine and Oechslin in [19] (then extended by Avoine, Dysli and Oechslin in [15]) and is based on the well-known rainbow tables, introduced by Oecshlin in [129].

In a nutshell, all of the $n.\delta$ possible answers for $f(\mathsf{k}_\mathcal{T})$ (i.e., the hash values of the δ successive keys for each of the n tags) are distributed uniformly in a table, whose size defines the time-memory trade-off. Each row of the database contains a succession of hash values. One value is obtained from the previous one by applying an arbitrary reduction

function composed with a hash function h. This reduction function takes in input a hash value and outputs an identifier in $\{1, \ldots, n\}$ and the "update value" in $\{1, \ldots, \delta\}$. Using these two values, the next hash value can be computed using the hash function h. The database has to store the first and the final column of this table. Upon receiving a hash value, the reader will compose a chain of values (as done in the construction of the table) until a match is found with the last column of the table. When this fact occurs, the reader reconstructs the corresponding row until it finds the previous value, and thus obtains the identifier and the update value.

If the reduction function maps all the possible values in a uniform manner in the database, an adversary is not able to predict the authentication time for a given tag, and thus to trace it. Moreover, contrary to the next solutions, this structure does not store the $n.\delta$ potential tag answers. However, it is not dynamic and cannot be modified. The whole table must thus be recomputed in order to introduce new tags.

Hash Tables. Another solution is to compose the database as a hash table, where the entries are indexed by the hash values. In such a database, the SearchID procedure is quite instantaneous ($O(1)$ in average). However, to avoid collisions, the hash index used should be as long as the output of the hash function. This is quite impracticable when $n.\delta$ is large. Moreover, this solution is not adapted for dynamic systems where the number of tags can increase during the system lifetime. Indeed, as the number of inputs increases, so does the probability of a collision in the hash index. If this fact occurs, then SearchID takes up to linear time (in $O(n)$).

Moreover, the database should keep the current key of each tag in another table. Indeed, if the tag is desynchronized, the use of the hash table allows the reader $\mathcal{R}$ to authenticate the tag and to obtain the used key, but $\mathcal{R}$ will not be able to recompute the previous ones (as the hash function is one-way). Consequently, to delete the previous entries of the database (to keep it as small as possible), $\mathcal{R}$ should use this new table to recompute all the previous theoretical answers to delete them.

B-trees. Finally, another possibility is to use a balanced binary search tree (B-tree). This technique ensures a complexity in $O(\log n)$ for the

SearchID procedure. The advantage of this structure is its dynamism. Contrary to a hash table, new entries can be indefinitely added in this structure without compromising its functioning. Moreover, in the worst case, a B-tree ensures a better complexity than a hash table where the complexity search is in $O(n)$ in the worst case (i.e., when collisions happen on the hash indexes).

These three practical solutions avoid time attacks for the variant of OSK-Prot, and thus ensure the TIMEFUL-NARROW-DESTRUCTIVE-privacy of the protocol. However, except for the OSK-AO solution, the size of the database is $O(n.\delta)$. This requirement may become quickly infeasible, especially if desynchronization attacks should be highly avoided (meaning that δ must be large enough).

Chapter 6

When Compromised Readers Meet RFID

In most widespread RFID systems, readers are not always connected online to the back-end, and may be mobile devices that have an intermittent access to it. In such real-life scenarios, readers must carry some sensitive information to authenticate the tags of the system during their offline periods (e.g., tags secrets). Additionally, the ubiquity of readers, usually located in unprotected areas, increases the risk of theft. Consequently, if an adversary $\mathcal{A}$ corrupts a stolen reader and obtains its secrets, then the security of the whole system is threatened by such a *compromised* reader. Furthermore, the system privacy is completely lost as $\mathcal{A}$ is clearly able to trace any tag thanks to the retrieved secrets. This new issue in RFID has been independently introduced by Garcia and van Rossum in [68] and by Avoine, Lauradoux and myself in [16].

In this chapter, we first formally model the "compromised readers" attack. Then, we demonstrate that a multi-reader-based RFID authentication protocol does not ensure privacy in such a context. Finally, we propose two practical solutions based on privacy-restoring mechanisms to face compromised readers: one for semi-offline systems, and another one for offline systems. We show that the offline solution is deployable in practice by analyzing the efficiency of its privacy-restoring mechanism during a 3-day automobile race that took place in 2010. Up to our knowledge, *restoring* privacy in RFID systems is a new concept introduced in this chapter.

6.1 Modelization of Compromised Readers

In this chapter, an RFID system $\mathcal{S}$ is considered to be composed of a trusted back-end, n tags and m readers, where the readers may not always be connected online to the back-end. This section presents the formal modelization of an adversary who can corrupt several readers.

6.1.1 Context

As already underlined so far in this thesis, existing privacy models have been designed following the assumptions that (i) the communication channel between tags and readers is not secure, (ii) tags are not necessarily tamper-resistant, and (iii) the readers and the back-end are always securely connected online and cannot be attacked. Yet, this last assumption is usually too strong to fit the reality. Indeed, many widespread RFID systems are designed such that readers can only be, at most, sporadically connected to the back-end. For instance, in some public transportation, the ticket validators in buses may only have access to the back-end when the vehicles are parked in their lot, usually at night.

To operate during offline intervals, readers must store the secret keys of all the distributed tags, e.g., in a Security Authentication Module (SAM). An adversary will thus be very interested in stealing such a reader and using expensive intrusive attacks to tamper with the device. Whether or not an adversary gets the keys from the stolen reader, this reader must anyway be considered as *compromised*.

6.1.2 Security Goals

As already stressed in the introductory chapter of this thesis, an RFID system $\mathcal{S}$ is expected to comply with the three following fundamental security properties. The two first notions of *availability* and *soundness* are the basis of the well functioning of an RFID system. The *privacy* one follows the intuitive indistinguishability notion exhibited by the JW model (see Section 3.3 for more details).

Definition 6.1 (Availability). *An authentication protocol is said to be* available *if the probability that a legitimate reader successfully authenticates a legitimate tag that could have been subjected to an attack is overwhelming.*

Definition 6.2 (Soundness). *An authentication protocol is said to be* sound *if the probability that an adversary is successfully authenticated as a legitimate tag by a legitimate reader is negligible.*

Definition 6.3 (Privacy). *An authentication protocol is said to be* private *if the probability that an adversary is able to differentiate one legitimate tag from another by interacting with the system is negligible.*

6.1.3 Adversary Means

In this chapter, the analysis of an RFID system is performed according to the three adversary classes given below. They are related to the adversary ability to corrupt readers and tags during an experiment.

The two first adversary classes against the RFID system $\mathcal{S}$ are the underlying ones given in the JW model (denoted STANDARD and FORWARD for the sake of clarity in this chapter). The choice of these classes is the result of the discussion related to the different levels of tag corruption given in Section 4.2.2. The STANDARD adversary is the classical one that can play/interact with all the entities of the system, and that can corrupt tags, except the challenge ones (if so) of the experiment. The FORWARD adversary is a STANDARD one without this restriction. Note that relay attacks are not considered here in the adversary classes since the proposed solution is not a distance bounding protocol.

The third class is the one proposed by Avoine, Lauradoux, and myself in [16], and is formally defined below. Note that this class is orthogonal to the two other ones, and can be therefore used as a combination.

Definition 6.4 (CORRUPT Adversary). *An adversary $\mathcal{A}$ against the RFID system $\mathcal{S}$ is said to be* CORRUPT *if she can corrupt readers and obtain their secrets.*

6.1.4 Adversary Goals

An adversary $\mathcal{A}$ is also defined by her objectives. This section formalizes the three security properties provided in Section 6.1.2 by experiments.

Let $\mathcal{S}$ be the RFID system of global security parameter λ. Let $\epsilon(.)$ be a negligible function. The adversary class P allowed for each experiment is such that:

$$P \in \{\emptyset, \mathsf{CORRUPT}\} \times \{\mathsf{STANDARD}, \mathsf{FORWARD}\}.$$

Note that the corruption of an entity does not mean that $\mathcal{A}$ has the control of the entity: she only knows the entity secrets, and the entity is still considered as legitimate.

Availability. $\mathcal{A}$ wants to make a legitimate tag $\mathcal{T}$ no longer authenticable by a legitimate reader $\mathcal{R}$. This attack is associated to Definition 6.1 and detailed in Figure 6.1.

Experiment $Exp^{\mathsf{Avail}}_{\mathcal{S},\mathcal{A}}[\lambda]$

1. The challenger $\mathcal{C}$ initializes the RFID system $\mathcal{S}$.
2. $\mathcal{A}$ interacts with the whole system, limited by her class P.
3. $\mathcal{A}$ outputs a challenge tag $\mathcal{T}$ and a reader $\mathcal{R}$, and makes $\mathcal{R}$ and $\mathcal{T}$ execute the protocol.

$Exp^{\mathsf{Avail}}_{\mathcal{S},\mathcal{A}}$ succeeds if $\mathcal{R}$ can no longer successfully authenticate $\mathcal{T}$.

Figure 6.1: Availability experiment.

The availability of the system $\mathcal{S}$ is ensured if

$$\Pr(Exp^{\mathsf{Avail}}_{\mathcal{S},\mathcal{A}}[\lambda] \text{ succeeds}) \leq \epsilon(\lambda).$$

Soundness. $\mathcal{A}$ wants to be successfully authenticated as a legitimate non-compromised tag by a legitimate non-compromised reader $\mathcal{R}$. This attack is associated to Definition 6.2 and detailed in Figure 6.2.

The soundness of the system $\mathcal{S}$ is ensured if

$$\Pr(Exp^{\mathsf{Sound}}_{\mathcal{S},\mathcal{A}}[\lambda] \text{ succeeds}) \leq \epsilon(\lambda).$$

Privacy. $\mathcal{A}$ wants to differentiate one legitimate tag $\mathcal{T}$ from another legitimate tag $\mathcal{T}'$. This attack is associated to Definition 6.3 and detailed in Figure 6.3.

The privacy of the system $\mathcal{S}$ is ensured if

$$\left|\Pr(Exp^{\mathsf{Priv}}_{\mathcal{S},\mathcal{A}}[\lambda] \text{ succeeds}) - \frac{1}{2}\right| \leq \epsilon(\lambda).$$

Experiment $Exp^{\mathsf{Sound}}_{\mathcal{S},\mathcal{A}}[\lambda]$

1. The challenger $\mathcal{C}$ initializes the RFID system $\mathcal{S}$.
2. $\mathcal{A}$ chooses a reader $\mathcal{R}$ that she will never corrupt.
3. $\mathcal{A}$ interacts with the whole system, limited by her class P.
4. $\mathcal{A}$ interacts with $\mathcal{R}$, and outputs an answer to $\mathcal{R}$.

$Exp^{\mathsf{Sound}}_{\mathcal{S},\mathcal{A}}$ succeeds if $\mathcal{R}$ authenticates $\mathcal{A}$ as being a legitimate non-compromised tag.

Figure 6.2: Soundness experiment.

Experiment $Exp^{\mathsf{Priv}}_{\mathcal{S},\mathcal{A}}[\lambda]$

1. The challenger $\mathcal{C}$ initializes the RFID system $\mathcal{S}$.
2. $\mathcal{A}$ interacts with the whole system, limited by her class P.
3. $\mathcal{A}$ chooses two non-compromised tags $\mathcal{T}$ and $\mathcal{T}'$.
4. $\mathcal{C}$ chooses a bit b at random. Then $\mathcal{C}$ assigns the challenge tags $\mathcal{T}_b = \mathcal{T}$ and $\mathcal{T}_{b\oplus 1} = \mathcal{T}'$.
5. $\mathcal{A}$ interacts with the whole system limited by her class P. If $\mathcal{A}$ is $\mathsf{CORRUPT}$, she cannot corrupt readers anymore.
6. $\mathcal{A}$ outputs a guess bit b'.

$Exp^{\mathsf{Priv}}_{\mathcal{S},\mathcal{A}}$ succeeds if $b = b'$.

Figure 6.3: Privacy experiment.

Note that this privacy experiment is a variant of the user-friendly JW one. We do not use the original JW model because (i) neither the interdiction to play with $\mathcal{T}^*_{b\oplus 1}$ nor the limitation (ρ, σ, τ) of adversary interactions is required, and (ii) JW does not define either an RFID system with several readers, or an adversary able to corrupt readers.

6.2 Privacy of a Multi-reader-based Protocol

This section presents TanSL, an authentication protocol specifically designed for RFID systems composed of several readers. TanSL is operational without the need of a continuous connection between readers and the back-end. It implements a remarkable secret key recomputation mechanism such that each reader only carries its own secrets. This section then analyses the privacy level of TanSL when it faces the problem of compromised readers.

6.2.1 TanSL Protocol

TanSL is the first protocol of the three multi-reader-based ones proposed in [159] by Tan, Sheng, and Li. It is a challenge/response protocol involving a single hash function. The secret key shared between each reader and each tag has the interesting feature to be computed on-the-fly by the tag at each protocol execution.

Description of the Protocol. At the initialization of the system, each tag $\mathcal{T}$ is assigned to a unique identifier $\mathsf{ID}_\mathcal{T}$ and a long-term secret key $\mathsf{k}_\mathcal{T}$. Each reader $\mathcal{R}$ is also assigned to an identifier $\mathsf{ID}_\mathcal{R}$ and stores, for every tag $\mathcal{T}$, its identifier $\mathsf{ID}_\mathcal{T}$ and a shared secret key $\mathsf{s}_{\mathcal{TR}}$ which is a hash value of $\mathcal{R}$'s identifier concatenated with $\mathcal{T}$'s long-term secret key: $\mathsf{s}_{\mathcal{TR}} = h(\mathsf{ID}_\mathcal{R}, \mathsf{k}_\mathcal{T})$, where h is a cryptographic hash function. The shared secret key $\mathsf{s}_{\mathcal{TR}}$ is used to perform the authentication protocol depicted in Figure 6.4.

The question $ques_\mathcal{R} = (ques^1_\mathcal{R}, \ldots, ques^i_\mathcal{R})$ (resp. $ques_\mathcal{T}$) represents i randomly chosen bit positions from E_2 (where $i \leq \frac{|E_2|}{2}$). The answer $ans_\mathcal{R}$ (resp. $ans_\mathcal{T}$) represents the actual bits in positions $ques^1_\mathcal{R}, \ldots, ques^i_\mathcal{R}$ (resp. $ques^1_\mathcal{T}, \ldots, ques^i_\mathcal{T}$) of E_2.

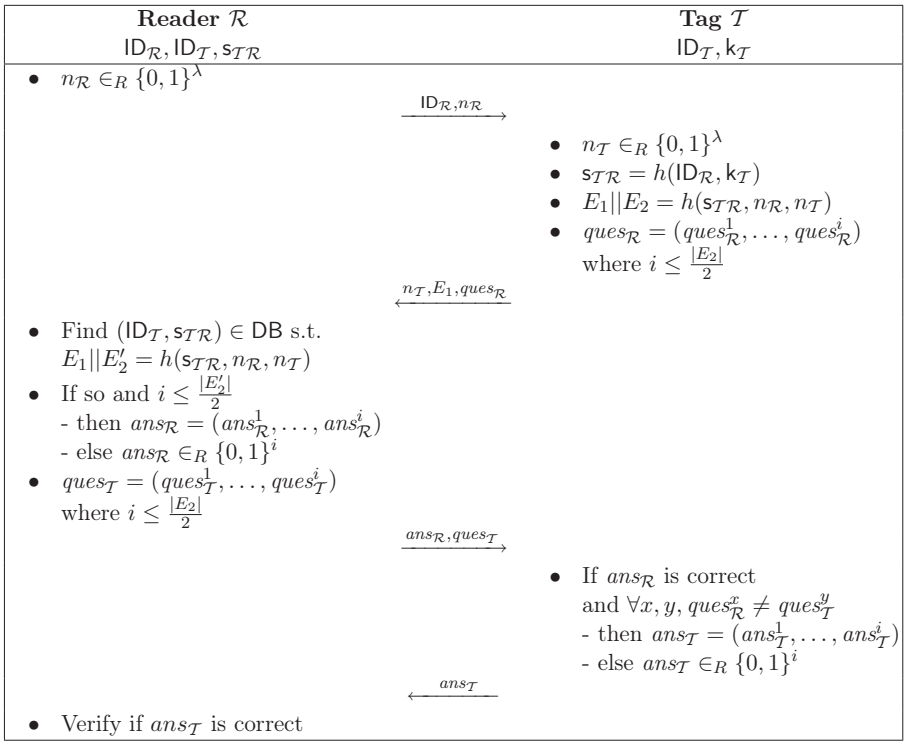


Figure 6.4: TanSL protocol.

Reader Complexity. As for SK-Prot (presented in Section 4.1.1), the reader carries out a linear exhaustive search on its database to find the correct pair $(\mathsf{ID}_\mathcal{T}, \mathsf{k}_\mathcal{T})$ among the pairs of all the tags of the system in order to authenticate the tag $\mathcal{T}$. The reader complexity is thus in $O(n)$.

6.2.2 Privacy Analysis

The privacy of TanSL is analyzed against CORRUPT-STANDARD adversaries as defined in Section 6.1.3 following the privacy experiment given in Section 6.1.4. The study shows that this promising protocol does not ensure privacy when facing an adversary that can corrupt readers.

Theorem 6.5. *TanSL does not ensure CORRUPT-STANDARD-privacy.*

Proof. We exhibit a CORRUPT-STANDARD adversary $\mathcal{A}$ that is always able to win the privacy experiment with probability of 1.

Let us assume that $\mathcal{A}$ corrupts a reader $\mathcal{R}$ at step 2 of the privacy experiment. $\mathcal{A}$ knows $\mathcal{R}$'s identifier $\mathsf{ID}_{\mathcal{R}}$ and all the pairs $(\mathsf{ID}_{\mathcal{T}}, \mathsf{s}_{\mathcal{TR}})$ that are required to allow $\mathcal{R}$ to authenticate every tag $\mathcal{T}$. She is thus able to differentiate $\mathcal{T}$ from $\mathcal{T}'$ at step 3 of the privacy experiment by verifying the answers E_1.

At step 5, $\mathcal{A}$ can impersonate $\mathcal{R}$ in front of any tag, especially $\mathcal{T}_0$ and $\mathcal{T}_1$. When she interacts with $\mathcal{T}_0$ (resp. $\mathcal{T}_1$) at that step, $\mathcal{A}$ verifies if the answer E_1 sent by $\mathcal{T}_0$ (resp. $\mathcal{T}_1$) is equal to the first half of $h(\mathsf{s}_{\mathcal{TR}}, n_{\mathcal{R}}, n_{\mathcal{T}_0})$ (resp. $h(\mathsf{s}_{\mathcal{TR}}, n_{\mathcal{R}}, n_{\mathcal{T}_1})$). If so, it means that $\mathcal{T}_0$ (resp. $\mathcal{T}_1$) corresponds to $\mathcal{T}$. $\mathcal{A}$ will always succeed in her attack because the tag long-term secret keys are never modified, which implies that the shared secret key $\mathsf{s}_{\mathcal{TR}}$ used by $\mathcal{T}$ to get authenticated by the reader $\mathcal{R}$ is fixed as well.

Thus, the probability that $\mathcal{A}$ wins the privacy experiment is 1. □

6.3 Solution for Semi-offline RFID Systems

The previous section demonstrated that the privacy of an RFID system based on the promising TanSL protocol is completely lost in case of compromised readers. Hence, it is critical for a system to be able to retrieve its privacy level in such a scenario. One trivial remedy is to physically renew all the tags of the system. We discard this solution because (i) it is too costly for the system provider, (ii) if we consider punctual events, the lack of time makes this task not manageable, and (iii) this would alarm the customers and degrade the image of the company.

The goal of the following work is to propose an efficient solution to handle the problem of compromised readers in semi-offline systems. This is performed through a procedure that is able to restore the privacy of the system. The solution aims to be deployable in practice with the exclusive use of lightweight cryptography (e.g., symmetric-key cryptosystem).

6.3.1 Semi-offline Architecture

A semi-offline RFID system $\mathcal{S}$ has the following properties. The readers can be sealed for their own protection, and are sporadically connected to the back-end through a secure channel. Tags are low-cost and thus

only capable of lightweight computations. In addition, they are tamper-resistant to a certain extent [98]. Consequently, each tag has a unique secret so that the cost of a tampering attack is higher than its benefit: this assumption prevents a large-scale attack where an adversary recovers the secret of all the tags by breaking only one of them.

6.3.2 Privacy-restoring Mechanism

To illustrate the proposed solution, let consider the example of public transportation. In such a semi-offline system, the readers inside the buses are disconnected from the system when they are in circulation during the day. At night, these readers get connected to the back-end when all the buses get back to the warehouse. Let now assume that a reader has been compromised (e.g., stolen or damaged[1]) during the day. At that point, the privacy of the system is completely lost.

To restore privacy in such semi-offline systems, the main idea of the solution is to add a protocol whose purpose is, once launched, to update all the readers of the system at the same time. This can be performed when all the readers are gathered at the warehouse. This update is further assumed to be securely performed, and no adversary can access the update protocol execution (e.g., in practice, the warehouse is a safe place and protected against any potential adversary).

Once the update protocol finished, all the updated readers are considered as *repaired* and carry new data in order to authenticate the legitimate tags of the system. Afterwards, each time that a repaired reader communicates with a tag $\mathcal{T}$, it transmits to $\mathcal{T}$ a counter v. This counter represents the number of updates achieved by the system (i.e., the number of times that a reader has been compromised). Once $\mathcal{T}$ receives a new value of v, it checks its trustworthiness and stores v's new value in its memory: $\mathcal{T}$ is now updated as well. Finally, when all the tags have stored the new value of v, the system privacy has been fully restored.

Note that, when a reader $\mathcal{R}$ is just corrupted (but not yet repaired), it is still a legitimate reader; once it is repaired, it implicitly means that another legitimate reader $\mathcal{R}'$ with the same identifier has been put into the system, and $\mathcal{R}$ is no longer legitimate.

[1] The detection of such a compromised reader is out of the scope of this thesis.

In Practice for the Privacy Evaluation. In the privacy experiment detailed in Section 6.1.4, four cases Z_1, Z_2, Z_3, and Z_4 appear when $\mathcal{A}$ is a CORRUPT adversary. They detail if $\mathcal{T}_0$ and $\mathcal{T}_1$ have been reached by the update.

(Z_1) $\mathcal{T}_0$ and $\mathcal{T}_1$ are updated. If Z_1 occurs, the probability that $\mathcal{A}$ differentiates $\mathcal{T}_0$ from $\mathcal{T}_1$ is $\frac{1}{2} + \epsilon(\lambda)$.

(Z_2) $\mathcal{T}_0$ and $\mathcal{T}_1$ are not updated. If Z_2 occurs, the probability that $\mathcal{A}$ differentiates $\mathcal{T}_0$ from $\mathcal{T}_1$ is 1.

(Z_3) $\mathcal{T}_0$ is updated and $\mathcal{T}_1$ is not updated. If Z_3 occurs, the probability that $\mathcal{A}$ differentiates $\mathcal{T}_0$ from $\mathcal{T}_1$ is 1.

(Z_4) $\mathcal{T}_0$ is not updated and $\mathcal{T}_1$ is updated. If Z_4 occurs, the probability that $\mathcal{A}$ differentiates $\mathcal{T}_0$ from $\mathcal{T}_1$ is 1.

Each tag update contributes to restore the system privacy. The privacy is completely restored when all the tags are updated. Consequently, we define $\mathfrak{m}(t)$ as a measurement at time t of the level of privacy restoration for $\mathcal{S}$. $\mathfrak{m}(t)$ is a non-increasing function depending on the four cases:

$$\mathfrak{m}(t) = \Big(\frac{1}{2} + \epsilon(\lambda)\Big)\ \Pr(Z_1) + \Pr(Z_2) + \Pr(Z_3) + \Pr(Z_4). \tag{6.1}$$

The four cases to draw $\mathcal{T}_0$ and $\mathcal{T}_1$ follow an hypergeometric distribution. The $u(t)$ function represents the number of tags updated at a given time t. Therefore:

$$\begin{aligned} \mathfrak{m}(t) \;=\; & \Big(\frac{1}{2} + \epsilon(\lambda)\Big)\Big(\frac{u(t)}{n}\Big)\Big(\frac{u(t)-1}{n-1}\Big) \\ & + \Big(1 - \frac{u(t)}{n}\Big)\Big(1 - \frac{u(t)}{n-1}\Big) + 2\Big(\frac{u(t)}{n-1}\Big)\Big(1 - \frac{u(t)}{n}\Big). \end{aligned} \tag{6.2}$$

The behavior of $\mathfrak{m}(t)$ is as follows. When t increases, so does $u(t)$. When $u(t) = 0$, then $\mathfrak{m}(t)$ is at its maximum of 1: the privacy restoration has not started yet. When $u(t)$ attains its maximum n (meaning that all the tags have been updated), then $\mathfrak{m}(t)$ reaches its lowest value of $\frac{1}{2} + \epsilon(\lambda)$: the privacy restoration has been fully done. Therefore, when $0 < u(t) < n$, $\mathfrak{m}(t)$ measures the current level of privacy restoration of $\mathcal{S}$ at time t.

6.3.3 Our Protocol

We propose a new challenge/response authentication protocol based on symmetric-key cryptography to fit the practical constraints set in Section 6.3.1. The main feature of the protocol is that the secret key shared by each tag and each reader is computed on-the-fly by the tag, as in TanSL. The secret key is thus unique for a given couple (tag, reader), instead of being a fixed long-term secret key associated to the tag.

Cryptographic Building Block. We only consider a symmetric-key cryptosystem (Enc/Dec) that is secure against key recovery (denoted "KR secure" in what follows).

Initialization. Let λ be the security parameter of the system. When the system is set up according to λ, a general counter v is initialized to zero. Each tag $\mathcal{T}$ is assigned with a unique identifier $\mathsf{ID}_{\mathcal{T}}$, a long-term secret key $\mathsf{k}_{\mathcal{T}}$, and a counter $\mathsf{v}_{\mathcal{T}}$, initialized to zero. Each reader $\mathcal{R}$ is assigned with the following values:

- a unique identifier $\mathsf{ID}_{\mathcal{R}}$,
- the general counter v, initialized to zero,
- for every tag $\mathcal{T}$, its identifier $\mathsf{ID}_{\mathcal{T}}$ and an encryption of its secret $\mathsf{s}_{\mathcal{TR}} = \mathrm{Enc}_{\mathsf{k}_{\mathcal{T}}}(\mathsf{ID}_{\mathcal{R}}||\mathsf{v})$.

This value $\mathsf{s}_{\mathcal{TR}}$ is a secret shared by $\mathcal{R}$ and $\mathcal{T}$ only. The back-end B stores $\mathsf{ID}_{\mathcal{R}}$, $\mathsf{ID}_{\mathcal{T}}$, $\mathsf{k}_{\mathcal{T}}$ and v.

Authentication. The authentication protocol is a 3-pass protocol as depicted in Figure 6.5. Note that when a check or a search does not succeed, the protocol is interrupted.

Update of the System. When a compromised reader is detected, the update protocol depicted in Figure 6.6 is carried out between the back-end B and each reader $\mathcal{R}$ of the system. We consider that all the readers are synchronized with the update at the same time as explained in Section 6.3.2.

Reader $\mathcal{R}$ $\mathsf{ID}_\mathcal{R}, \mathsf{v}, \mathsf{ID}_\mathcal{T}, \mathsf{s}_{\mathcal{TR}}$		**Tag** $\mathcal{T}$ $\mathsf{ID}_\mathcal{T}, \mathsf{k}_\mathcal{T}, \mathsf{v}_\mathcal{T}$
• $n_\mathcal{R} \in_R \{0,1\}^\lambda$		
	$\xrightarrow{\mathsf{ID}_\mathcal{R}, \mathsf{v}, n_\mathcal{R}}$	
		• Check $\mathsf{v} \geq \mathsf{v}_\mathcal{T}$ • $n_\mathcal{T} \in_R \{0,1\}^\lambda$ • $\mathsf{s}_{\mathcal{TR}} = \mathrm{Enc}_{\mathsf{k}_\mathcal{T}}(\mathsf{ID}_\mathcal{R} \| \mathsf{v})$ • $E = \mathrm{Enc}_{\mathsf{s}_{\mathcal{TR}}}(n_\mathcal{R} \| n_\mathcal{T})$
	$\xleftarrow{E}$	
• Find $(\mathsf{ID}_\mathcal{T}, \mathsf{s}_{\mathcal{TR}}) \in \mathsf{DB}_\mathcal{R}$ s.t. $n_\mathcal{R} \| n'_\mathcal{T} = \mathrm{Dec}_{\mathsf{s}_{\mathcal{TR}}}(E)$		
	$\xrightarrow{n'_\mathcal{T}}$	
		• Check $n'_\mathcal{T} = n_\mathcal{T}$ • $\mathsf{v}_\mathcal{T} = \mathsf{v}$

Figure 6.5: Authentication protocol.

Back-end B $\mathsf{ID}_\mathcal{R}, \mathsf{ID}_\mathcal{T}, \mathsf{k}_\mathcal{T}, \mathsf{v}$		**Reader** $\mathcal{R}$ $\mathsf{ID}_\mathcal{R}, \mathsf{v}, \mathsf{ID}_\mathcal{T}, \mathsf{s}_{\mathcal{TR}}$
• $\mathsf{v} = \mathsf{v}^{\mathrm{new}} = \mathsf{v} + 1$ • $\forall \mathcal{T} : \mathsf{s}^{\mathrm{new}}_{\mathcal{TR}} = \mathrm{Enc}_{\mathsf{k}_\mathcal{T}}(\mathsf{ID}_\mathcal{R} \| \mathsf{v}^{\mathrm{new}})$		
	$\xrightarrow[\{\mathsf{ID}_\mathcal{T}, \mathsf{s}^{\mathrm{new}}_{\mathcal{TR}}\}_{\forall \mathcal{T}}]{\mathsf{v}^{\mathrm{new}}}$	
		• $\mathsf{v} = \mathsf{v}^{\mathrm{new}}$ • $\forall \mathcal{T} : \mathsf{s}_{\mathcal{TR}} = \mathsf{s}^{\mathrm{new}}_{\mathcal{TR}}$

Figure 6.6: Update protocol.

6.4 Solution for Offline RFID Systems

In the previous section, a solution to face the problem of compromised readers in semi-offline RFID systems has been proposed. Yet today, ad-hoc RFID infrastructures with completely offline readers as the ones deployed for ephemeral events (e.g., culture shows, political conferences, sport events in isolated sites) are more than common. One example is a music festival organized by a city, where the concert areas are split in different districts. In such an event, it is possible to buy a ticket for the concerts, but also for the parking lot or the camping place. Connecting all so spread-out places together would not be economically affordable. Using another kind of system like GSM is neither reasonable because of its cost, communication rate, coverage, etc. Instead in such infrastructures, readers may be only connected once to the back-end server, during the setup procedure, and then kept offline.

This section goes a step further in the problem of compromised readers by presenting a solution for poorly connected environments such as offline RFID systems.

6.4.1 Offline Architecture

An offline RFID system $\mathcal{S}$ has the following properties. Once initialized by the system administrator, readers can no longer get connected to the back-end. Tags are not tamper-resistant. However, they are reasonably-costly, implying moderate capabilities in terms of calculation, communication and storage, although capable of verifying electronic signatures and processing symmetric-key cryptography, as explained in Section 6.4.7. An example is the SLE 66 family [86] where one tag costs less than 2 USD, yet containing a crypto engine supporting RSA and ECC.

6.4.2 Privacy-restoring Mechanism

To restore privacy in offline systems, the main idea of our solution is to integrate a privacy-restoring mechanism into the authentication protocol itself. This mechanism spreads the information that a reader has been compromised and repaired/updated to all the tags.

With this mechanism, the protocol is able to face an adversary who can corrupt up to α readers ($\alpha \leq m$) where m is the total number

of readers in the system. When only one reader is compromised, it is considered w.l.o.g. to be $\mathcal{R}_1$. It is assumed that the system administrator is able to detect such a compromised reader because it has been stolen or damaged. In such a case, he is expected to restore the security of $\mathcal{R}_1$ by renewing its secrets and credentials, including replacing the physical device, if needed.

Since all the devices are offline, $\mathcal{R}_1$'s repaired/updated information is injected into an arbitrary reader of the system. It is considered w.l.o.g. that the information is injected in $\mathcal{R}_1$ itself, after being repaired. Then the repaired $\mathcal{R}_1$ propagates its updated information to all its connecting tags. Thanks to their mobility, these "marked" tags, as mules, carry and propagate $\mathcal{R}_1$'s updated information to the readers they meet. These "marked" readers then propagate in turn $\mathcal{R}_1$'s updated information. It is possible to point out the parallel with viruses spread between floppy disks and computers (in the early age of computer science). Finally, when all the tags have received $\mathcal{R}_1$'s updated information, the system privacy has been fully restored.

As in Section 6.3.2, when a reader $\mathcal{R}$ is just corrupted (but not yet repaired), it is still a legitimate reader; once it is repaired, it implicitly means that another legitimate reader $\mathcal{R}'$ with the same identifier has been put into the system, and $\mathcal{R}$ is no longer legitimate.

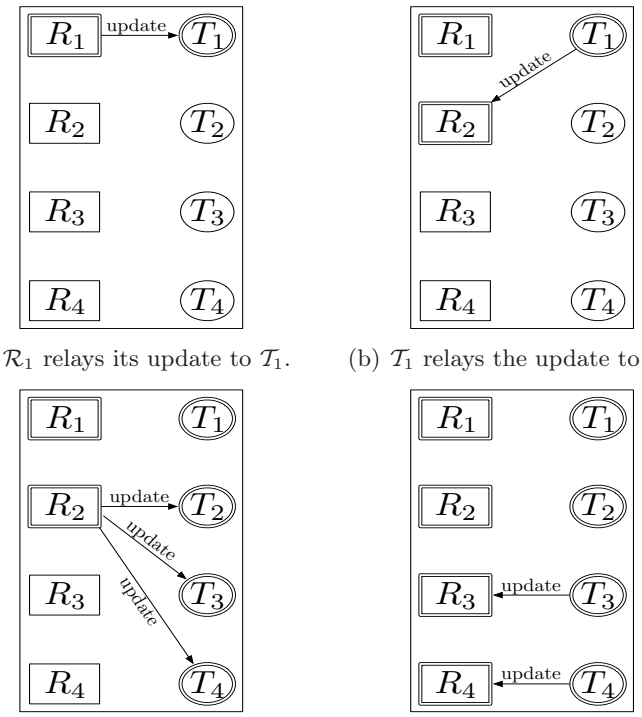
Remark. The information to spread is related to the fact that a reader has been compromised; but the propagation process can also be used to spread any kind of information (e.g., the new schedule of an event) whose authenticity must be ensured.

Figure 6.7 illustrates the behavior of the spread information with four readers and four tags. In the example, $\mathcal{R}_1$ has been compromised and repaired. As caption, a device drawn in a double-box means that this device carries the following information: "$\mathcal{R}_1$ has been compromised and repaired, and here are its new data".

As in Section 6.3.2, the privacy restoration follows the $\mathfrak{m}(t)$ measurement given in Eq.(6.2) (see Section 6.3.2 for more details).

6.4.3 Information Spread in the Literature

The main feature of the privacy-restoring mechanism relies on the propagation of the information that a reader has been compromised to all the



(a) $\mathcal{R}_1$ relays its update to $\mathcal{T}_1$.

(b) $\mathcal{T}_1$ relays the update to $\mathcal{R}_2$.

(c) $\mathcal{R}_2$ relays the update to $\mathcal{T}_2$, $\mathcal{T}_3$, $\mathcal{T}_4$.

(d) $\mathcal{T}_1$ and $\mathcal{T}_2$ relay the update to $\mathcal{R}_3$ and $\mathcal{R}_4$.

Figure 6.7: An example of information propagation with our protocol.

entities of the RFID system. This section surveys the information spread mechanisms in several disciplines. It presents analyses performed on data transmission from entities to entities in many domains, from medicine to computer viruses or networks. Note that the context of offline RFID systems is a particular framework that corresponds to bipartite graphs where readers can only communicate with tags, and vice-versa.

The earliest field dealing with information spread is epidemiology in medicine and biology, and more specifically epidemiological models with Bernouilli's precursory works [48]. This research topic aims to create mathematical methods to describe the transmission of contagious diseases through individuals of a population. In such models, it is important to clearly specify several points, such as the structure of the

population, the outbreak of the disease and the basic reproduction number. The epidemiological model that is the closest to our RFID system is the one for sexual transmission disease (STD) in an heterosexual population (i.e., bipartite population). The analogy with our framework is that there are two subgroups in the population (readers and tags), and the STD (the information about compromised readers) that can be spread between two elements of different subgroups, but never between two elements belonging to the same subgroup. Several results have been published for this epidemiological model [83, 114, 123, 147]. But most of them consider that an infected individual eventually dies or cures. This is not the case in our framework, since tags and readers never erase/ignore the information to spread. Therefore, the information spread in our framework is more efficient than a classical contagious disease in medicine.

Another key-domain of data propagation is virus spread in computer systems. In the early age of viruses, the spread was essentially processed via floppy disk exchanges. Therefore, the creation of trustworthy models has always been valuable for information security in order to better understand the spread of new viruses.

In the 1980s, Cohen was the pioneer in defining and describing computer viruses and spreading process [41, 42]. In 1991, Kephart and White proposed in [104] one of the first formalization of virus propagation for computers using mathematical epidemiological models. In 2004, Serazzi and Zanero provided a review of the most popular models about virus and worm spread [150]. Their study does not clearly match our problem though, as the main threat since the last decade has no longer been the viruses but the worms, which do not propagate following a bipartite graph. Finally in 2007, Omic, Kooij, and Van Mieghem investigated the propagation of viruses on a complete bipartite graph [135]. Their mathematical model is able to predict the probability of spread accurately. This work is very close to our framework, but the bipartite graph is not complete in our case, since not all the readers and tags communicate with each other. Their results show that the average number of infected nodes is around 85% when the spreading rate is high.

Correlated investigations on virus in RFID have been performed by Rieback, Crispo, and Tanenbaum [144], and by Rieback [143]. They demonstrate that a self-replicating RFID virus carried on an RFID tag

may infect the back-end server using SQL injections, but they do not consider the propagation efficiency in their work.

The idea of information spread has also been studied in mobile ad hoc networks and delay tolerant networks. The main motivation comes from the fact that node mobility can be used to forward and disseminate information in the whole system: a given mobile node, which carries some data, transmits them each time it meets another mobile node. This field is strongly connected to RFID systems, since the nodes are mobile, and therefore the propagation of the information can unexpectedly reach several parts of the system. In [166], Zhang proposes a survey of many "store-and-forward" routing protocols. Two opposed kinds of protocol can be found in the literature. They basically depend on the rate of message redundancy chosen for the spread. On one extreme, there are protocols like Epidemic [161], based on flooding: every node stores and propagates the information to every encountered node, and thus never erases the information. Clearly for such protocols, the success probability in delivering the information to the destination is very high, but it uses all the nodes to reach its goal and thus consumes many network resources (e.g., node memory, bandwidth). On the other extreme, there are protocols like Single-copy [156]: only one copy of the information is in circulation. Here, the consumption of network resources is very low, but the delay to deliver the information to its target can be very long. Between these two opposites, there are many protocols [81, 97, 110, 120, 152]. They are basically a trade-off between statistic profiles, delivery rate, energy consumption to optimize delivery delay.

To summarize, none of the given results is really exploitable in offline RFID systems, as this framework does not exactly fit the described ones. A strong deduction is that the information propagation mechanism in an environment is clearly specific to the application and its framework. Therefore, the following section presents a novel solution with its own information spread mechanism to restore privacy in offline RFID systems.

6.4.4 Our Protocol

We present a challenge/response authentication protocol inspired by a variant of the Needham-Schroeder public-key protocol [122] where the tag encrypts data with the reader public key. The main difference is that the secret encrypted by the tag is computed on-the-fly by the tag

and is unique for a given couple (tag, reader), instead of being a fixed long-term secret key associated to the tag.

Cryptographic Building Blocks. We consider a public-key cryptosystem (Enc/Dec), a signature scheme (Sign/Verif), and a MAC function. The cryptosystem is IND-CCA2 secure (see [23, 141] for more details about this scheme). The MAC and signature scheme verify the EF-CMA property [74].

Initialization. Let λ be the global security parameter of the system. When the system is set up according to λ, the back-end B receives a pair of public/private keys $(\mathsf{P}_B, \mathsf{K}_B)$. Each tag $\mathcal{T}$ is assigned with a unique identifier $\mathsf{ID}_\mathcal{T}$, and a long-term secret key $\mathsf{k}_\mathcal{T}$. Each reader $\mathcal{R}$ is assigned with the following values:

- a unique public identifier $\mathsf{ID}_\mathcal{R}$,
- a public version number $\mathsf{v}_\mathcal{R}$ of $\mathcal{R}$ certificate initialized to zero,
- a pair of public/private keys $(\mathsf{P}_\mathcal{R}, \mathsf{K}_\mathcal{R})$,
- its public data $\mathsf{d}_\mathcal{R} = (\mathsf{ID}_\mathcal{R} || \mathsf{v}_\mathcal{R} || \mathsf{P}_\mathcal{R})$,
- a lightweight certificate of its public data (signed by the back-end B) $\mathsf{C}_\mathcal{R} = \mathsf{d}_\mathcal{R} || \mathsf{t}_\mathcal{R} || S_{\mathsf{C}_\mathcal{R}}$, where $S_{\mathsf{C}_\mathcal{R}} = \mathrm{Sign}_{\mathsf{K}_B}(\mathsf{d}_\mathcal{R} || \mathsf{t}_\mathcal{R})$ is a signature produced by the back-end B, and $\mathsf{t}_\mathcal{R}$ is the creation timestamp of the certificate.

Memory Content. During the system initialization, back-end, readers and tags are loaded with their own values, and additional tables described below.

For every tag $\mathcal{T}$, every reader $\mathcal{R}$ stores the couple $(\mathsf{ID}_\mathcal{T}, \mathsf{s}_{\mathcal{T}\mathcal{R}})$ in its own database $\mathsf{DB}_\mathcal{R}$, where $\mathsf{s}_{\mathcal{T}\mathcal{R}}$ is the result of a MAC applied on $\mathsf{k}_\mathcal{T}$ combined with $\mathsf{ID}_\mathcal{R}$ and $\mathsf{v}_\mathcal{R}$ ($\mathsf{s}_{\mathcal{T}\mathcal{R}} = \mathrm{MAC}(\mathsf{k}_\mathcal{T} || \mathsf{ID}_\mathcal{R} || \mathsf{v}_\mathcal{R})$). This value $\mathsf{s}_{\mathcal{T}\mathcal{R}}$ is a secret computed by the back-end, and shared by $\mathcal{R}$ and $\mathcal{T}$ only. The reader also has a value $\mathsf{NewC}_\mathcal{R}$ that is the unique certificate containing all the new data for all the compromised and already repaired readers. At the system setup, $\mathsf{NewC}_\mathcal{R}$ is empty.

Then for every reader $\mathcal{R}$, every tag $\mathcal{T}$ stores the couple $(\mathsf{ID}_\mathcal{R}, \mathsf{v}_\mathcal{R})$ in its own database $\mathsf{DB}_\mathcal{T}$. Note that $\mathcal{T}$ does not store the certificate of every reader. $\mathcal{T}$ also has a value $\mathsf{NewC}_\mathcal{T}$ which has the same definition as $\mathsf{NewC}_\mathcal{R}$.

Finally, the back-end keeps the values $(\mathsf{ID}_\mathcal{R}, \mathsf{v}_\mathcal{R}, \mathsf{t}_\mathcal{R})$ of every reader $\mathcal{R}$ and $(\mathsf{ID}_\mathcal{T}, \mathsf{k}_\mathcal{T})$ of every tag $\mathcal{T}$.

Authentication. The authentication protocol is a 3-pass protocol as depicted in Figure 6.8. Note that when a check or a verification does not succeed, the protocol is interrupted.

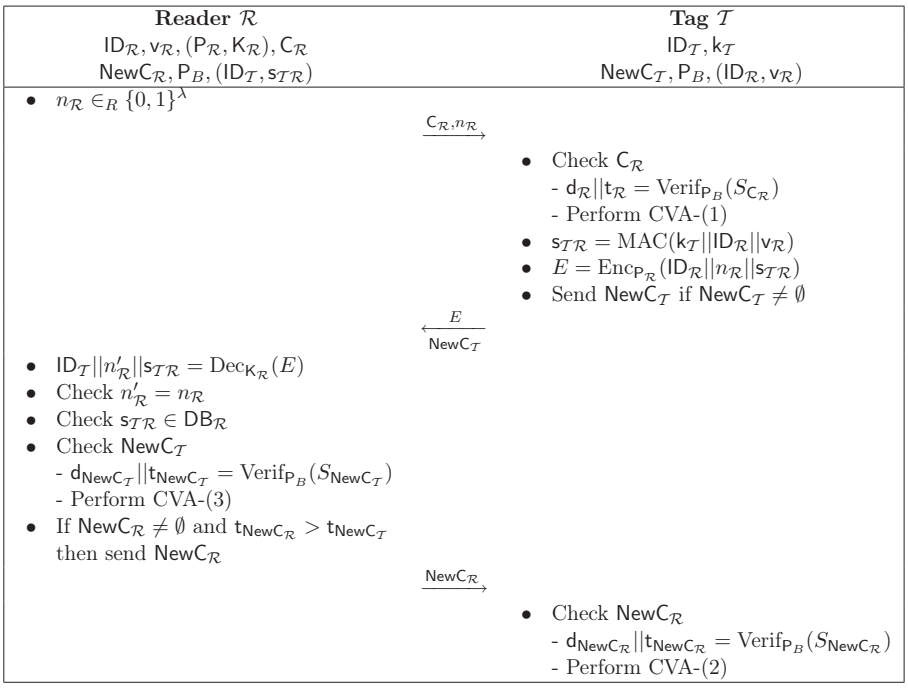


Figure 6.8: Authentication protocol.

The Certificate Verification Algorithm (CVA).

(1) When the tag $\mathcal{T}$ verifies the certificate $\mathsf{C}_\mathcal{R}$: $\mathcal{T}$ compares if $\mathsf{v}_\mathcal{R}$ is upper than the one stored in its database $\mathsf{DB}_\mathcal{T}$ for $\mathcal{R}$. If so, then: (i) $\mathcal{T}$ updates $\mathsf{v}_\mathcal{R}$ in $\mathsf{DB}_\mathcal{T}$, and (ii) $\mathcal{T}$ records the certificate

in $\mathsf{NewC}_{\mathcal{T}}$. If $\mathsf{v}_{\mathcal{R}}$ is lower than the one stored in $\mathsf{DB}_{\mathcal{T}}$ for $\mathcal{R}$, $\mathcal{T}$ interrupts the protocol.

(2) When the tag $\mathcal{T}$ verifies the certificate $\mathsf{NewC}_{\mathcal{R}}$: $\mathcal{T}$ compares if $\mathsf{t}_{\mathsf{NewC}_{\mathcal{R}}}$ is newer than the one of its stored $\mathsf{NewC}_{\mathcal{T}}$. If so, then: (i) $\mathcal{T}$ overwrites $\mathsf{NewC}_{\mathcal{T}}$ with the new received one (i.e., $\mathsf{NewC}_{\mathcal{R}}$), and (ii) $\mathcal{T}$ updates $\mathsf{v}_{\mathcal{R}_i}$ in $\mathsf{DB}_{\mathcal{T}}$ of every compromised and already repaired reader $\mathcal{R}_i$ cited in $\mathsf{NewC}_{\mathcal{R}}$.

(3) When the reader $\mathcal{R}$ verifies the certificate $\mathsf{NewC}_{\mathcal{T}}$: $\mathcal{R}$ compares if $\mathsf{t}_{\mathsf{NewC}_{\mathcal{T}}}$ is newer than the one of its stored $\mathsf{NewC}_{\mathcal{R}}$. If so, then $\mathcal{R}$ overwrites $\mathsf{NewC}_{\mathcal{R}}$ with the new received one (i.e., $\mathsf{NewC}_{\mathcal{T}}$).

Update of the System. When the system administrator realizes that the reader $\mathcal{R}_1$ has been compromised, he requires the back-end B to repair $\mathcal{R}_1$ by updating $\mathcal{R}_1$ data. Basically, B computes the following.

- Update $\mathsf{v}_{\mathcal{R}_1}^{\text{new}} = \mathsf{v}_{\mathcal{R}_1} + 1$.
- Update $(\mathsf{P}_{\mathcal{R}_1}^{\text{new}}, \mathsf{K}_{\mathcal{R}_1}^{\text{new}})$.
- Update $\mathsf{d}_{\mathcal{R}_1}^{\text{new}} = \mathsf{ID}_{\mathcal{R}_1} || \mathsf{v}_{\mathcal{R}_1}^{\text{new}} || \mathsf{P}_{\mathcal{R}_1}^{\text{new}}$.
- Create the new certificate $\mathsf{C}_{\mathcal{R}_1}^{\text{new}} = \mathsf{d}_{\mathcal{R}_1}^{\text{new}} || \mathsf{t}_{\mathcal{R}_1}^{\text{new}} || \text{Sign}_{\mathsf{K}_B}(\mathsf{d}_{\mathcal{R}_1}^{\text{new}} || \mathsf{t}_{\mathcal{R}_1}^{\text{new}})$.
- Compute the new values of $\mathsf{DB}_{\mathcal{R}_1}$:
$$\forall \mathcal{T} : \mathsf{s}_{\mathcal{T}\mathcal{R}_1} = \text{MAC}(\mathsf{k}_{\mathcal{T}} || \mathsf{ID}_{\mathcal{R}_1} || \mathsf{v}_{\mathcal{R}_1}^{\text{new}}).$$

$\mathcal{R}_1$ is also reinitialized with a new certificate in $\mathsf{NewC}_{\mathcal{R}_1}$. The latter contains in the clear all the couples $(\mathsf{ID}_{\mathcal{R}_i} || \mathsf{v}_{\mathcal{R}_i})$ for all the compromised and already repaired readers $\mathcal{R}_i$ (including $\mathcal{R}_1$), and only one unique signature of all these data: $\text{Sign}_{\mathsf{K}_B}(\{\forall \mathcal{R}_i : (\mathsf{ID}_{\mathcal{R}_i} || \mathsf{v}_{\mathcal{R}_i})\} || \mathsf{t}_{\mathsf{NewC}_{\mathcal{R}_1}})$, where $\mathsf{t}_{\mathsf{NewC}_{\mathcal{R}_1}}$ is the creation timestamp of this certificate. Finally, the system administrator changes/replaces $\mathcal{R}_1$ with these new values.

6.4.5 Security Analysis

The security analysis of our protocol is performed following the adversary classes defined in Section 6.1.3 and the security properties experiments given in Section 6.1.4.

Theorem 6.6. *If the public-key cryptosystem is IND-CCA2 secure, and if the MAC and signature scheme verify the EF-CMA property, then our RFID protocol ensures (i) availability and soundness against* CORRUPT-FORWARD *adversaries, (ii)* FORWARD*-privacy, and (iii)* CORRUPT-STANDARD*-privacy when the challenge tags are in the* Z_1 *case.*

Proof (Availability). In this proof, let us consider $\mathcal{A}$ as being the adversary of the availability experiment. We design an adversary $\mathcal{A}'$ taking advantage of $\mathcal{A}$ in order to break the EF-CMA property (as defined in [74]) of the underlying signature scheme (called SS here). Let $\mathcal{T}^*$ and $\mathcal{R}^*$ be respectively the tag and reader output by $\mathcal{A}$ at the end of $Exp^{\mathsf{Avail}}_{\mathcal{S},\mathcal{A}}$, and assume that $\mathcal{A}$ wins (i.e., $\mathcal{R}^*$ rejects $\mathcal{T}^*$). $\mathcal{R}^*$ starts the protocol by sending $n_{\mathcal{R}^*}$ and $\mathsf{C}_{\mathcal{R}^*}$. This certificate necessarily contains a valid signature, as this value can only be modified in $\mathcal{R}^*$'s memory by the system administrator. Consequently, $\mathcal{T}^*$ necessarily validates the first step of the $\mathsf{C}_{\mathcal{R}^*}$ verification. Then, depending on the result of CVA-(1), the two following cases are possible.

Case 1: $\mathcal{T}^*$ outputs the usual message E (and potentially $\mathsf{NewC}_{\mathcal{T}}$). E cannot be modified by $\mathcal{A}$ as she does not intervene during the protocol execution. $\mathcal{R}^*$ retrieves the value $\mathrm{MAC}(\mathsf{k}_{\mathcal{T}^*}||\mathsf{ID}_{\mathcal{R}^*}||\mathsf{v}_{\mathcal{R}^*})$ which is in $\mathsf{DB}_{\mathcal{R}^*}$, since it cannot be modified by anybody else than the system administrator. Thus, if $\mathcal{T}^*$ responds, it is necessarily accepted by $\mathcal{R}^*$.

Case 2: $\mathcal{T}^*$ interrupts the protocol, meaning that, according to CVA-(1), the received $\mathsf{v}_{\mathcal{R}^*}$ is lower than the one $\mathsf{v}'_{\mathcal{R}^*}$ stored in DB_{T^*}. This case only happens if $\mathcal{T}^*$ has received (either in a previous $\mathsf{C}_{\mathcal{R}^*}$, or in a previous $\mathsf{NewC}_{\mathcal{R}''}$) a signature on a message containing (at least) the couple $(\mathsf{ID}_{R^*}, \mathsf{v}'_{\mathcal{R}^*})$. As $\mathcal{R}^*$ is legitimate, the back-end cannot have produced this signature. This signature is thus necessary a forgery computed by $\mathcal{A}$.

Consequently, $\mathcal{A}'$ will use the signature included in the forged certificate sent by $\mathcal{A}$ to $\mathcal{T}^*$ (as explained in Case 2) to perform the EF-CMA experiment. Clearly, we have:

$$Adv^{\text{EF-CMA}}_{\text{SS},\mathcal{A}'} = Adv^{\mathsf{Avail}}_{\mathcal{S},\mathcal{A}} = \Pr(Exp^{\mathsf{Avail}}_{\mathcal{S},\mathcal{A}}[\lambda] \text{ succeeds})$$

which is negligible by the EF-CMA assumption of SS. □

Proof (Privacy). If $\mathcal{A}$ is FORWARD, the messages $\mathsf{NewC}_{\mathcal{R}}$ and $\mathsf{NewC}_{\mathcal{T}}$ are never sent, and the protocol is reduced to PK-Prot proved FORWARD-private as showed in Section 4.1.5.

We use the game technique as described by Shoup in [151] to prove the privacy of the protocol against CORRUPT-STANDARD adversaries. During this proof, we consider that (i) when a reader $\mathcal{R}$ is corrupted, then it is directly updated with new values, and that (ii) the challenge tags $\mathcal{T}_0$ and $\mathcal{T}_1$ are in the Z_1 case and cannot be corrupted.

Note that only the modifications of the original experiment are specified in the games: the unspecified queries stay unchanged.

Game 0: this game corresponds to the privacy experiment $Exp^{\mathsf{Priv}}_{\mathcal{S},\mathcal{A}}$ performed by the adversary $\mathcal{A}$ on the system $\mathcal{S}$ of security parameter λ. Let S_0 denote the event $b = b'$ in Game 0. Thus, we obviously have that $\Pr(S_0) = \Pr(Exp^{\mathsf{Priv}}_{\mathcal{S},\mathcal{A}}[\lambda]$ succeeds).

Let us assume that $\mathcal{A}$ performs q $\mathcal{O}^{\textsc{SendTag}}$ queries on the challenge tags $\mathcal{T}_0$ and $\mathcal{T}_1$. These queries are called "encryption queries" in the sequel of the proof. From Game 0, we introduce q transitions games established as follows.

Game i: this is the same game as Game $(i-1)$ except that one additional encryption query has been replaced as defined below (where $\mathcal{R}$ is a non-compromised reader in the experiment):

- the i^{th} first encryption queries are such that

$$E = \mathrm{Enc}_{\mathsf{P}_{\mathcal{R}}}(\mathsf{ID}_{\mathcal{R}}||n_{\mathcal{R}}||n_{\mathcal{T}}),$$

  where $n_{\mathcal{T}} \in_R \{0,1\}^{\lambda_{\mathrm{MAC}}}$ and λ_{MAC} is the output size of the MAC,
- the $(q-i)$ next encryption queries are such that

$$E = \mathrm{Enc}_{\mathsf{P}_{\mathcal{R}}}(\mathsf{ID}_{\mathcal{R}}||n_{\mathcal{R}}||\mathsf{s}_{\mathcal{T}\mathcal{R}}),$$

  where $\mathcal{T} = \mathcal{T}_0$ or $\mathcal{T}_1$ is the tag pseudonym given in argument to $\mathcal{O}^{\textsc{SendTag}}$.

Let S_i denote the event $b = b'$ in Game i, and $\Pr(S_i)$ denote its success probability.

Game $(i-1)$ to Game i (for $1 \leq i \leq q$) only differ from one encryption query. If the success probability of $\mathcal{A}$ is affected by such a modification, then a distinguisher $\mathcal{D}$ can use this difference to win the IND-CCA2 experiment (as defined in [141]) of the underlying cryptosystem (called CS here) used in the protocol.

For instance, let define a distinguisher $\mathcal{D}$ that simulates the system $\mathcal{S}$ to $\mathcal{A}$. For the $(i-1)$ first encryption queries asked by $\mathcal{A}$, $\mathcal{D}$ computes the answers E using a nonce $n_{\mathcal{T}}$. For the $(q-i)$ last encryption queries asked by $\mathcal{A}$, $\mathcal{D}$ correctly computes the answers E using $\mathsf{s}_{\mathcal{TR}}$ (where $\mathcal{T} \in \{\mathcal{T}_0, \mathcal{T}_1\}$, and $\mathcal{R}$ is a non-compromised reader). When $\mathcal{A}$ performs the i^{th} encryption query, $\mathcal{D}$ defines two messages $m_0 = (\mathsf{ID}_{\mathcal{R}}||n_{\mathcal{R}}||\mathsf{s}_{\mathcal{TR}})$ and $m_1 = (\mathsf{ID}_{\mathcal{R}}||n_{\mathcal{R}}||n_{\mathcal{T}})$, and sends them to the challenger $\mathcal{C}'$ of the IND-CCA2 experiment $Exp^{\text{IND-CCA2}}_{\text{CS},\mathcal{D}}$. Then, $\mathcal{D}$ transmits the answer $E = \text{Enc}_{\mathsf{P}_{\mathcal{R}}}(m_{b^*})$ of $\mathcal{C}'$ to $\mathcal{A}$, where b^* is the challenge bit of $Exp^{\text{IND-CCA2}}_{\text{CS},\mathcal{D}}$. From this answer, $\mathcal{A}$ undergoes Game $(i-1)$ if $b^* = 0$, or Game i otherwise. At the end of $Exp^{\text{IND-CCA2}}_{\text{CS},\mathcal{D}}$, $\mathcal{D}$ returns 1 if $\mathcal{A}$ wins the game (i.e., $b = b'$), 0 otherwise. Therefore, the probabilities of S_{i-1} and S_i can be expressed as follows:

$$\begin{aligned} \Pr(S_{i-1}) &= \Pr(\mathcal{D} \rightarrow 1|\ b^* = 0), \\ \Pr(S_i) &= \Pr(\mathcal{D} \rightarrow 1|\ b^* = 1). \end{aligned}$$

This implies:

$$|\Pr(S_{i-1}) - \Pr(S_i)| = Adv^{\text{IND-CCA2}}_{\text{CS},\mathcal{D}}.$$

Game q: this is the final game of this proof, where all the q encryption queries have been replaced by $\text{Enc}_{\mathsf{P}_{\mathcal{R}}}(\mathsf{ID}_{\mathcal{R}}||n_{\mathcal{R}}||n_{\mathcal{T}})$. Let S_{final} denote the event $b = b'$ in Game q, and $\Pr(S_{\text{final}})$ denote its success probability. Since the responses to all the encryption queries do not use any data specific to the challenge tags, $\mathcal{A}$ is not able to differentiate them: she can only answers at random $b = 0$ or 1. Therefore $\Pr(S_{\text{final}}) = \frac{1}{2}$.

From all the transitions, the conclusion is that:

$$\begin{aligned}\left|\Pr(Exp^{\mathsf{Priv}}_{\mathcal{S},\mathcal{A}}[\lambda] \text{ succeeds}) - \frac{1}{2}\right| &= \left|\Pr(S_0) - \Pr(S_{\text{final}})\right| \\ &= \Big|\Pr(S_0) - \Pr(S_1) + \Pr(S_1) - \ldots \\ &\quad - \Pr(S_{q-1}) + \Pr(S_{q-1}) - \Pr(S_{\text{final}})\Big| \\ &\leq \sum_{i=0}^{q-1}\left|\Pr(S_i) - \Pr(S_{i+1})\right| \\ &\leq q.Adv^{\text{IND-CCA2}}_{\text{CS},\mathcal{D}}\end{aligned}$$

which is negligible since the CS cryptosystem is assumed to be IND-CCA2 secure and q is a polynomial value. Consequently, the protocol ensures CORRUPT-STANDARD-privacy. □

Proof (Soundness). We also use the game technique of Shoup to prove the soundness of the protocol against CORRUPT-FORWARD adversaries. Note that $\mathcal{A}$ can corrupt any tag and reader, except the ones used to win the experiment, otherwise the attack is trivial. As a reminder, $\mathcal{A}$ chooses her target reader $\mathcal{R}$ at the beginning of the soundness experiment.

Game 0: this game corresponds to the soundness experiment $Exp^{\mathsf{Sound}}_{\mathcal{S},\mathcal{A}}$ performed by the adversary $\mathcal{A}$ on the system $\mathcal{S}$ of security parameter λ. Let S_0 denote the event "$\mathcal{A}$ successfully impersonates a non-compromised tag in front of $\mathcal{R}$" in Game 0. Thus, we obviously have that $\Pr(S_0) = \Pr(Exp^{\mathsf{Sound}}_{\mathcal{S},\mathcal{A}}[\lambda] \text{ succeeds})$.

Here, the "encryption queries" refer to all the $\mathcal{O}^{\text{SendTag}}$ queries on tags whose recipient is the target reader $\mathcal{R}$. The next steps of the proof are similar to the ones of the privacy proof: (i) we assume that $\mathcal{A}$ performs at most q' encryption queries, and (ii) from Game 0, we introduce q' transitions games based on the indistinguishability of the CS cryptosystem.

In the same vein as the privacy proof, the following equation holds for $1 \leq i \leq q'$:

$$|\Pr(S_{i-1}) - \Pr(S_i)| = Adv^{\text{IND-CCA2}}_{\text{CS},\mathcal{D}}.$$

At the end of these steps, the following game is obtained.

Game q': this is the game where all the q' encryption queries have been replaced by $\mathrm{Enc}_{\mathsf{P}_\mathcal{R}}(\mathsf{ID}_\mathcal{R}||n_\mathcal{R}||n_\mathcal{T})$. Let $S_{q'}$ denote the event "$\mathcal{A}$ successfully impersonates a non-compromised tag in front of $\mathcal{R}$" in Game q', and $\Pr(S_{q'})$ denote its success probability.

Now, from Game q', let us introduce nb_{tag} transitions (one per non-compromised tag). Each transition discards the possibility of $\mathcal{A}$ being able to forge the valid secret key $\mathsf{s}_{\mathcal{TR}} = \mathrm{MAC}(\mathsf{k}_\mathcal{T}||\mathsf{ID}_\mathcal{R}||\mathsf{v}_\mathcal{R})$ shared between a given non-compromised tag $\mathcal{T}$ and the target reader $\mathcal{R}$.

Game $(q'+j)$: this is the same game as Game $(q'+j-1)$ except that $\mathcal{A}$ cannot forge the valid secret key $\mathsf{s}_{\mathcal{TR}}$ of one additional non-compromised tag:

- $\mathcal{A}$ cannot forge the valid secret key of the j^{th} first non-compromised tags,
- $\mathcal{A}$ can forge the valid secret key of the $(nb_{\mathrm{tag}}-j)^{\mathrm{th}}$ last non-compromised tags,

Let $S_{q'+j}$ denote the event "$\mathcal{A}$ successfully impersonates a non-compromised tag in front of $\mathcal{R}$" in Game $(q'+j)$, and $\Pr(S_{q'+j})$ denote its success probability.

The transition from Game $(q'+j-1)$ to Game $(q'+j)$ (for $1 \leq j \leq nb_{\mathrm{tag}}$) is based on the failure event F being "$\mathcal{A}$ can forge the secret key of the j^{th} non-compromised tag". This means that these two consecutive games proceed identically unless F occurs. Applying Lemma 1 given by Shoup in [151], we have that:

$$|\Pr(S_{q'+j-1}) - \Pr(S_{q'+j})| \leq \Pr(F) = Adv^{\text{EF-CMA}}_{\mathrm{MAC},\mathcal{A}}.$$

Game $(q'+nb_{\mathrm{tag}})$: this is the final game of this proof, where all the q' encryption queries have been replaced by $\mathrm{Enc}_{\mathsf{P}_\mathcal{R}}(\mathsf{ID}_\mathcal{R}||n_\mathcal{R}||n_\mathcal{T})$, and where $\mathcal{A}$ cannot forge the valid secret key of any non-compromised tag. Let S_{final} denote the event "$\mathcal{A}$ successfully impersonates a non-compromised tag in front of $\mathcal{R}$" in Game $(q'+nb_{\mathrm{tag}})$, and $\Pr(S_{\mathrm{final}})$ denote its success probability. Since $\mathcal{A}$ cannot recover/forge a valid secret key of a non-compromised tag, she can only send a random message E to $\mathcal{R}$: therefore $\Pr(S_{\mathrm{final}}) \leq \epsilon(\lambda)$.

From all the transitions, the conclusion is that:

$$\begin{aligned}
\Pr(Exp^{\mathsf{Sound}}_{\mathcal{S},\mathcal{A}}[\lambda] \text{ succeeds}) &= \Pr(S_0) - \Pr(S_1) + \Pr(S_1) - \ldots - \Pr(S_{q'}) \\
&\quad + \Pr(S_{q'}) - \ldots - \Pr(S_{\text{final}}) + \Pr(S_{\text{final}}) \\
&\leq \Pr(S_{\text{final}}) + \sum_{i=0}^{q'-1} \Big|\Pr(S_i) - \Pr(S_{i+1})\Big| \\
&\quad + \sum_{j=0}^{nb_{\text{tag}}-1} \Big|\Pr(S_{q'+j}) - \Pr(S_{q'+j+1})\Big| \\
&\leq \epsilon(\lambda) + q'.Adv^{\text{IND-CCA2}}_{\text{CS},\mathcal{D}} + nb_{\text{tag}}.Adv^{\text{EF-CMA}}_{\text{MAC},\mathcal{A}}
\end{aligned}$$

which is negligible by the IND-CCA2 and EF-CMA assumptions and since q' and nb_{tag} are polynomial values. □

6.4.6 Efficiency Analysis: Practical Case Study of a 3-Day Sport Event

To illustrate the efficiency of our protocol in offline infrastructures, let consider a real-life RFID ticketing system deployed by RFIDea [142] during a 3-day automobile race in 2010. The event area is in the countryside, and stretches over several square kilometers, where deploying wired or wireless networks would not be economically affordable for a single event. Using GSM is neither realistic due to the cost, communication rate, coverage, etc.

This system is composed of 55 readers and 102 110 tags. Each tag stands for a badge that allows a person to attend the event. Readers are initialized once by the system administrator and then given to the agents in the field until the end of the event. Without any means to communicate with the back-end, those handled readers must carry the tags secret keys to be able to authenticate all spectators' and employees' badges. Agents are not mobile, while spectators and employees are. Spectators may move inside the area, e.g., from a Silver zone to a Gold zone according to the rights of their tickets, but they can also leave and return to the event area whenever they want. The RFID system is thus suitable to easily manage the mobility of all the participants to the event.

The last day is the most important one: the first two days are the training and qualification days, and the last day is the race. Not all the

tags work during all the event: some spectators may only want to see the final competition, while some employees may only work some days. Therefore, tags categories are 1/2/3-day tickets.

We highlight that one ticket is typically sold between USD 200 and USD 700. For the event organizer, the factory price of one ticket essentially includes the tag manufacturing process, the customizing of the paper ticket and tag content, and the shipment of the ticket to the customer. The cost of the microcircuit is therefore negligible and can be easily included in the total price of the ticket.

The fear of compromised readers is here realistic: for example, several readers were stolen during the same event in 2009. Spectators and employees were hence traceable, since no adequate solution to handle that problem had been put into place at that time. Also, the ticketing company agrees to spend time dealing with privacy before the event, but certainly not during the D-Day rush.

Experimental Conditions. The theoretical propagation of our protocol has been simulated when one reader has been compromised, using the readers logs of that event. The RFID application managing the event is designed in such a way that its system administrator was only able to supply the last log of every tag for every reader. Hence, if a tag has been authenticated twice by the same reader, then its last authentication was only known: the presented results are a lower bound for the propagation. Holding all the logs would allow us to show better performances.

Analysis of the Experimental Results. Let us first analyze the speed of the information spread. $\mathcal{R}_1$ denotes the most used reader of the event. Let consider that the update of the system is performed on $\mathcal{R}_1$, and that the time at which the update is put into place is called the ISSP (information spread starting point).

Figure 6.9 depicts the spread speed of the update when $\mathcal{R}_1$ launches the spread: the plotted data are the number $u(t)$ of tags updated at time t. The stable periods represent the two nights of the event, where the event venue is closed, i.e., the information spread is in stand-by. The propagation increases from one day to another: the more the tags are updated, the better the propagation is. The number of tags updated at the end of the event is 101 637 when the ISSP is at the beginning of the

event: this is 99.5% of the total number of tags in the system.

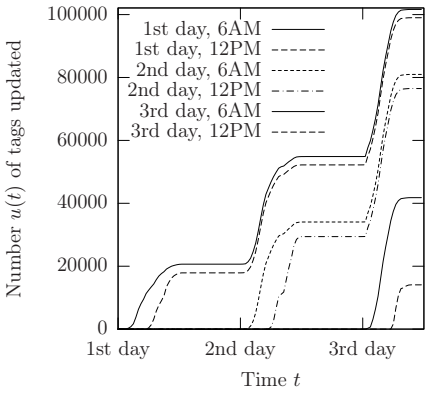


Figure 6.9: Propagation result at time t, depending on the ISSP, for $\mathcal{R}_1$.

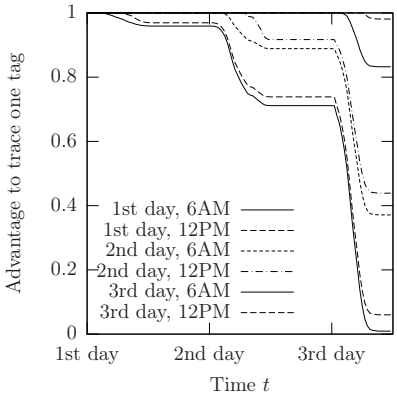


Figure 6.10: Practical advantage of success to trace one tag knowing $u(t)$ at time t, depending on the ISSP, for $\mathcal{R}_1$.

Then, let us evaluate the adversary practical advantage to trace one tag when the ISSP is done by $\mathcal{R}_1$. Figure 6.10 is the plot of the theoretical advantage from Eq.(6.2), with the practical results of $u(t)$ for the propagation behavior for $\mathcal{R}_1$. The curves represent this advantage at time t, depending on the ISSP. Like $\mathfrak{m}(t)$'s behavior, the advantage to trace one tag is a monotonically non-increasing function on $u(t)$. When the ISSP is at the beginning of the event, since almost all the tags are updated at the end, then the advantage to trace one tag is close to 0. The same test of update spread has been run for every reader of the system. The result is that the spread has the same behavior for every reader in average.

In fact, the analysis on the whole three days does not illustrate fairly the spread efficiency because of the 1-day tickets. Actually, if the ISSP is at 6AM the third day, the tags out of circulation during this day will never receive the update, and they will bias the propagation results. The later the ISSP is, the more the set of never-updated tags will increase. For example in Figure 6.9, the curves related to the ISSP done the third day show that the update does not reach half of the tags. In reality, only 47 694 tags over 102 110 were in circulation during the last day of the

event. Consequently, we analyze the information spread done on $\mathcal{R}_1$ only during this last day, and show trustworthy results when only considering the set $\mathcal{N}$ of these 47 694 tags. Figure 6.11 depicts the advantage to trace one tag knowing the percentage of updated tags of $\mathcal{N}$. If the ISSP is during the morning, this advantage fluctuates between 0.2 and 0.3.

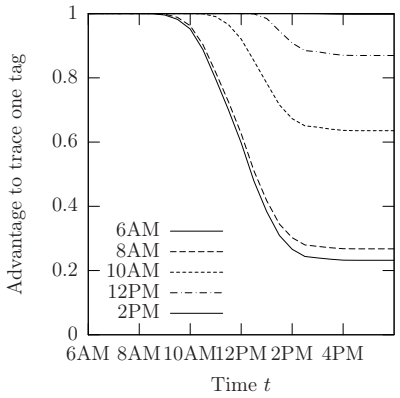


Figure 6.11: Practical advantage to trace one tag knowing $u(t)$ in comparison to the number of tags in circulation the last day, at time t of the last day, depending on the ISSP, for $\mathcal{R}_1$.

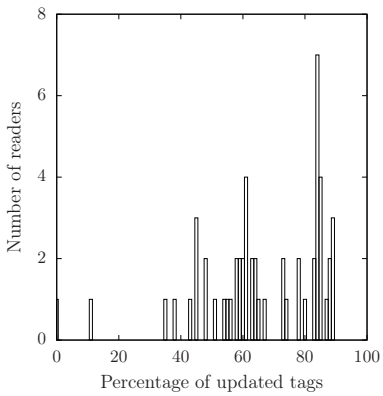


Figure 6.12: Number of readers that cause the update of x% of the tags in circulation the last day, at the end of the last day, when the ISSP is at 6AM the last day.

Finally, let us study the propagation behavior on the set $\mathcal{N}$ during the last day, wherever the ISSP is put into place. First, for every reader $\mathcal{R}$, the percentage x of updated tags of $\mathcal{N}$ at the end of the last day has been measured when the ISSP is done at 6AM that day by $\mathcal{R}$. Then, for each percentage x, the number of readers that cause this percentage x are counted, and the results are plotted in the histogram of Figure 6.12. Clearly more than 80% of the readers cause the update of at least 50% of the tags in $\mathcal{N}$; more than one third of the readers cause the update of at least 80% of the tags in $\mathcal{N}$. A side result is that two readers cause the update of only 0.004% and 11% of these tags: these low percentages may result from either a broken-down reader, or an employees' reader (no so much used). This result shows that the choice of the reader to start the ISSP does not really influence the spread among the tags in $\mathcal{N}$

during the last day.

In conclusion, our results show that the propagation can be efficient in our real-life 3-day sport event. The advantage to trace one tag significantly decreases if the ISSP is at the beginning of the event. If the ISSP is during the second day, around 80% of the tags are reached by the information. Considering only the tags in circulation the last day, the advantage to trace one tag during this day is low. This is a good outcome for this specific scenario. As final remark, do not forget that these results are the lower bound of the propagation: if we had all the logs of the event, the propagation would have still better results.

6.4.7 Practical Considerations

The aim of this work is to provide a practical and easily deployable protocol. The choice of the algorithms and parameters are consequently discussed below: they demonstrate that our protocol is compliant with real-life constraints. To do so, let consider two off-the-shelf tags, namely SLE 66CLX 360P [86] and NXP JCOP41 [127]. In what follows, the numerical values provided about SLE come from the specification of the device, while the numerical values about JCOP come from our own practical tests applied to a JCOP41 v.2.2.1.

The Cryptographic Building Blocks. Let set the security parameter λ as being the pair $(\lambda_{\mathrm{PK}}, \lambda_{\mathrm{SK}})$, where $\lambda_{\mathrm{PK}} = 1024$ bits and $\lambda_{\mathrm{SK}} = 128$ bits. An adequate public-key cryptosystem for our protocol is RSA-OAEP, given that IND-CCA2 property is required. We also suggest RSA-PSS, which is EF-CMA, as signature scheme. Choosing a 1024-bit RSA modulus and a 17-bit exponent appears to be a fair trade-off between security and efficiency in our protocol [115]. Following these choices, Table 6.1 gives the sizes of the values involved in our protocol.

Data	$\mathsf{ID}_{\mathcal{T}}, \mathsf{ID}_{\mathcal{R}}, \mathsf{v}_{\mathcal{R}}$	$\mathsf{t}_{\mathcal{R}}$	$n_{\mathcal{R}}$	$\mathsf{k}_{\mathcal{T}}$	$\mathsf{P}_{\mathcal{R}}$	$\mathsf{C}_{\mathcal{R}}$	E
Size	32	16	64	128	1041	2145	1024

Table 6.1: Data sizes in bits of our protocol, with $\lambda_{\mathrm{PK}} = 1024$ bits and $\lambda_{\mathrm{SK}} = 128$ bits.

Tag Memory. Each tag $\mathcal{T}$ is required to store its identifier $\mathsf{ID}_{\mathcal{T}}$, its long-term secret key $\mathsf{k}_{\mathcal{T}}$, the back-end public key P_B, the database $\mathsf{DB}_{\mathcal{T}}$ containing all the pairs $(\mathsf{ID}_{\mathcal{R}}, \mathsf{v}_{\mathcal{R}})$ of the m readers. Additionally, each tag may need to store a certificate $\mathsf{NewC}_{\mathcal{T}}$, which contains (i) the pairs $(\mathsf{ID}_{\mathcal{R}}, \mathsf{v}_{\mathcal{R}})$ of α readers, (ii) the timestamp of this certificate, and (iii) the signature of all these values, where α readers have been compromised.

From Table 6.1, the size of the tag EEPROM must be at least $64(m+\alpha)+2241$ bits. When $m = 55$ and $\alpha = 10$, which are fairly realistic values, a 6401-bit EEPROM is needed, that is about 0.8KB. This widely fits the EEPROMs of the two considered tags, namely SLE and JCOP, which respectively offer 36KB and 72KB.

Remark. Two improvements may be added to our solution to decrease the size of the tag memory needed. First, in the setup phase of the protocol, every $\mathsf{v}_{\mathcal{R}}$ is initialized to zero, and every tag $\mathcal{T}$ stores every pair $(\mathsf{ID}_{\mathcal{R}}, \mathsf{v}_{\mathcal{R}})$ in $\mathsf{Tab}_{\mathcal{T}}$. If only a few readers are compromised, then one better strategy can be adopted to refine the storage of these values: $\mathcal{T}$ may store only one $\mathsf{v}_{\mathcal{R}}$ for the non-compromised readers, and may store the complete pairs $(\mathsf{ID}_{\mathcal{R}}, \mathsf{v}_{\mathcal{R}})$ for the compromised and already repaired ones. Second, when considering that the readers identifiers are random values, the storage of the pairs $(\mathsf{ID}_{\mathcal{R}}, \mathsf{v}_{\mathcal{R}})$ on $\mathsf{DB}_{\mathcal{T}}$ is memory-consuming. But if these identifiers are sequential, then $\mathcal{T}$ only needs $\lceil \log_2(m) \rceil$ bits of memory to store them. With these two improvements, the required memory is as low as 3851 bits, that is around 0.482KB, using the numerical values provided in Table 6.1.

Transmission Time. During one protocol execution, the data exchanged between a reader and a tag are the reader certificate $\mathsf{C}_{\mathcal{R}}$ and nonce $n_{\mathcal{R}}$, the tag answer E, and the values $\mathsf{NewC}_{\mathcal{T}}$, $\mathsf{NewC}_{\mathcal{R}}$, if needed. In the worst case, the number of bits exchanged is hence $64\alpha + 5313$, which equals 5953 bits when $\alpha = 10$. The average data rate of SLE is 424 bits/ms, yielding a total transmission time equal to 14.04ms. For our JCOP tests, the transmission of 5953 bits takes around 68.04ms.

Tag Computation Time. A tag has to compute its answer E, and the verification of the two certificates $\mathsf{C}_{\mathcal{R}}$ and $\mathsf{NewC}_{\mathcal{R}}$ (i.e., the unique update certificate). These computations do not depend on the number of compromised readers. The calculation time at 15MHz for an RSA

operation (encryption and signature verification) when $|N| = 1024$ and $|e| = 17$ is 7ms for SLE [86]. Therefore, the expected calculation time is here 21ms. The performances of our JCOP for the same test values $|N|$ and $|e|$ are 104.2ms for signature verification, and 123.3ms for encryption. Thus, the expected calculation time on our JCOP is 331.7ms.

Chapter 7

Untraceability Model for RFID

After several years of research on cryptographic models for privacy in RFID systems as surveyed in Chapter 3, it appears that no universally accepted model has been designed yet. Experience shows that security experts usually prefer to use their own ad-hoc model (e.g., in Chapter 2) than the existing ones which are perceived as rigid and intricate. Furthermore, the study performed in Chapter 4 pointed out the technical drawbacks of the eight most famous models published so far. In particular, their lack of comprehensiveness and their impossibility to refine the privacy assessment of different protocols has been highlighted.

In this chapter, we first present additional arguments that emphasize the necessity to define a new model capable of comparing protocols meaningfully. We consequently issue an untraceability model that is operational where the previous models were not. The model aims to be easily understandable and manageable. This spirit led to a modular model where adversary actions (*oracles*), capabilities (*selectors* and *restrictions*), and goals (*experiment*) emerge in a way that is natural and intuitive. This design enhances the ability to (i) formalize new adversarial assumptions (such as time attacks or compromised readers introduced in Chapters 5 and 6) and future evolutions of the technology, and (ii) provide a finest privacy evaluation of the protocols.

7.1 The Need of a New Model

Based on the scientific literature related to RFID referenced in [8] (more than 800 articles), we performed a quantitative survey on the usage of privacy models for formal analyses. It appears that only 33 articles exploit one of the existing models presented in Chapter 3 as is to carry out privacy proofs and/or attacks, and the majority of authors (i.e., 20 articles) opts for the Vaudenay model [162] in such a case. Note however that 39 other articles use variants of the existing models, in particular variants of the JW model[1] because of its appreciated user-friendliness.

Unfortunately, even if the Vaudenay model is attested to be the most comprehensive RFID model issued so far, Chapter 4 showed that this model is unable to distinguish protocols that intuitively ensure different privacy levels. In particular, the Vaudenay model grants the SK-Prot, MW-Prot, and O-FRAP protocols the same privacy level, namely WEAK-privacy, although these three protocols are built on different underlying key infrastructures (i.e., unique long-term secrets for SK-Prot, correlated secrets for MW-Prot, and key-update mechanism for O-FRAP). Slight modifications or extensions of the model would not be enough to overcome these differentiation issues. A fundamental restructuring of the Vaudenay model should be undertaken in order to fairly distinguish these protocols as sketched below.

Privacy Experiment. The inability to distinguish the SK-Prot and MW-Prot protocols mainly results from the definition of the privacy experiment. Indeed, all the tags created by the adversary during the experiment are "challenge tags" (as explained in Section 4.2.1) and are thus subject to the same attacks that might be performed by the adversary. Consequently, the Vaudenay model cannot formalize the intermediate "real-life" adversary defined in Sections 4.1.2 and 4.1.6 who can be interested in tampering with her own tag to trace other users when the system protocol is based on correlated secrets.

In order to capture this adversary in [162], it would be necessary to (i) modify the privacy experiment such that it naturally defines two types

[1] For instance, the JW privacy experiment can be refined from the original one to only fit the specifications of the analyzed system. Such an example can be found in Section 6.1.4.

of tag (challenge and non-challenge ones) and such that the adversary can only win the experiment with an analytical conclusion on the challenge tags, and (ii) define the adversary classes such that the adversary is not mandatory to follow the same rules for both challenge and non-challenge tags.

Free Tags. The inability to distinguish SK-Prot and O-FRAP arises from the definition of free tags. This concept represents a tag that is not in the adversary neighborhood. During a so-called blinded phase, it is thus impossible for the adversary to communicate with a free tag. Since the adversary is the only entity that can interact with the tags of the system in [162], then no legitimate authentication occurs on a free tag.

Yet in real-life scenarios, tags may perform complete legitimate authentications when they are out the adversary line of sight [50, 55, 109]. When considering protocols like O-FRAP, key-update mechanisms carried out at the end of each legitimate authentication are meant to strengthen system privacy in such scenarios as detailed in Section 4.1.6. The Vaudenay model does not capture this privacy nuance, though.

In order to formalize such realistic scenarios in [162], it would be necessary to (i) define an honest entity that will operate on the free tags, (ii) precisely establish the way this honest entity manages the potential legitimate authentications of the free tags, and (iii) provide a new notion of privacy that encapsulates such scenarios.

Additionally, [162] only deals with RFID architectures that consist of a unique reader coupled with the system database. This assumption is quite restrictive and avoids analyzing many practical RFID systems, in particular when the readers are not always fully connected to the back-end as highlighted in Chapter 6 and in [68]. This restriction further prevents dealing with large-scale RFID systems where the readers are not always synchronized, as detailed in Section 7.2.1.

We consequently build a model able to compare protocols meaningfully which considers different potential system architectures, and which nevertheless benefits from the interesting features present in previous models, namely:

- the Universal/Existential untraceability concepts introduced in the Avoine model [9],

- the unambiguous and user-friendly indistinguishability-based experiment suggested in the JW model [99], and
- the notion of adversary classes proposed in the Vaudenay model [162].

7.2 RFID System

Chapter 3 surveyed the eight most well-known RFID privacy models published in the literature. These models generally consider that an RFID system is only based on an anonymous identification protocol involving a single reader and several tags. Yet, as highlighted in Chapter 6, such a restriction on the number of readers has been clearly identified as too strong to fit reality. Additionally, restraining the system purpose to anonymous identification is also an unnecessary boundary, given that the evolution of RFID leads us nowadays to be able to analyze the privacy of other kinds of system (e.g., ownership transfer systems).

This section defines the three building blocks composing an RFID system $\mathcal{S}$, namely the *architecture*, the *initialization procedures*, and the *protocol(s)* executed by the entities of the system.

7.2.1 System Architecture

RFID privacy models presented in Chapter 3 generally consider that a system consists of readers connected to a central back-end through "secure channels" which can be defined as follows.

Definition 7.1 (Secure Channel). *A channel is said to be* secure *if it ensures the properties (defined in [10]) of confidentiality, authentication, integrity, liveliness and sequentiality.*

Under the assumption that channels between the back-end and the readers are secure, the existing models consider that all these entities can be reduced to a single autonomous one. However, this reduction can be too drastic and may contrast with practical scenarios. For instance, spreading new data in a large-scale multi-reader system may take several hours, even if such channels are secure. During this spreading period, readers may be desynchronized, which is in contradiction with the "single-entity reduction". This reduction can be performed, though, if

the channels between the back-end and the readers are "instantaneous secure", as defined below.

Definition 7.2 (Instantaneous Secure Channel). *A secure channel is said to be* instantaneous *if the sender is assured that the packet will be delivered to the correct receiver within a negligible time.*

Hence, a system architecture can be reduced to a single autonomous reader if and only if (i) either the channels between the back-end and the readers, and the channels between the readers are instantaneous secure, (ii) or the readers are undesynchronizable, e.g., the information they contain is never updated. This justifies the need to introduce two categories of system architectures.

Definition 7.3 (Single-Entity (SE) Architecture). *An RFID system architecture is said to be* single-entity (SE) *if it can be reduced to a single autonomous reader.*

Definition 7.4 (Multiple-Entity (ME) Architecture). *An RFID system architecture is said to be* multiple-entity (SE) *if it cannot be reduced to a single autonomous reader.*

Interestingly, some well-known protocols (built for SE architectures) are no longer correct when considered in ME architectures. An adversary could take advantage of this issue to easily undermine their privacy, as illustrated in Example 7.5. As far as we know, this is the first time this problem has been raised in RFID privacy models.

Example 7.5. The OSK protocol [134] is correct in an SE architecture, but an active adversary can perform a desynchronization attack [99]. In an ME architecture where the readers update their data after each successful identification, OSK is no longer correct, and a simple passive adversary observing whether or not the identification succeeds is able to distinguish the tags.

7.2.2 Initialization Procedures

In the model, an RFID system $\mathcal{S}$ is setup by a procedure InitSystem that (i) generates the public and private values of $\mathcal{S}$ depending on its

security parameter λ, and (ii) initializes the readers and the potential back-end according to the chosen architecture. To allow a dynamic generation of tags, a procedure CreateTag is defined and called aside of InitSystem. Tags can potentially be setup with unique data and must consequently be registered in $\mathcal{S}$ to be recognized afterwards. This action is not necessarily performed when the tag is created, and then requires the definition of an independent procedure RegisterTag.

The three procedures determine each entity initial *internal state*, which is the data stored in the entity memory. The reader internal state may furthermore contain the system database (e.g., in an SE architecture). For the sake of clarity, the i^{th} reader (resp. tag) is assigned, when created, to an arbitrary unique label, denoted $\mathcal{R}_i$ (resp. $\mathcal{T}_i$), or simply $\mathcal{R}$ (resp. $\mathcal{T}$) when no ambiguity occurs.

7.2.3 Protocols

The behaviors and interactions of the system entities are described through protocol(s). Each of them determines the actions (e.g., computation, random generation) that the entities have to perform to reach a given objective. The following definition is based on the one provided in [116].

Definition 7.6 (Protocol). *A* protocol*, denoted* Prot*, involving one or more entities, describes a sequence of steps interleaved by transitions to achieve a well-defined objective. Each step defines a list of actions that must be performed by a specific entity. Each transition defines an event (internal or external) required by one involved entity to move to the next step.*

Each entity involved in the protocol executes its own *algorithm* to reach the protocol objective.

Definition 7.7 (Algorithm). *The* algorithm *of an entity for a protocol* Prot *refers to the subset of steps and transitions that are assigned to this entity in the protocol* Prot*.*

The execution of an algorithm is as follows. By default, an entity $\mathcal{X}$ is in its initial step waiting for an event either internal (e.g., a timeout) or external (e.g., presence of a tag in the electromagnetic field of a reader, reception of a message). This event activates the first transition of $\mathcal{X}$

and thus starts an algorithm execution. This execution is ended when an event (internal and/or external) activates a transition that brings $\mathcal{X}$ back to its initial state, typically when the protocol objective is reached. To avoid frozen executions, each step should have a default transition that closes the execution, moving $\mathcal{X}$ back to its initial step when an unexpected event happens (e.g., a timeout for readers as defined in the ISO/IEC 14443 standard [93]).

During an algorithm execution, an entity may exchange several data with the external world, typically with other involved entities. This external view of an algorithm execution can be enriched by auxiliary information extracted from the activation of transitions, for instance it can be the reception/emission time of a message or its recipient/issuer. Each algorithm execution defines a data set called "transcript".

Definition 7.8 (Transcript). *The* transcript *of an algorithm execution is a set of data that contains (i) the information that activates the transitions based on external events, and (ii) the auxiliary information related to these external events.*

Transcripts are indexed with a counter incremented each time $\mathcal{X}$ engages in a new algorithm execution. $\pi^i_{\mathcal{X},\text{PROT}}$ denotes the transcript of the i^{th} execution of $\mathcal{X}$'s algorithm related to PROT.

Each algorithm execution may further modify the entity internal state. Some collecting methods can be applied to capture the information contained in an entity internal state (e.g., tampering attacks). Each capture defines a data set called "snapshot".

Definition 7.9 (Snapshot). *A* snapshot *of an entity internal state is the set of all the data stored in the entity memory at the time a collecting method is applied.*

Snapshots are also incrementally indexed, and $\varepsilon^i_{\mathcal{X}}$ denotes the i^{th} snapshot of $\mathcal{X}$'s internal state.

7.3 Adversary

The model formalizes the actions that can be carried out on an RFID system $\mathcal{S}$ with *oracles*. The oracles allow an adversary $\mathcal{A}$ to interact with

$\mathcal{S}$ and collect data. The model also defines the adversary capabilities to access the collected data, using the concept of *selectors*.

7.3.1 Oracles

The three families of oracles that can be queried on an already initialized system are defined below.

(i) Oracles that dynamically add tags to the system $\mathcal{S}$

- $\mathcal{O}^{\text{CreateTag}}() \rightarrow \mathcal{T}$: executes the CreateTag procedure, and returns the label $\mathcal{T}$.
- $\mathcal{O}^{\text{RegisterTag}}(\mathcal{T}) \rightarrow \emptyset$: executes the RegisterTag procedure on the tag $\mathcal{T}$.

(ii) Oracles that completely or partly execute a protocol

- $\mathcal{O}^{\text{Execute}}(\text{Prot}, \mathcal{X}_1, \ldots, \mathcal{X}_\alpha) \rightarrow (\pi^{i_1}_{\mathcal{X}_1,\text{Prot}}, \ldots, \pi^{i_\alpha}_{\mathcal{X}_\alpha,\text{Prot}})$: executes the protocol Prot between the entities $(\mathcal{X}_1, \ldots, \mathcal{X}_\alpha)$, fills up and outputs their transcripts $(\pi^{i_1}_{\mathcal{X}_1,\text{Prot}}, \ldots, \pi^{i_\alpha}_{\mathcal{X}_\alpha,\text{Prot}})$.

This oracle limits the analysis to eavesdroppers, while the two following ones allow the fragmentation of the previous oracle, and can be used to represent active adversaries.

- $\mathcal{O}^{\text{Launch}}(\text{Prot}, \mathcal{X}) \rightarrow \pi^i_{\mathcal{X},\text{Prot}}$: makes $\mathcal{X}$ launch a new execution of the protocol Prot, fills the transcript $\pi^i_{\mathcal{X},\text{Prot}}$ of the i^{th} execution of $\mathcal{X}$'s corresponding algorithm, and outputs $\pi^i_{\mathcal{X},\text{Prot}}$ when all the actions related to the oracle query are performed.
- $\mathcal{O}^{\text{Send}}(\text{Prot}, \mathcal{X}, m) \rightarrow \pi^i_{\mathcal{X},\text{Prot}}$: sends a message m to $\mathcal{X}$, fills the transcript $\pi^i_{\mathcal{X},\text{Prot}}$ of the i^{th} execution of $\mathcal{X}$'s corresponding algorithm, and outputs $\pi^i_{\mathcal{X},\text{Prot}}$ when all the actions related to the oracle query are performed.

Remark. An entity may directly start a protocol execution upon reception of an $\mathcal{O}^{\text{Send}}$ query.

(iii) Oracle that captures information about the system $\mathcal{S}$

- $\mathcal{O}^{\text{Corrupt}}(\mathcal{X}) \rightarrow \varepsilon^i_{\mathcal{X}}$: outputs the i^{th} snapshot $\varepsilon^i_{\mathcal{X}}$ of $\mathcal{X}$'s internal state.

7.3.2 Selectors

Using the oracles, the adversary $\mathcal{A}$ can obtain transcripts and snapshots. Both are useless, though, without any means to read them. This section introduces the concept of *selectors*, which provide $\mathcal{A}$ with the ability to read the information they contain.

Given a selector s, $\pi.\mathsf{s}$ (resp. $\varepsilon.\mathsf{s}$) denotes the information accessible through s contained in the transcript π (resp. the snapshot ε).

The formalization with selectors allows an extreme extendability of the model. With the technology and environment progress, it further eases the definition of new suitable selectors. For illustrative purposes, the list of selectors that at least allow the definition of the already existing adversaries given in the literature related to RFID privacy models is provided below.

(i) Selectors related to transcripts

- msg: ability to extract from transcripts the messages sent over the channels reader-tag.
- result: ability to extract from transcripts the result of the protocol (**success** or **failure**).

Example 7.10. Let $\mathcal{A}$ being an adversary $\mathcal{A}$ who queries the $\mathcal{O}^{\text{Execute}}$ oracle with a reader $\mathcal{R}$ and a tag $\mathcal{T}$, where Prot is SK-Prot (presented in Section 4.1.1). If $\mathcal{A}$ has the selector msg, then she gets $\pi^1_{\mathcal{R},\text{SK-Prot}}.\mathsf{msg} = (n^1_{\mathcal{R}}, n^1_{\mathcal{T}} || F_{\mathsf{k}_{\mathcal{T}}}(n^1_{\mathcal{R}}, n^1_{\mathcal{T}}))$ which are the messages exchanged during the first protocol execution. If $\mathcal{A}$ has the selector result, then she obtains $\pi^1_{\mathcal{R},\text{SK-Prot}}.\mathsf{result}$, which is **success** if $\mathcal{T}$ is registered.

Other selectors could be defined, such as $\mathsf{msg}_{\mathcal{R}}$ and $\mathsf{msg}_{\mathcal{T}}$ to represent the ability to obtain the messages sent over the forward (reader to tag) and backward (tag to reader) channels. These two specific selectors refine the analysis by considering the asymmetry of the reader-tag channels, given that the forward one can be eavesdropped at a longer distance in practice [80, 160] as already explained in Sections 3.2 and 4.2.3.

(ii) Selectors related to snapshots

- eeprom: ability to extract from tag snapshots the non-volatile memory.
- ram: ability to extract from tag snapshots the volatile memory.

The distinction between the non-volatile and the volatile memory of a tag internal state has been first considered in [139], then developed in [6]. All the existing models presented in Chapter 3 only offer one choice upon tag corruption: either the adversary can only obtain the non-volatile memory, or she can obtain both. The model is not limited to such restrictive configurations given that selectors are controlled independently.

7.3.3 Formalizing New Adversaries

The model can be easily extended in order to take into account news security threats like time attacks or the potential corruption of readers, as described below.

Time Attacks. Chapter 5 proposed a new kind of adversary who is able to deduce the time spent by a reader to authenticate a given tag during a protocol execution. To model such an attack, it is possible to define a selector timer that gives the adversary the ability to extract from transcripts the execution time of the reader.

Compromised Readers. Chapter 6 presented an adversary who is allowed to corrupt readers and obtain all their data. To model such a reader corruption, it is thus required to define two selectors $\mathsf{eeprom}_{\mathcal{R}}$ and $\mathsf{ram}_{\mathcal{R}}$ that give the adversary the ability to extract from reader snapshots the non-volatile and volatile memory.

7.4 Untraceability

This section presents the *untraceability experiment* that formalizes the attack performed by the adversary on an RFID system. It also describes

the *adversary classes*, i.e., the actions and capabilities given to the adversary during her attack. Finally, this section defines the two levels of resistance to this attack which are modulated by the adversary classes.

7.4.1 Untraceability Experiment

In the untraceability (UNT) experiment, the adversary can interact with the system $\mathcal{S}$ within some limitations, which define the adversary classes (introduced in Section 7.4.3). At some point during the experiment, the adversary selects two tags $\mathcal{T}_i$ and $\mathcal{T}_j$, called challenge tags[2]. The challenger $\mathcal{C}$ then executes the ChallProc procedure (defined below) which randomly reassigns the labels $\mathcal{T}_i$ and $\mathcal{T}_j$, and respectively performs c_1 and c_2 (parameters of the experiment) protocol executions[3] on these tags. This formalizes their potential evolution out of the adversary control. The adversary goal is to find the match between the former and the latter labels of the challenge tags.

Note that step 1 of the ChallProc procedure has been set up in order to avoid the trivial attack explained in [44][4]. For clarity reasons, once the challenge tags have been reassigned, their labels are denoted $\widetilde{\mathcal{T}}_i$ and $\widetilde{\mathcal{T}}_j$. The values c_1 and c_2 may potentially be equal to 0, and characterize the untraceability properties as discussed in Section 7.4.4. The untraceability experiment $Exp^{\mathsf{UNT}}_{\mathcal{S},\mathcal{A}_P}$ is formally defined in Figure 7.1, where the adversary $\mathcal{A}$ belonging to the class P is denoted $\mathcal{A}_P$, and b is a random bit. The ChallProc procedure is formally defined in Figure 7.2, where the values b, c_1, c_2, and the challenge tags $\mathcal{T}_i$ and $\mathcal{T}_j$ are its parameters.

[2]This implies that the adversary must create at least two tags to perform the untraceability experiment.

[3]If the system to analyze contains several protocols, then one pair (c_1, c_2) per protocol must be given as parameters of the experiment and of the ChallProc procedure. Since most RFID systems consist of a single protocol, we only present this case all along this section for the sake of clarity.

[4]In a nutshell, if $\mathcal{C}$ only relabels two challenge tags that are not in the same step of a protocol execution, then the adversary may trivially differentiate them from their behaviors.

Experiment $Exp^{\mathsf{UNT}}_{\mathcal{S},\mathcal{A}_P}(\lambda, b, c_1, c_2)$

1. $\mathcal{C}$ runs InitSystem(1^λ).
2. $\mathcal{A}_P$ interacts with the system $\mathcal{S}$.
3. $\mathcal{A}_P$ selects two challenge tags $\mathcal{T}_i$ and $\mathcal{T}_j$, and $\mathcal{C}$ runs ChallProc$(b, \mathcal{T}_i, \mathcal{T}_j, c_1, c_2)$.
4. $\mathcal{A}_P$ interacts with the system $\mathcal{S}$.
5. $\mathcal{A}_P$ outputs a guess bit b'.

Output: Return 1 if $b = b'$ and 0 otherwise.

Figure 7.1: Untraceability experiment of the model.

ChallProc$(b, \mathcal{T}_i, \mathcal{T}_j, c_1, c_2)$

1. $\mathcal{C}$ closes the algorithm executions still run by the challenge tags, if any.
2. $\mathcal{C}$ relabels the two challenge tags as follows:
   - if $b = 0$, then the labels remain unchanged,
   - if $b = 1$, then the labels are swapped.
3. $\mathcal{C}$ performs c_1 (resp. c_2) $\mathcal{O}^{\text{Execute}}$ queries on $\widetilde{\mathcal{T}_i}$ (resp. $\widetilde{\mathcal{T}_j}$) with the entity(ies) that maximize $\mathcal{A}_P$'s success probability, and provides their transcripts to $\mathcal{A}_P$.

Figure 7.2: ChallProc procedure of the model.

7.4.2 Restrictions of the Experiment

During the untraceability experiment, the use of the oracles can be limited by means of *restrictions*. Some useful ones are defined below.

The first restriction controls the use of $\mathcal{O}^{\text{RegisterTag}}$.

- `Reg`: $\mathcal{T}_i$ and $\mathcal{T}_j$ must be registered.

This restriction is particularly useful to discard the attack that consists in distinguishing non-registered tags from registered ones. However, `Reg` may not always be desired, for instance when a system must be resistant to such an attack, as highlighted in the following example.

Example 7.11. In the case of public transportation, a new pass without any contract (i.e., not yet registered) should be indistinguishable by an adversary from a registered one.

Then, if $\mathcal{O}^{\text{Corrupt}}$ can be queried, the following restrictions can be established to graduate its use.

- `Forward`: $\mathcal{T}_i$ and $\mathcal{T}_j$ cannot be corrupted.
- `Backward`: $\widetilde{\mathcal{T}_i}$ and $\widetilde{\mathcal{T}_j}$ cannot be corrupted.
- `Side`: both `Forward` and `Backward` restrictions must be respected.

7.4.3 Adversary Classes

The explicit characterization of the adversary performing the untraceability experiment is achievable through the concept of *adversary class*. The following definition ensures a fine granularity on the adversary classes.

Definition 7.12 (Adversary Class). *An* adversary class *P is defined by three sets O, S, R, and is denoted by the 3-tuple (O, S, R), where:*

- *O is the set of available oracles,*
- *S is the set of available selectors,*
- *R is the set of restrictions regarding the use of the available oracles during the experiment.*

There exists a partial order relation on the set of adversary classes.

Definition 7.13 (Stronger Class). *Let X be a set of restrictions and Y be a set of oracles, then $X|_Y$ denotes the subset of restrictions belonging to X that are related to the oracles in Y.*
A class $P = (O, S, R)$ is said to be stronger *than another class $P' = (O', S', R')$, denoted $P \geq P'$, if and only if (i) $O' \subseteq O$, (ii) $S' \subseteq S$, and (iii) $R|_{O \cap O'} \subseteq R'|_{O \cap O'}$.*

In order to lighten the notations, several typical adversary classes are presented. The goal is not to provide an exhaustive list, but to highlight the most intuitive ones. Let first denote

$$\mathcal{O}_{\text{basic}} = \{\mathcal{O}^{\text{CreateTag}}, \mathcal{O}^{\text{RegisterTag}}, \mathcal{O}^{\text{Execute}}, \mathcal{O}^{\text{Launch}}, \mathcal{O}^{\text{Send}}\}$$

the set of basic oracles. The subsequent typical class can be straightforwardly defined.

- $\mathsf{CLASSIC} = (\mathcal{O}_{\text{basic}}, \{\mathsf{msg}, \mathsf{result}\}, \{\mathtt{Reg}\})$.

This class can be extended by adding the oracle $\mathcal{O}^{\text{Corrupt}}$ and the selector eeprom as follows.

- $\mathsf{STRONG} = (\mathcal{O}_{\text{basic}} \cup \{\mathcal{O}^{\text{Corrupt}}\}, \{\mathsf{msg}, \mathsf{result}, \mathsf{eeprom}\}, \{\mathtt{Reg}\})$.

If the restrictions `Forward`, `Backward`, or `Side` are added to the STRONG class, then the respective FORWARD, BACKWARD, and SIDE classes are obtained. According to Definition 7.13, they also verify:

The *narrow* adversaries[5] can also be formalized in the model, when the result selector is not available. The resulting classes are then denoted with the prefix NARROW, e.g., NARROW-STRONG.

[5] A narrow adversary does not know whether the protocol execution succeeded. The concept has been highlighted by Juels and Weis in [99], and Vaudenay introduced the terminology in [162]. See Chapter 3 for more details.

7.4.4 Universal and Existential Untraceabilities

The granularity of the model is even more refined with the concepts of Universal and Existential untraceabilities. The Universal untraceability represents the case where no adversary $\mathcal{A}_P$ can win the untraceability experiment, whatever the experiment parameters b, c_1, and c_2 are.

Definition 7.14 (P-Universal Untraceability). *An RFID system $\mathcal{S}$ is said P-*Universal untraceable*, denoted P-*Universal-UNT*, if:*

$$\forall (c_1, c_2) \in \mathbb{N}^2, \forall \mathcal{A}_P \in P, \forall b \in \{0,1\}: \left| \Pr\Big(Exp^{\mathsf{UNT}}_{\mathcal{S},\mathcal{A}_P}(\lambda, b, c_1, c_2) \rightarrow 1 \Big) - \frac{1}{2} \right| \leq \epsilon(\lambda).$$

Considering the case $(c_1, c_2) = (0, 0)$ is enough to prove the P-Universal-UNT property when the adversary class P does not contain any restriction to limit the number of $\mathcal{O}^{\text{Execute}}$ queries or to limit the number of both $\mathcal{O}^{\text{Launch}}$ and $\mathcal{O}^{\text{Send}}$ queries. Indeed, if such an adversary is not able to win when $(c_1, c_2) = (0, 0)$, she is neither able to win when $(c_1, c_2) \neq (0, 0)$, since she is free to perform complete protocol executions herself and can thus simulate $\mathcal{C}$'s queries to $\mathcal{O}^{\text{Execute}}$ in the first case.

On the other side, the Existential untraceability represents the case where no adversary $\mathcal{A}_P$ can win the untraceability experiment for some given parameters.

Definition 7.15 (P-Existential Untraceability). *An RFID system $\mathcal{S}$ is said P-*Existential untraceable*, denoted P-*Existential-UNT*, if:*

$$\exists (c_1, c_2) \in \mathbb{N}^2, \forall \mathcal{A}_P \in P, \forall b \in \{0,1\}: \left| \Pr\Big(Exp^{\mathsf{UNT}}_{\mathcal{S},\mathcal{A}_P}(\lambda, b, c_1, c_2) \rightarrow 1 \Big) - \frac{1}{2} \right| \leq \epsilon(\lambda).$$

Definitions 7.14 and 7.15 yield the following properties, where $A \Rightarrow B$ means that: if an RFID system $\mathcal{S}$ satisfies the untraceability property A, then $\mathcal{S}$ also satisfies the untraceability property B.

Property 7.16. *Given an adversary class P:*

$$P\text{-Universal-UNT} \Rightarrow P\text{-Existential-UNT}.$$

Property 7.17. *Given two adversary classes P and P', if $P \geq P'$ then:*

$$P\text{-}\mathsf{Universal\text{-}UNT} \Rightarrow P'\text{-}\mathsf{Universal\text{-}UNT},$$

and

$$P\text{-}\mathsf{Existential\text{-}UNT} \Rightarrow P'\text{-}\mathsf{Existential\text{-}UNT}.$$

Several papers [50, 55, 109] stress that realistic adversaries are not always able to permanently control the interactions of the tags. Consequently, they introduce protocols that seem to be resistant to these realistic adversaries. The model is able to analyze such protocols and provide them with Existential untraceability proofs. In particular, some protocols may reach a high level of untraceability when complete protocol executions are performed out of the adversary control, while other models consider that these protocols are weak. An example of such a protocol is given with the analysis of the O-FRAP protocol in Section 7.5.3.

7.5 Untraceability Analyses

The model allows a precise evaluation of the untraceability level of protocols. In order to highlight this fact, but also to illustrate the differences between Universal and Existential untraceability properties, the SK-Prot, MW-Prot, O-FRAP, and PK-Prot protocols presented in Chapter 4 are analyzed within the model. The studies are performed with the typical adversary classes defined in Section 7.4.3 in an SE architecture (see Section 7.2.1 for more details).

7.5.1 Analysis of SK-Prot

First of all, this protocol is neither FORWARD-Existential-UNT nor BACKWARD-Existential-UNT in an SE architecture. This is because one single corruption of a tag $\mathcal{T}$ allows an adversary to trace $\mathcal{T}$ unconditionally (i.e., for any pair (c_1, c_2)), as explained in Section 4.1.1. Yet, the protocol reaches a reasonable untraceability level in Universal scenarios.

Theorem 7.18. *SK-Prot is SIDE-Universal-UNT in an SE architecture.*

Proof. An adversary $\mathcal{A}_{\mathsf{SIDE}}$ is not allowed to tamper with the challenge tags, but she can tamper with all the other tags of the system. Since

all the tags secrets are independent and fixed, the knowledge of the non-challenge tags secrets does not help $\mathcal{A}_{\mathsf{SIDE}}$ in tracing the challenge tags.

Consequently, the only set of information that may help $\mathcal{A}_{\mathsf{SIDE}}$ for her attack is the messages exchanged by the challenge tags during their protocol executions. Obviously, nothing can be learned from the nonces outputted by the reader. Challenge tags respond with a freshly generated nonce and the output of a PRF, containing the appropriate key and the nonces. Even if an adversary can control the nonce sent to a tag, answers of tags are always different (and unpredictable) as the adversary cannot control the random value inserted by the tag during the computation of the PRF. In conclusion, if an adversary is able to distinguish the challenge tags, she is able to break the PRF assumption of the function.□

7.5.2 Analysis of MW-Prot

This protocol is not SIDE-Existential-UNT in an SE architecture. This is due to its key-management, as pointed out in the attack given in Section 4.1.2 which can be converted here as follows. Assume an adversary $\mathcal{A}_{\mathsf{SIDE}}$ that collects a complete path of keys after a tag corruption. Let consider the first level of the tree. Informally, tags can be split in two sets, those that belong to the same subtree of the corrupted tag and the others. Thanks to the obtained key, $\mathcal{A}_{\mathsf{SIDE}}$ can decide if a tag belongs to the same subtree or not. Consequently, if $\mathcal{A}_{\mathsf{SIDE}}$ selects the two challenge tags, one from each set, then she can easily distinguish them breaking the SIDE untraceability property. This remain true for any pair (c_1, c_2) as keys are never updated during the life of the system. However, the protocol ensures the weakest untraceability level for Universal scenarios.

Theorem 7.19. *MW-Prot is CLASSIC-Universal-UNT in an SE architecture.*

Proof. This proof is very similar to the one of Theorem 7.18. The only difference is that a CLASSIC adversary is not allowed to tamper with any tag of the system. Therefore, the only set of information that she can collect for her attack is the messages exchanged during protocol executions. With the same reasoning as the proof of Theorem 7.18, the same conclusion can be made: if an adversary is able to distinguish two tags, she is able to break the PRF assumption of the function. □

7.5.3 Analysis of O-FRAP

As already highlighted in Section 4.1.4, O-FRAP is not FORWARD-Universal-UNT in an SE architecture with the following attack. The adversary $\mathcal{A}_{\mathsf{FORWARD}}$ starts a protocol execution with a random tag, denoted $\mathcal{T}_i$, using the query $\mathcal{O}^{\text{Send}}(\text{O-FRAP}, \mathcal{T}_i, n_{\mathcal{R}})$, and receives $\pi^1_{\mathcal{T}_i,\text{O-FRAP}}$. Then, she does not pursue the protocol execution. She chooses this tag and another random one $\mathcal{T}_j$ for the ChallProc procedure. As Universal-UNT is considered without any restriction regarding the number of oracles queries, $(c_1, c_2) = (0, 0)$ and $\mathcal{C}$ only relabels these tags $\widetilde{\mathcal{T}}_i$ and $\widetilde{\mathcal{T}}_j$. Then, $\mathcal{A}_{\mathsf{FORWARD}}$ queries $\mathcal{O}^{\text{Corrupt}}(\widetilde{\mathcal{T}}_i)$ and uses the obtained secret key from $\varepsilon^1_{\widetilde{\mathcal{T}}_i}$.eeprom to recompute the potential answer of this tag in the first protocol execution. If this message is equal to the one contained in $\pi^1_{\mathcal{T}_i,\text{O-FRAP}}$.msg, it means that $\mathcal{T}_i = \widetilde{\mathcal{T}}_i$, otherwise $\mathcal{T}_i = \widetilde{\mathcal{T}}_j$. $\mathcal{A}_{\mathsf{FORWARD}}$ always succeeds in performing this attack as the secret key she obtained has not been updated[6].

O-FRAP is neither BACKWARD-Existential-UNT in an SE architecture. Firstly, the adversary $\mathcal{A}_{\mathsf{BACKWARD}}$ can obtain the secret keys of the challenge tags before the ChallProc procedure with the eeprom selector. Secondly, all the values used for the tags key update mechanism are sent in clear on the reader-tag channels during the protocol executions: $\mathcal{A}_{\mathsf{BACKWARD}}$ can obtain these values with the msg selector. Therefore, $\mathcal{A}_{\mathsf{BACKWARD}}$ can recompute the key updates of the challenge tags, and distinguish them after the ChallProc procedure. This protocol nevertheless reaches interesting untraceability levels.

Theorem 7.20. *O-FRAP is SIDE-Universal-UNT and FORWARD-Existential-UNT in an SE architecture.*

Proof. When an adversary $\mathcal{A}_{\mathsf{SIDE}}$ is playing the Universal experiment, the challenge tags cannot be corrupted, thus the previous attacks are not possible. Since tag answers are the result of a PRF, $\mathcal{A}_{\mathsf{SIDE}}$ cannot learn any information about the answers of the challenge tags, even knowing some "valid" couples (input, output) from non-challenge tags that have been corrupted.

Considering an Existential experiment where $(c_1, c_2) = (1, 1)$, an adversary $\mathcal{A}_{\mathsf{FORWARD}}$ cannot perform the attack given above. Indeed, the

[6] A tag key update only arises when an O-FRAP execution is complete.

challenge tags perform a complete protocol execution which necessarily updates their secret key during the ChallProc procedure. From the PRF assumption, $\mathcal{A}_{\mathsf{FORWARD}}$ is unable to retrieve the previous secret key of a given tag. Consequently, $\mathcal{A}_{\mathsf{FORWARD}}$ will not find a match with transcripts obtained before the ChallProc procedure, and thus cannot retrieve the correct bit otherwise than at random. □

7.5.4 Analysis of PK-Prot

Finally, PK-Prot is not STRONG-Existential-UNT in an SE architecture even if the public-key cryptosystem used is IND-CCA2 secure. Actually, this protocol does not process any key update mechanism of the tag secret keys. Since a STRONG adversary can corrupt the challenge tags before and after the ChallProc procedure, then she can distinguish them with their secret keys obtained with the eeprom selector. This remain true for any pair (c_1, c_2). Yet, this protocol reaches quite high untraceability levels.

Theorem 7.21. *If the public-key cryptosystem is IND-CCA2 secure, then PK-Prot is BACKWARD-Universal-UNT and FORWARD-Universal-UNT in an SE architecture.*

Proof. We only provide the sketch of proof of the theorem, since the complete proof is largely inspired by the one carried out for the same protocol in [34]. It is based on the game technique of Shoup (another very similar proof is given in Section 6.4.5).

Let $\mathcal{A}$ denote the adversary, either BACKWARD or FORWARD, of the Universal experiment. The initial game describes the normal behavior of all the oracles and how transcripts and snapshots have to be generated. The aim of this proof is to reach, through intermediate games, the final game where all the answers of the challenge tags are replaced by encryptions of nonces. It is obvious that $\mathcal{A}$'s success probability in this last game is $\frac{1}{2}$, as no information can leak from such messages. If all the differences of success probabilities of two successive games are negligible, then it is possible to conclude that $\mathcal{A}$'s success probability in the initial game (i.e., in the experiment) is equal to $\frac{1}{2}$ plus a negligible factor. Each transition game replaces one more "normal" encryption of the challenge tags by an encryption of a nonce. If $\mathcal{A}$'s success probabilities

in two successive games are distinguishable, then it is possible to design an IND-CCA2 distinguisher $\mathcal{D}$ that takes advantage of these two games to win the IND-CCA2 experiment.

Before concluding, several remarks are highlighted to clarify some detail of the proof. The corruption of challenge tags (either before or after the ChallProc procedure) does not modify this proof since knowing the plaintext is useless to recognize the potential ciphertext (under the IND-CCA2 assumption). Regarding $\mathcal{D}$, it is not possible to consider that $\mathcal{D}$ knows the decryption key. To correctly generate the result of a protocol execution, two cases are possible. If the ciphertext has been produced by $\mathcal{D}$ in response to an $\mathcal{O}^{\text{Send}}$ query on any tag, then $\mathcal{D}$ can trivially accept (or reject) the ciphertext as $\mathcal{D}$ knows which tag should have produced it. If the ciphertext has been created by $\mathcal{A}$, then the ciphertext cannot be the one that corresponds to the challenge ciphertext of the IND-CCA2 experiment. $\mathcal{D}$ can consequently request its decryption oracle, and then decide if this answer should be accepted or rejected.

As a conclusion, the success probability of an adversary, either BACKWARD or FORWARD, in the Universal experiment is equal to $\frac{1}{2}$ plus a negligible factor. □

7.6 Impact of the Model

The research community in cryptography pays particular attention to RFID given the privacy concerns of customers and the pressure of the authorities to advance the field. The model introduced in this chapter consequently focuses on RFID, but its impact goes beyond this domain, and a design requirement was to make it easily applicable to other pervasive technologies. It has been further fashioned to study any kind of system, not exclusively identification ones.

The proposed model uses an approach to RFID privacy that is inspired from others. Its design aims at helping the community to compare protocols meaningfully by providing the following properties.

- *Extendability.* RFID is an evolving technology, and the related attacks as well. Considering that architectures, adversaries, etc. may change in the future is thus a conservative assumption. The model considers that such changes will arise. In particular, the model

introduces new considerations in the formalization of system architectures which are more realistic than what was considered until now. A portfolio of adversary classes has been provided but additional classes can be defined to fit new attacks and deal with new scenarios without modifying the foundations of the model.

- *Granularity.* Heavy cryptographic building blocks with security proofs are unlikely to meet the constrained resource requirements of ubiquitous environments. Instead, degraded primitives without security proofs are preferred in practice. The granularity of the model through the Universal and Existential notions provides one step to mitigate this problem in RFID. It allows protocols to benefit from proofs with a precise and identified untraceability level, not necessarily the highest one. In particular, the model enables the analysis of protocols when the adversary does not control all the interactions of the entities.

Finally, the definition of the adversary actions (oracles), capabilities (selectors and restrictions), and goals (experiment) are inspired from real life, which makes the model intuitively usable.

Chapter 8

Toward Privacy Certification

So far, this thesis only focused on the privacy problems and solutions related to RFID systems. Yet, the recent technological advances in computer sciences have brought the question of privacy in broader IT environments at the forefront of citizens' concerns. Nowadays, when customers subscribe to an IT service or solution, they yearn for guarantees that their privacy will not be threaten. Despite this general worry, no current standardized certification is available today to practically assess the privacy level of IT environments. This chapter aims to fill this gap by addressing privacy in the less restrictive domain of ubiquitous computing systems based on any pervasive technology, including RFID.

In this chapter, we first introduce the context in which we foresee the development of a privacy certification. We then display the privacy landscape in Europe from the legal, societal and non-academic information security point of views. In particular, we present (i) the most important European legislations related to the protection of citizens' privacy and (ii) the existing approaches that have been developed to provide and evaluate privacy in IT environments. Finally, we lay the foundation stones on a general methodology that will help IT stakeholders in designing together an international privacy certification able to assess the privacy level of ubiquitous computing systems.

8.1 Context of Privacy Certification

This section introduces the IT systems and entities that are targeted by an expected privacy certification.

8.1.1 Ubiquitous Computing (UbiComp) Systems

Since the 1980s, the technological hardware and software advances have contributed to the evolution and modification of the networked systems topology. Nowadays, these systems are no longer limited to interconnected computers. As illustrated in Figure 8.1, they also include many pervasive devices either embedded into everyday objects that are carried by persons (e.g., smartphones, GPS navigation devices, tablets, pacemakers) or integrated into objects that support persons in their daily activities (e.g., monitoring household appliances or plants, tracking books or bicycles).

As all these pervasive devices execute tasks on behalf of their holder, they gather a lot of personal data. The (potentially) excessive collection, manipulation, flow, etc. of personal data in such system may thus endanger their customers privacy as described in the following section. Consequently, UbiComp systems should be subject to privacy controls that evaluate the protections put in place to preserve privacy.

8.1.2 Privacy of Customers

The massive widespread of UbiComp systems has brought up major concerns about the protection of customers privacy, either from the research or from the authorities points of view (see Section 8.2 for more details). Privacy is also a concept strongly fixed in the public mind, as pointed out by Spiekermann in [154]:

> "*A global survey found that 88% of people are worried about who has access to their data; over 80% expect governments to regulate privacy and impose penalties on companies that do not use data responsibly.*"

Despite this general agreement, customers seem to be disposed to disclose their personal data in specific circumstances, e.g., to gain additional services. For instance, people using loyalty cards do not mind to reveal their consumption preferences to get some discount vouchers.

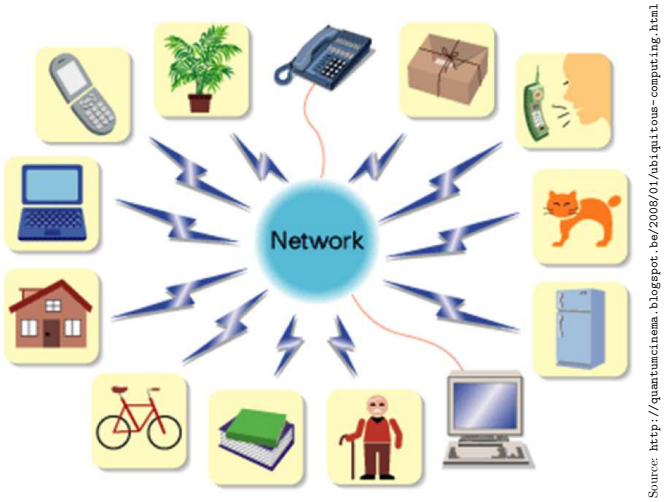


Figure 8.1: Everyday life UbiComp systems.

To illustrate this contradiction, Kumpost and Matyás presented two experimental studies in [105] displaying how people value their personal data. In both experiments, the authors gave a questionnaire to a sample of people using a cover story to hide the real intention of the study. The questions were related to the financial reward expected by each person to disclose their personal data for different data usage (e.g., collection of their mobiles phone data every five minutes during one month for academic purposes). The goal was to identify the monetary value that people would attach to such a voluntary disclose. The main conclusion of both studies is that the answered value is very modest in average and clearly depends on the data collection usage (higher value for commercial purposes than for academic purposes).

Customers neither seem to be clearly aware of the consequences that may be caused by privacy breaches, or underestimate the risks. In fact, they do not necessarily picture the (possibly malicious) entities, either system operator[1] or external adversary, who could ill-advisedly use their personal data, as illustrated in the following examples.

[1]In this chapter, "operator" designates a company that supplies an IT service or solution.

- In 2012, a document containing the private contact details of more than 1.4 million clients of the national Belgian railway company (SNCB) was available on the SNCB website during one day because of an involuntary human mistake. Taking advantage of this breach, a SNCB user created a website[2] that listed the name of all the SNCB customers affected by the disclose.

- In 2006, AOL intentionally released the detailed search logs (around 20 million) of nearly 658,000 users on one of their websites. Even if the disclosure was intended for research purposes, AOL has been sued for the "violation of the federal Electronic Communications Privacy Act [...and for] the tort of public disclosure of private facts" [131].

- In 2011, the Sony PlayStation Network (PSN) has been the victim of an external hacking attack. The personal data of more than 70 million online gamers (in particular their name, address, birthdate, PSN login and password) have been stolen.

To mitigate their lack of awareness, customers should be educated to the risks and impacts of the problems related to privacy. An official privacy certification that assesses the privacy level of UbiComp systems would definitely help customers in developing an automatic reflex to preserve their privacy.

8.2 Current Privacy Landscape

During the last decade, the concept of privacy has been intensively addressed by the legal, societal, and non-academic information security communities. As a result, many regulations, initiatives and methodologies have been proposed to contribute to the effort of defining means for providing and assessing the privacy of IT systems. This section presents the different approaches and existing solutions that can be found and applied in the European Union (EU).

[2]Which is no longer available.

8.2.1 European Legal Obligations on Privacy

The concept of privacy in the legal framework is quite recent, since it has only been explicitly put into words in the Universal Declaration of Human Rights of 1948:

> "*No one shall be subjected to arbitrary interference with his privacy, family, home or correspondence, nor to attacks upon his honour and reputation. Everyone has the right to the protection of the law against such interference or attacks.*" (Article 12).

In Europe, its first appearance is in Article 8 of the European Convention on Human Rights of 1953 which states a right to respect everyone's "private and family life, his home and his correspondence". Since then, the EU sets up a rich legislation to protect citizens' privacy. The most significant and influential acts are presented below.

Data Protection Directive 95/46/EC [62]

This directive of 1995 is the first European one related to privacy, and stands for the reference text on the protection of individuals' personal data. It targets the processing and the free movement of personal data within the EU only.

Key Notions of the Directive. This act defines the two following key notions that are used in all the subsequent European legislative acts.

"Personal data" denotes any information related to a physical person identified or identifiable, particularly by an identification number or any specific element proper to his identity (e.g., physical, physiological, economical, cultural, social).

"Processing of personal data" denotes any (set of) operation(s) performed on personal data with or without automatized procedures (e.g., collection, record, conservation, consulting, distribution).

Main Principles. The directive establishes several rules regarding the process of personal data. In particular, any personal data must be:

- processed faithfully and lawfully;
- collected for specified and explicit purposes;
- not processed subsequently, except for historical, statistical or scientific purposes conducted with suitable safeguards (e.g., anonymizing the data prior processing);
- correct and up-to-date, otherwise should be erased or rectified;
- not excessive regarding the purposes for which they are collected;
- kept during an appropriate duration time limit, i.e., no longer than the time necessary to the legitimate purpose achievements.

The act additionally specifies the criteria of legitimate data processing, the information that should be provided to the data owner, as well as the owner's rights to access his data or to object their processing (e.g., for direct marketing).

Furthermore, confidentiality and security of data processing are required. Data processing should only be performed by an authorized person in order to achieve the specified legitimate purposes. Data collection and processing should also be subject to technical protecting mechanisms, especially when data processing implies data transmission over a network. Appropriate security levels should be put in place according to the potential risks of data processing and of the data nature itself (e.g., for sensitive personal data such as bank account credentials).

Finally, the transfer of personal data to a third country (i.e., not part of the EU) may only be possible if the receiving third country keeps ensuring an adequate level of protection to the transfered personal data.

Article 29 Working Party. This working group is an independent advisory board that has been set up under Directive 95/46/EC. Its organization and missions are defined in Articles 29 and 30 of the directive.

In particular, the working group: (i) advises the European Commission on any proposed modification of the directive or any new potential safeguard measure improving data protection, (ii) promotes a uniform application of the directive within the EU via the existing partnerships

between the supervisory authorities of each Member State, and (iii) provides recommendations and expert guidelines on the protection of personal data for the general public.

E-Privacy Directive 2002/58/EC [63]

This directive is an extension of the Data Protection Directive to regulate the processing of personal data and the protection of users' privacy in the electronic communication sector.

Key Notions of the Directive. For the sake of clarity, this act specifies the three following terminologies.

"Communication" denotes any information provided by a user to send and/or receive a message and/or data (e.g., name, address).

"Traffic data" denotes any information related to the routing of a communication within a network (e.g., data to route, duration, time or volume routing).

"Location data" denotes any information related to the geographic position of an user's electronic equipment (e.g., geographic coordinates, identification of the equipment network cell at a given time).

Main Principles. Regarding the security of electronic communications, this act refers to the security principles of the Data Protection Directive, and adapts them to electronic networks. It further states that the provider of an electronic communication service is mandatory to inform users of the potential security breaches with a comprehensive description of their risks and repercussions.

Confidentiality of communications and related traffic data are also required. In particular, no user should be able to eavesdrop, intercept or store communications and traffic data of another user. Recording is however possible for lawful professional use or sale transaction proof.

Traffic data that are processed and stored by a provider must be anonymized or erased when they are no longer necessary for the communication purpose, except for billing or interconnection payments. Traffic data should however not be processed for marketing purposes, except (i) when this action is mandatory to provide the service or, (ii) when

the user has given his consent[3]. Furthermore, users must be informed of the potential processing of their data and its duration time prior to their consent. As for Directive 95/46/EC, data processing should only be performed by an authorized person of the provider company for the service purposes. Several exceptions are nevertheless specified (e.g., detention on suspicion, charges of penal violation, national security).

The article referring to location data is mainly intended for mobile communication services when the user's position should be established before any service provision. Location data may be processed for the duration necessary to provide the service either after anonymizing the data or after the user's consent. As for traffic data, users must be informed of the potential processing of their location data, its duration time, and whether their data will be transmitted to a third party (only possible in case of value added service) prior to their consent.

Finally, users should be informed when their data are included in a printed and/or electronic directory. They further should be able to check, correct, or erase their data from the directory. Spamming for marketing purposes without prior user's consent should be forbidden.

Data Retention Directive 2006/24/EC [64]

This directive is an amendment of the E-Privacy Directive. It specifies the obligations of providers on the conditions and durations of telecommunication data retention. Theses obligations should be applied to users' traffic and location data, and to data allowing users' identification, but should not be applied to the communication contents.

Main Principles. Firstly, this act requires providers to retain some specific data as stated in the Data Protection and the E-Privacy directives. These data can be the user's name, address, IP address, date, communication start and end time (this list is not exhaustive).

The access to such data should only be provided to the competent national authorities in some specific cases (mainly for legal actions or national security). Their retention duration should not be less than six months and should not exceed the maximum of two years. Data should be destroyed at the end of the retention duration. The exceptions

[3]Note that the user should be allowed to withdraw his consent at any time.

to this statement are data that have been legitimately accessed during the retention duration: the directive considers that such data may be sensitive or conflictual and may thus be useful for future lawful accesses.

The retained data should be protected with the same security level as traffic and location data. Finally, they should also be subject to technical protecting mechanisms against incorrect or illegal operations (e.g., accidental destruction or loss, unauthorized access or disclosure).

8.2.2 Existing Ways to Provide Privacy

This section presents several initiatives and guidelines for building privacy-friendly IT systems. They consist of the codes of good practice and the technical, yet non-cryptographic, existing means.

Privacy-by-Design Initiative

The legal context presented so far clearly showed that privacy in IT environments is a key concept that should always be considered to respect individual freedoms. Nevertheless, it is not clear whether these directives have a practical impact on deployed systems.

To that purpose, the Commissioner for Information and Privacy of Ontario, Ann Cavoukian, strives to promote the concept of "privacy-by-design" (PbD) [39], i.e., the protection of personal data should be up and running in every large-scale deployed IT system. PbD has been further extended to accountable business practices and to physical design of networked infrastructures. It should apply to any personal data, but especially to sensitive ones (e.g., health and financial data).

The Seven Fundamental Principles. They have been settled by Cavoukian to achieve PbD in any IT system.

***Proactive and preventative*:** IT providers should be proactive rather than reactive, by anticipating the potential privacy risks. The privacy measures should be put in place before the deployment of the system as preventions rather than after as remedies.

***Default setting*:** privacy should be automatically provided by any system. The user should not be mandatory to add security mechanisms to the system in order to protect his privacy.

***Embedded*:** privacy should be included in the design and architecture of any system. It should not be an add-on but rather be integrated as a core functionality of the system.

***Positive-sum*:** privacy should be an additional functionality of the system. Its setting up should not be at the cost of another vital functionality such as security.

***Full lifecycle protection*:** privacy should be ensured from the very beginning of the system life as a default setting to its end. Data should be securely protected during its whole existence in the system, including during its retention and destruction, in an end-to-end fashion.

***Visibility and transparency*:** all IT or business stakeholders should be subject to independent control, and should provide clear descriptions of their operations to users and providers.

***Respect*:** IT providers should develop systems where users' privacy comes first. They should always seek for achieving and/or providing strong security, suitable reports, improved user-friendly options.

Integrating PbD in the European Legal Framework. Today, as PbD is not mandatory, it is not automatically implemented in IT systems as a design building block. Yet, Communication 2003/265 [57] has been the premise in Europe that asks IT systems to apply PbD:

> "*The Commission considers that the use of appropriate technological measures is an essential complement to legal means and should be an integral part in any efforts to achieve a sufficient level of privacy protection.*"

Since then, the European Commission has explicitly proposed PbD as a requirement in its demand for a reform and globalization of the Data Protection Directive, published in Communication 2012/011 [59].

Engineering Privacy-by-Design

The seven fundamental principles on how to achieve PbD are somehow high-level notions that leave the door open to many questions on how

to implement privacy in practice. In particular, it is not clear how PbD should be translated to the engineering point of view. Below are presented the two approaches of engineering PbD proposed by Spiekermann and Cranor in [155].

Privacy-by-policy. This approach refers to the cases where systems operators are reluctant to minimize personal data collection and processing. In such cases, operators can apply the "notice and choice" method. In a nutshell, operators are asked to provide as much information as possible about data collection and processing through a thorough privacy policy of the system. For instance, operators should publicize the privacy policies of the partners with whom they share data. Customers should be able to access their data and to choose the way their personal data will be processed and/or used after the collection (e.g., for marketing).

This approach does not really mitigate the risks of privacy breaches. It simply is a way (i) for operators to lawfully protect themselves by clearly indicating which privacy is ensured by the system, and (ii) for informed customers to unofficially make them responsible of the dissemination of their personal data by accepting or not such policy terms.

Privacy-by-architecture. This second approach is based on the structural design choices that can be applied by the operator. In fact, it is possible to define two typical kinds of architecture as follows.

***Customer-centric architecture*:** personal data always stay on the customer side/device (e.g., stored in his GSM but never transmitted to the rest of the system). This design limits privacy breaches toward the operator or an external adversary who may hack the system database. In such architectures, it is always possible to use a "trusted intermediary" that anonymizes the data sent by the customer to the operator in order to get the service. However, this design may threaten the privacy of the customer if there is no security mechanism implemented on his device that protects his personal data from an attack directly targeting his device.

***Operator-centric architecture*:** the customer device is in contact with the rest of the system in order to get the service provided by the operator. Consequently, the operator may need to collect, store

and process the customer personal data. Contrary to customer-centric architectures, this design may increase data leakage in favor of the operator or an external adversary hacking the system database. Yet, some security mechanisms can be put in place to limit these threats, such as anonymity or pseudonymity[4]. Finally, the collected data may not necessarily need to be stored in a central database, but can rather be split into several ones: this method would limit the privacy damages in case of database attacks.

Supported by Gürses, Troncoso, and Diaz in [76], engineering PbD through architectural design choices should always apply "data minimization" practices. To do so, [76] recommends operators to clearly define (i) the purpose of the system to limit abusive and improper data collection and processing, and (ii) the privacy adversaries to provide better data protection. The authors of [76] also advocate engineers to be educated on the current and well-established cryptographic building blocks and up-to-date on the security and privacy state-of-the-art research.

RFID Case: Recommendation 2009/387/EC [60]

In the framework of this thesis, the first EU communication providing several advices with respect to privacy in RFID systems is the European Commission Recommendation 2009/387/EC published in May 2009.

Main Guidelines. First of all, this act asks for the creation of a Privacy Impact Assessment (PIA) for RFID applications that should be approved by the Article 29 Working Party. We develop the exact content of this PIA in Section 8.2.3.

It states that the Data Protection and E-Privacy directives directly apply to RFID systems. Privacy and security (i.e., data confidentiality, integrity and availability) should be by-design. In any case, a joint work between stakeholders and EU institutions is asked to evaluate the information security and privacy of commercialized RFID applications:

[4]Particular attention should be paid when using pseudonymity. Indeed, since the operator may be able to recover the original data of its customers if needed, pseudonymity is not the best practice if customers want privacy protections against the operator.

> "*Member States should support the Commission in identifying those applications that might raise information security threats with implications for the general public. For such applications, Member States should ensure that operators, together with national competent authorities and civil society organizations, develop new schemes, or apply existing schemes, such as certification or operator self-assessment, in order to demonstrate that an appropriate level of information security and protection of privacy is established in relation to the assessed risks.*" (Article 6).

The recommendation further demands operators to issue a clear and concise information policy for each supplied RFID application. The policy should in particular indicate if personal data are processed by the application and if the location of tags is monitored. Furthermore, operators should inform users of the presence of RFID readers via a standardized EU logo.

The last main guideline of the recommendation targets RFID applications in retail trade. For this business sector, operators should additionally inform users of the presence of RFID tags that are positioned on or embedded in a product. In such a case, tags should be deactivated or removed from the product at the point of sale.

8.2.3 Existing Privacy Evaluation Methodologies

Some (non-)official methods can already be found to assess the privacy of IT solutions. This section surveys the most well-known ones.

Common Criteria [96]

The Common Criteria for Information Technology Security Evaluation (CC) is the ISO/IEC 15408 standard for computer security certification. It results from the unification of three existing standards (European, Canadian and American) in order to facilitate the selling of IT products so that they only need to be evaluated by one unique and comprehensive standard. The CC allows approved laboratories to test a given IT product according to some specific security requirements or claims formulated by users and providers.

The privacy class defined in Part 2 of the CC [96] is the security functional component targeting the framework of this thesis. It sets the four privacy requirements that an IT product should provide to ensure user's protection against discovery and misuse of his identity by other users, while maintaining the well-functioning of the studied product.

***Anonymity*:** the product resource or service should not disclose the user's identity[5], i.e., any potential data that identify a user.

***Pseudonymity*:** the product resource or service should not disclose the user's identity, but the user should be accountable for the resource/service use. Therefore, it should be possible to recover the real user's identity from the pseudonym attributed to the user for processing purposes related to the product.

***Unlinkability*:** multiple uses of the product resources or services by the same user should not be linkable by other users.

***Unobservability*:** the use of the product resource or service by a user should not be observable by other users.

However, the CC only defines security rules that an IT product should follow, and tests whether the privacy requirements are ensured by the product. It finally assigns a grade, called Evaluation Assurance Level (EAL1 through EAL7), to the product. This EAL grade does not actually measure the security level of an IT product, but only designates the depth to which the evaluation has been conducted.

Privacy Seals

Another approach to evaluate the privacy of an IT solution can be performed via the granting of a privacy seal. This seal is a certification along with a graphic symbol intended to inform customers that a solution is compliant with the data protection laws. The evaluation of the solution should be performed by a trusted third party. Customers should therefore be aware of that fact and be able to detect fake/forged seals

[5]Note that the anonymity requirement can however be flexible to capture either limited or full privacy policies. For instance, some authorized users such as system administrators may be allowed to know the users' identities.

by checking their validity[6]. Note however that the evaluation result is binary: either the seal is awarded to the solution or not, but the privacy level of the solution is not analyzed nor assessed.

E-commerce Privacy Seals. The most widespread privacy seals are the ones impacting the e-commerce. Indeed, an online shop may receive a privacy seal acknowledging that it fulfills the privacy requirements set by the issuing organization. These requirements are generally related to the data collection, processing and retention carried out by the site during customers purchases. The well-reputed issuing organizations are the two American TRUSTe[7] and BBBOnline[8], and the Canadian-American WebTrust[9] to name a few. Note that TRUSTe and BBBonline evaluation may additionally meet the US-EU Safe Harbor privacy principles: US operators receiving data of European customers must comply with the EU Data Protection Directive.

European Privacy Seal (EuroPriSe). In Europe, manufacturers and vendors of IT products and solutions can be granted by the famous EuroPriSe seal that certifies the product/solution compliance with the EU privacy legislation. An example of such a seal is provided in Figure 8.2.

The assessment process is conducted in two phases. The first one is the product/solution evaluation performed by accredited experts, from both legal (e.g., Ernst & Young) and technical (e.g., Siemens) privacy fields. The large catalogue of criteria includes some requirements similar to the ones of e-commerce privacy seals, such as the processing, protection and security management of personal data. The second phase is the validation of the evaluation report (with respect to its methodology, consistency and completeness) carried out by an independent certification organization (e.g., Unabhängiges Landeszentrum für Datenschutz). An EuroPriSe seal is only granted for two years: a whole reevaluation must be conducted to renew it.

[6]For instance, some malicious websites put a simple picture that is meant to represent a valid seal, but this pseudo seal cannot be checked by the customer.

[7]`http://www.truste.com/`

[8]`http://www.bbb.org/`

[9]`http://www.webtrust.net/`

Figure 8.2: An EuroPriSe seal.

Notes on Privacy Seals. In general, a privacy seal is an asset for a company. It is displayed in order to increase the customer's trust on the IT solution and expectations in the company that provides the solution. Consequently, it should not be too easy to obtain a privacy seal, otherwise such a symbol would be of no value. These symbols should neither be too numerous so that they do not bring customers confusion and do not devaluate the message disseminated by a privacy seal.

Unofficial Privacy Audits

One more independent technique to examine, verify, compare and conclude about the performances of a system in terms of privacy is undertaking an audit. In such a context, the system to analyze can be a company, an IT solution, a network, or any other kind of structure. This kind of verification can be performed by the company itself or a third party. This practical solution aims for determining the quality and integrity of privacy practices performed by the system, and enhancing their transparency.

Personally Identifiable Information (PII). Privacy audits mainly focus their analysis on the potential PIIs managed by the studied system, where PIIs are defined by the NIST[10] recommendation [124]. As examples, a PII can be a full name, a national ID number, an IP address, a driver's license number, a fingerprint, a credit card number, etc. Yet,

[10]National Institute of Standards and Technology.

the notion of PII is slightly different from the notion of personal data of Directive 95/46/EC since the NIST does not consider all personal data as PII: for instance, the gender is not a PII.

General Objectives. First of all, such evaluations periodically determine the degree of compliance of the analyzed system with (i) the applicable privacy laws and regulations, and (ii) the required privacy principles, policies and practices. Privacy audits also aim to reveal the gaps between required and actual privacy management, operational and technical controls that are put in place. Their final goal is to provide bases for improving the system privacy through the proposal of remediation steps, if needed. Note that, generally, privacy audits only consider PII data as the ones to be protected in a system.

Notes on Privacy Audits. These methodologies do not only check the IT part of the studied system. They also investigate the operational privacy issues, e.g., the potential risk if someone forgets PII unattended on a printer or fax.

RFID Privacy Impact Assessment Framework [58]

The US memorandum related to privacy for the e-government [133] defines a Privacy Impact Assessment (PIA) as follows.

> "*A PIA is an analysis of how information is handled: (i) to ensure handling conforms to applicable legal, regulatory, and policy requirements regarding privacy, (ii) to determine the risks and effects of collecting, maintaining and disseminating information in identifiable form in an electronic information system, and (iii) to examine and evaluate protections and alternative processes for handling information to mitigate potential privacy risks.*"

A PIA is further meant to assist legal institutions in determining the impact of a (potentially developing) IT initiative on individual privacy.

From the RFID Recommendation 2009/387/EC, the European Commission published in 2011 a PIA framework for RFID applications approved by the Article 29 Working Party. This document describes the

analytical guidelines to help RFID operators to assess privacy risks (and identify the measures to be taken to address them) before a new RFID application is introduced onto the market. It should additionally assist operators in establishing and maintaining the compliance with privacy and data protection laws, controlling the risks of their applications (especially in terms of operator/user privacy and user trust), as well as evaluating their PbD efforts. The RFID PIA process consists of two phases, as described below.

1. Initial analysis phase: the operator should go through the decision graph depicted in Figure 8.3 in order to determine if a PIA should be processed for its application and, if so, whether it should be a full or small scale PIA. This will mainly depend on whether the application processed personal data (as defined in the Data Protection Directive) or the tags are carried by persons.

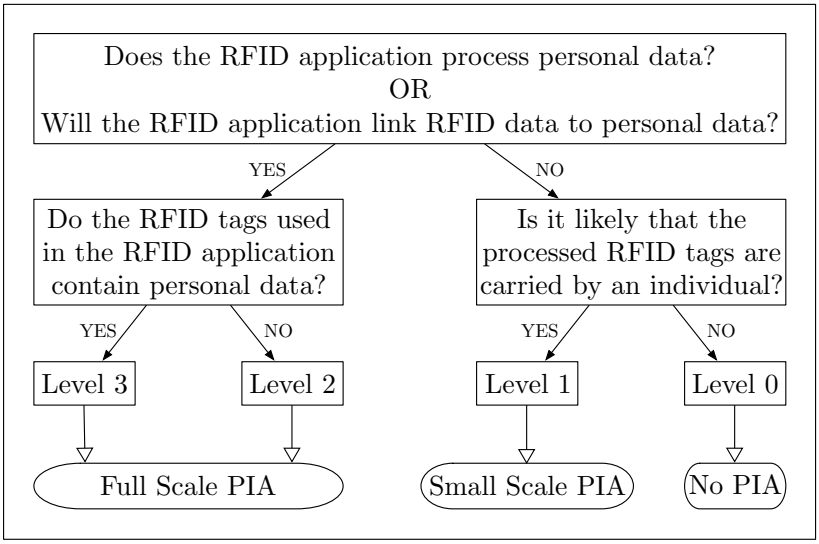


Figure 8.3: RFID PIA decision graph [58].

A full scale PIA should be more detailed than a small one, and should include all the potential risks as well as all the strategies to mitigate them. A small scale PIA should not be as comprehensive,

since the potential risks of such applications are lower than the ones of applications needing a full scale PIA.

2. Risk assessment phase:

Step 1 (RFID application characterization): the RFID operator has to provide the full description of the studied application, including the list of stored data, their storage duration, the data flows and potential interfaces with other systems.

Step 2 (Identification of the relevant risks): the RFID operator must determine the potential risks of the application (e.g., collection exceeding purpose, illegitimate data processing, secret data collection), their impact, and their probability to occur. It should not limit the analysis to the nature of stored data, and should also study the processing of data, their retention time, the mechanisms for data flows, etc.

Step 3 (Identification of the current controls): the RFID operator should analyze the controls (e.g., tags protections such as full/selective tag memory encryption, tamper resistance, deactivation/removal) put in place or planned within its application to mitigate the risks highlighted in step 2.

Step 4 (Documentation of resolution and residual risks): finally, the RFID operator should provide a report of the PIA that contains all the analyses performed in the previous steps.

Notes on the PIA. This privacy evaluation is not yet an European legal constraint: no RFID operator is mandatory to proceed a PIA. It is neither retroactive. Likewise, an application deployed in one EU country and transposed in another EU country may not necessarily be evaluated by a new PIA. This should only happen when the purposes, materials, or processed data are modified from the original application.

In the end, a PIA is a non-official (yet highly regarded) document that is able to ensure (i) the customer that his privacy is not threatened, and (ii) good faith of the RFID operator about its application.

8.3 Reaching Privacy Certification

The privacy landscape provided so far in this chapter revealed that no general methodology has been designed yet to practically assess the privacy level of UbiComp systems, or to compare the privacy levels of two systems. However, the convergence toward an official privacy certification seems inevitable if we want an international homogenization regarding the privacy assessment of UbiComp systems. This section furnishes the main ideas that could be observed to reach such a certification.

8.3.1 A More Technical View

Privacy Beyond Law

The existing legislative acts simply lay down general criteria related to data protection (e.g., non excessive process of personal data) that should be followed by a system to comply with the common will. Yet, they neither explicitly display which technical building block should be put in place to ensure privacy, nor provide methods to analyze the privacy of a system.

Additionally, laws are made by governments, and thus are distinct from one geographical region to another. Yet, the aim of a privacy certification is to be international, meaning that it does not depend on the geographical location of the system to certify. This approach differs from the one followed by privacy seals and audits.

Not an Audit

The goal of the certification is not to help an operator in improving its system, but rather to analyze a system as is: the operator should already apply the PbD principles by engineering its system using the privacy-by-architecture approach.

Additionally, we consider that privacy-by-policy is not a technical means to provide privacy (as explained in Section 8.2.2). Hence, privacy-by-policy should not be analyzed in the technical assessment part of a privacy certification.

What Privacy Certification Should Technically Analyze.

The certification should provide practical and technical methodologies capable of assessing the privacy level of a system from the cryptography and information security point of view. By generalizing the privacy notion given in the introductory chapter of this thesis to UbiComp systems, the certification should analyze the untraceability and resistance against personal information leakage properties of such systems.

Analyzing Untraceability. This notion has already been addressed in this thesis with the proposal of a formal cryptographic untraceability model for RFID systems in Chapter 7. In the context of UbiComp systems, the certification should analyze if it is possible to trace the device of a customer (and therefore the customer itself). Consequently, the model proposed in Chapter 7 could be one of the methodologies/tools provided by the certification to assess the privacy level of less restrictive UbiComp systems by simply redefining the system to analyze (see Section 7.2 for more details).

Analyzing Resistance Against Personal Information Leakage. The certification should analyze if it is possible to obtain or extract the personal data of customers from a system. The goal of the study would be to investigate how much a potential personal information leakage would harm the customers privacy.

To reach this objective, the study should firstly describe in details the UbiComp system to analyze. In particular, it should report where are located the personal data, how they are stored and transmitted (e.g., anonymously or not), and who has access to them (e.g., only the operator or also third parties). Then, the study should clearly identify the adversary goals, means and corresponding success probability in obtaining customers personal data according to the system design. Finally, it should establish the level of privacy that is ensured by the analyzed system in case of personal information leakage. For instance, the k-anonymity model[11] [158] could be one of the methodologies/tools provided by the certification to assess this level.

[11]The k-anonymity refers to the level of difficulty in uniquely identifying an individual from a set of data.

One of the inevitable future works will be to thoroughly define the way and process to analyze the resistance against personal information leakage of UbiComp systems. This could be carried out with the proposal of a formal and technical assessment methodology.

8.3.2 Comprehensive Privacy Analysis of a System

Contrary to all the works presented in this thesis, the certification should perform a comprehensive privacy analyze of an UbiComp system.

The Need of a Layer-by-layer Analysis. As a matter of fact, the analysis should take into consideration every layer of the architectural communication model of UbiComp systems (e.g., physical, communication and application layers in the case of RFID systems as described in Section 1.2.3). For instance, Avoine and Oechslin exhibited in [20] that the actual RFID privacy-friendly solutions are generally limited to the application layer, and demonstrated that the privacy of such systems is anyway threaten if no solution is additionally implemented at each layer of the system.

How to Perform a Layer-by-layer Analysis. One way to provide a comprehensive analysis is by evaluating the privacy (untraceability and resistance against personal information leakage) of each layer of the studied system. This study may also include and use the study results of the lower layers. For instance, the analysis may show that the privacy level of a given layer should be (at least) equal to the privacy level of the lower layer. Note that the information leakage analysis can be considered as useless if the layer does not contain/use any customer personal data. At the end of the analysis, each layer should obtain a grade that defines its privacy level.

Remark. The architectural layers of an UbiComp system may be implemented by different companies. For example, the manufacturer of RFID tags (who only furnishes the hardware components of the system) will implement the physical and communication layers, and the operator will only implement the application one.

8.3.3 Labeling Privacy

Defining Privacy Labels. The ultimate goal of the certification would be to grant an UbiComp system with a privacy "label". One way to establish this label is by using the results of the comprehensive analysis described in Section 8.3.2. Indeed, a global privacy grade could be computed as the weighted mean of the grades obtained by each layer of the system. All the possible global privacy grades could be split into intervals, and each interval of grades could be associated to a label. The global privacy grade of a system would thus determine its privacy label.

Note that this global privacy grade could be only one part of the privacy label. The certification could consider both technical (as presented in Section 8.3.1) and non-technical analyses during the labeling process. For instance, the certification could additionally examine if the privacy-by-policy approach is respected by the system, and take into account this result to establish its privacy label.

Customer Trust Indicator. Contrary to a seal, this label would act as a clear privacy rating of the studied system. This "trust indicator" would be easily understandable by any customer. It would develop the customer awareness of the specific privacy levels that can be provided by a given system. Customers would thus be capable of making their own decision to use a UbiComp system or not depending on its privacy label. Customers would further be able to compare the privacy level of two systems that are geared toward the same purpose. One can compare this assessment form to the well-known European energy labels [61] used for white goods as illustrated in Figure 8.4.

Always Reassessing Privacy. Finally, the certification should not grant a system a label once for good. Privacy should be periodically reevaluated, or at least each time a change occurs in the studied system (e.g., when the data storage infrastructure is revised). This demand results from the fast evolution of IT and of individual needs to be protected. Firstly, privacy assessments should be up-to-date according to the last (potentially trendy) modifications of an already evaluated system. Then, as customers' insight in privacy issues will eventually increase, especially with the growth of new technologies, privacy assessments should be adaptable according to the new customers' requirements.

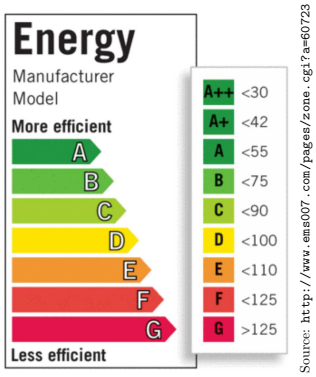


Figure 8.4: EU energy label.

8.3.4 Viability of a Privacy Certification

This section discusses the lines of inquiry that should be examined in order to decide on the viability of a privacy certification.

Which Status? The first point that should be settled is to determine whether the privacy certification should be mandatory (e.g., asked by the legal entities) or voluntary.

In the second case, the certification should be incentivizing, and operators should be led by strong motivations to invest in this kind of assessment. For instance, the certification of their system could increase their sales or improve their reputation/notoriety. This could also be a quality assurance: their system is certified to fulfill a certain privacy level. In some cases where the certification is voluntary, it may also be required to win a contract or a new market. For example, smartcards used in electronic passports should be at least EAL4+-secure certified (see Section 8.2.3 for more details on EALs).

Which Certifying Entity? The second question that should be settled is to determine the entity(ies) that would perform the privacy certification and award the label.

One possibility is to follow the same rule as the CC certification: only accredited organizations would be able to evaluate and/or award

a system as trusted third parties. For instance in France, the ANSSI[12] is the national agency in charge of awarding the CC certification to IT products once their security evaluation has been performed by a CESTI[13] (e.g., CEA-Leti).

Otherwise, the certification process can be thought as a self-certification practice carried out by the system operator itself. In this situation, the latter could ask the other companies that took part in the system implementation to perform the privacy analysis of their layers (as explained in Section 8.3.2). If the certification has a mandatory status, then some potential controls (either random or because of a complaint filing) can be put in place by the authorities to verify the operators self-certifications.

What Costs? The privacy certification should not be a long-time process that requires efforts in terms of monetary cost or workload. Otherwise, it could quickly become obsolete because of the fast evolution of IT as described in Section 8.3.3.

In particular, the fees of obtaining a privacy label should not be too expensive. Indeed, the potential profit that can result from selling a certified system should clearly outclass the certification cost. The fees should neither dissuade the operators from asking or performing the certification, especially if the latter is not mandatory.

[12] Agence Nationale de la Sécurité des Systèmes d'Information.
[13] Centre d'Évaluation de la Sécurité des Technologies de l'Information.

Conclusion

The research of this thesis was directed toward RFID technology. In particular, it targeted the field of privacy in RFID systems designed for identification and/or authentication purposes. The topic has been intensively addressed during the last decade by both the research and legal communities. This vehemence has been motivated by the ubiquity of RFID in daily life applications and, as a consequence, by the related customers' privacy concerns brought by this intrusive technology. This thesis addressed the privacy issues in RFID systems from the cryptography and information security point of view.

First of all, this thesis provided an overview of the RFID technology. It next directly got to the heart of the matter by presenting an illustration of the privacy problems that can be discovered in RFID systems. It showed a significant traceability attack on an RFID authentication protocol built with a tree-based key infrastructure.

This thesis then surveyed the eight most well-known RFID adversary models that can be used by system designers to analyze the privacy of their RFID solutions. This study disclosed a main worry: no model is able to accurately and fairly distinguish protocols designed with different cryptographic building blocks or key infrastructures. It further classified the models by comparing their features and privacy notions and demonstrated that none of these models is comprehensive or globally outclasses the other ones.

Two new kinds of attack ignored so far in privacy analyses of RFID systems have also been proposed. Firstly, the thesis stressed that an adversary may have access to a side channel that leaks the computational time of an RFID reader: an adversary can thus perform a time attack by measuring the time taken by a reader to identify and/or authenticate

a given tag. This thesis demonstrated that the privacy-friendliness of many key-reference RFID protocols is threatened by this new attack. Then, the RFID community usually considers that an adversary is able to tamper with the channel tag-reader, possibly with the tag, but rarely with the reader. When considering large-scale RFID systems, e.g., mass transportation or ticketing, the last threat is no longer a fiction. A typical case is the loss or theft of a handheld reader. This thesis formally modeled the problem of compromised readers in RFID systems and proposed two privacy-friendly authentication protocols that meet our expectations in terms of both security and privacy in this context.

As a preliminary conclusion, the thesis showed that (i) no comprehensive model able to compare protocols meaningfully has been designed so far, and (ii) security experts usually prefer using their own ad-hoc model than the existing ones which are perceived as rigid and intricate. This thesis aimed to fill that gap. It introduced an untraceability model that is operational where the previous ones were not. The model has been shaped to be extendable to fit new adversary classes and deal with new scenarios, including future evolutions of the technology. It has also been designed to benefit from a fine granularity so that it can precisely identify the untraceability level of any RFID system.

Finally, this thesis examined the privacy question in broader IT environments, namely ubiquitous computing systems. In particular, it explored the European privacy landscape in the other domains that affect customers' privacy, that is law, society and non-academic information security. The investigation exhibited the lack of standardized privacy certification intended for practically and comprehensively assessing the privacy level of such systems. The thesis ended up by furnishing the main ideas that could be followed in order to reach this objective. It especially highlighted that such a certification should be graduated, using labels with a clear pictogram as EU energy ones. As a standardized trust indicator, privacy labels could turn into a decisive factor for customers when choosing to whether or not use an ubiquitous computing system.

Bibliography

[1] Basel Alomair, Andrew Clark, Jorge Cuellar, and Radha Poovendran. Scalable RFID Systems: A Privacy-Preserving Protocol with Constant-Time Identification. In *40th Annual IEEE/IFIP International Conference on Dependable Systems and Networks – DSN 2010*, Chicago, IL, USA, June 2010. IEEE.

[2] Ross Anderson and Markus Kuhn. Tamper Resistance - A Cautionary Note. In *2nd USENIX Workshop on Electronic Commerce*, pages 1–11, Oakland, CA, USA, November 1996.

[3] Ross Anderson and Markus Kuhn. Low Cost Attacks on Tamper Resistant Devices. In Bruce Christianson, Bruno Crispo, T. Mark A. Lomas, and Michael Roe, editors, *5th International Workshop on Security Protocols*, volume 1361 of *Lecture Notes in Computer Science*, pages 125–136, Paris, France, April 1997. Springer.

[4] Ross J. Anderson. Chapter 3: Protocols. In *Security Engineering: A Guide to Building Dependable Distributed Systems*, pages 63–92. John Wiley & Sons, Inc., 2001.

[5] Frederik Armknecht, Ahmad-Reza Sadeghi, Alessandra Scafuro, Ivan Visconti, and Christian Wachsmann. Impossibility Results for RFID Privacy Notions. *Transaction on Computational Science XI*, 6480:39–63, 2010.

[6] Frederik Armknecht, Ahmad-Reza Sadeghi, Ivan Visconti, and Christian Wachsmann. On RFID Privacy with Mutual Authentication and Tag Corruption. In Jianying Zhou and Moti Yung,

editors, *8th International Conference on Applied Cryptography and Network Security – ACNS 2010*, volume 6123 of *Lecture Notes in Computer Science*, pages 493–510, Beijing, China, June 2010. Springer.

[7] Aisha Aseeri and Omaima Bamasak. HB-MP*: Towards a Man-in-the-Middle-Resistant Protocol of HB Family. In *1st International Conference on Wireless Communications and Mobile Computing – MIC-WCMC 2011*, Istanbul, Turkey, June 2011.

[8] Gildas Avoine. RFID Security & Privacy Lounge, by the UCL's Information Security Group. `http://www.avoine.net/rfid/`.

[9] Gildas Avoine. Adversary Model for Radio Frequency Identification. Technical Report LASEC-REPORT-2005-001, Swiss Federal Institute of Technology (EPFL), Security and Cryptography Laboratory (LASEC), Lausanne, Switzerland, February 2005.

[10] Gildas Avoine. *Cryptography in Radio Frequency Identification and Fair Exchange Protocols*. PhD thesis, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, December 2005.

[11] Gildas Avoine, Muhammed Ali Bingol, Xavier Carpent, and Siddika Berna Ors Yalcin. Privacy-Friendly Authentication in RFID Systems: On Sub-Linear Protocols based on Symmetric-Key Cryptography. *IEEE Transactions on Mobile Computing*, 99, September 2012.

[12] Gildas Avoine, Xavier Carpent, Benjamin Martin, and Tania Martin. Chapitre X: La Sécurité du Sans Contact et ses Specificités. In *La Carte à Puce, Vecteur de Système de Confiance*. HERMES Science, to appear, 2013.

[13] Gildas Avoine, Iwen Coisel, and Tania Martin. Time Measurement Threatens Privacy-Friendly RFID Authentication Protocols. In Siddika Berna Ors Yalcin, editor, *6th International Workshop on RFID Security – RFIDSec 2010*, volume 6370 of *Lecture Notes in Computer Science*, pages 138–157, Istanbul, Turkey, June 2010. Springer.

[14] Gildas Avoine, Iwen Coisel, and Tania Martin. A Privacy-Restoring Mechanism for Offline RFID Systems. In *5th ACM Conference on Wireless Network Security – WiSec 2012*, pages 63–74, Tucson, AZ, USA, April 2012. ACM.

[15] Gildas Avoine, Etienne Dysli, and Philippe Oechslin. Reducing Time Complexity in RFID Systems. In Bart Preneel and Stafford E. Tavares, editors, *Selected Areas in Cryptography – SAC 2005*, volume 3897 of *Lecture Notes in Computer Science*, pages 291–306, Kingston, ON, Canada, August 2005. Springer.

[16] Gildas Avoine, Cédric Lauradoux, and Tania Martin. When Compromised Readers Meet RFID. In Heung Youl Youm and Moti Yung, editors, *10th International Workshop on Information Security Applications – WISA 2009*, volume 5932 of *Lecture Notes in Computer Science*, pages 36–50, Busan, Korea, August 2009. Springer.

[17] Gildas Avoine, Benjamin Martin, and Tania Martin. Tree-Based RFID Authentication Protocols Are Definitively Not Privacy-Friendly. In Siddika Berna Ors Yalcin, editor, *6th International Workshop on RFID Security – RFIDSec 2010*, volume 6370 of *Lecture Notes in Computer Science*, pages 103–122, Istanbul, Turkey, June 2010. Springer.

[18] Gildas Avoine, Tania Martin, and Jean-Pierre Szikora. NXP Mifare Classic: Une Star Déchue. *Multi-system & Internet Security Cookbook (MISC) Magazine*, Special Issue(2), November–December 2008.

[19] Gildas Avoine and Philippe Oechslin. A Scalable and Provably Secure Hash Based RFID Protocol. In *International Workshop on Pervasive Computing and Communication Security – PerSec 2005*, pages 110–114, Kauai Island, HI, USA, 2005. IEEE.

[20] Gildas Avoine and Philippe Oechslin. RFID Traceability: A Multilayer Problem. In Andrew Patrick and Moti Yung, editors, *Financial Cryptography – FC 2005*, volume 3570 of *Lecture Notes in Computer Science*, pages 125–140, Roseau, The Commonwealth Of Dominica, February–March 2005. Springer.

[21] Gildas Avoine and Jean-Jacques Quisquater. Passport Security. In *Encyclopedia of Cryptography and Security (2nd Ed.)*, pages 913–916. Springer, 2011.

[22] Olivier Baudron, Fabrice Boudot, Philippe Bourel, Emmanuel Bresson, Johann Corbel, Laurent Frisch, Henri Gilbert, Marc Girault, Louis Goubin, Jean-François Misarsky, Phong Nguyen, Jacques Patarin, David Pointcheval, Guillaume Poupard, Jacques Stern, and Jacques Traoré. GPS - An Asymmetric Identification Scheme for on the Fly Authentication of Low Cost Smart Cards. A proposal to NESSIE, 2001.

[23] Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations Among Notions of Security for Public-Key Encryption Schemes. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO 1998*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45, Santa Barbara, CA, USA, August 1998. Springer.

[24] Salvatore Bocchetti. Security and Privacy in RFID Protocols. Master's thesis, Università degli Studi di Napoli Federico II, Italy, 2006.

[25] Carl Bosley, Kristiyan Haralambiev, and Antonio Nicolosi. HB^N: An HB-Like Protocol Secure Against Man-in-the-Middle Attacks. Cryptology ePrint Archive, Report 2011/350, 2011.

[26] Julien Bringer and Hervé Chabanne. Trusted-HB: A Low-Cost Version of HB^+ Secure Against Man-in-the-Middle Attacks. *IEEE Transactions on Information Theory*, 54(9):4339–4342, 2008.

[27] Julien Bringer, Hervé Chabanne, and Dottax Emmanuelle. HB^{++}: A Lightweight Authentication Protocol Secure against Some Attacks. In *IEEE International Conference on Pervasive Services, Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing – SecPerU 2006*, Lyon, France, June 2006. IEEE.

[28] Julien Bringer, Hervé Chabanne, and Thomas Icart. Efficient Zero-Knowledge Identification Schemes which respect Privacy. In Wanqing Li, Willy Susilo, Udaya Kiran Tupakula, Reihaneh Safavi-Naini, and Vijay Varadharajan, editors, *ACM Symposium on In-*

formation, Computer and Communication Security – ASIACCS 2009, pages 195–205, Sydney, Australia, March 2009. ACM.

[29] Mike Burmester, Breno de Medeiros, and Rossana Motta. Anonymous RFID Authentication Supporting Constant-cost Key-lookup Against Active Adversaries. *International Journal of Applied Cryptography*, 1(2):79–90, 2008.

[30] Mike Burmester, Tri van Le, and Breno de Medeiros. Provably Secure Ubiquitous Systems: Universally Composable RFID Authentication Protocols. In *Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm 2006*, pages 1–10, Baltimore, MD, USA, August–September 2006. IEEE.

[31] Mike Burmester, Tri Van Le, Breno de Medeiros, and Gene Tsudik. Universally Composable RFID Identification and Authentication Protocols. *ACM Transactions on Information and System Security – TISSEC'09*, 12(4):21:1–21:33, 2009.

[32] Levente Buttyán, Tamás Holczer, and István Vajda. Optimal Key-Trees for Tree-Based Private Authentication. In *Workshop on Privacy Enhancing Technologies – PET 2006*, Cambridge, UK, June 2006.

[33] Sébastien Canard and Iwen Coisel. Data Synchronization in Privacy-Preserving RFID Authentication Schemes. In *4th International Workshop on RFID Security – RFIDSec 2008*, Budapest, Hungary, July 2008.

[34] Sébastien Canard, Iwen Coisel, Jonathan Etrog, and Marc Girault. Privacy-Preserving RFID Systems: Model and Constructions. Cryptology ePrint Archive, Report 2010/405, 2010.

[35] Sébastien Canard, Iwen Coisel, and Marc Girauld. Security of Privacy-Preserving RFID Systems. In *IEEE International Conference on RFID Technology and Applications – RFID-TA 2010*, pages 269–274, Guangzhou, China, 2010. IEEE.

[36] Ran Canetti. Universally Composable Security: A New Paradigm for Cryptographic Protocols. Cryptology ePrint Archive, Report 2000/067, 2000.

[37] Ran Canetti. Security and Composition of Cryptographic Protocols: A Tutorial. Cryptology ePrint Archive, Report 2006/465, 2006.

[38] Jose Carrijo, Rafael Tonicelli, and Anderson C. A. Nascimento. A Fault Analytic Method against HB$^+$. Cryptology ePrint Archive, Report 2010/508, 2010.

[39] Ann Cavoukian. Privacy-by-Design. `http://privacybydesign.ca/`, 1995.

[40] David Chaum. The Dining Cryptographers Problem: Unconditional Sender and Recipient Untraceability. *Journal of Cryptology*, 1(1):65–75, 1988.

[41] Fred Cohen. *Computer Viruses*. PhD thesis, University of Southern California, 1985.

[42] Fred Cohen. Computer Viruses: Theory and Experiments. *Journal of Computers and Security*, 6(1):22–35, 1987.

[43] Iwen Coisel and Tania Martin. Untangling RFID Privacy Models. *Journal of Computer Networks and Communications*, July 2012.

[44] Robert H. Deng, Yingjiu Li, Moti Yung, and Yunlei Zhao. A New Framework for RFID Privacy. In Dimitris Gritzalis, Bart Preneel, and Marianthi Theoharidou, editors, *15th European Symposium on Research in Computer Security – ESORICS 2010*, volume 6345 of *Lecture Notes in Computer Science*, pages 1–18, Athens, Greece, 2010. Springer.

[45] Ton van Deursen. 50 Ways to Break RFID Privacy. In Simone Fischer-Hubner, Penny Duquenoy, Marit Hansen, Ronald Leenes, and Ge Zhang, editors, *6th IFIP WG 9.2, 9.6/11.7, 11.4, 11.6/PrimeLife International Summer School – Privacy and Identity Management for Life – PrimeLife 2010*, volume 352 of *IFIP Advances in Information and Communication Technology*, pages 192–205, Helsingborg, Sweden, August 2010. Springer.

[46] Ton van Deursen. *Security of RFID Protocols*. PhD thesis, University of Luxembourg, Luxembourg, 2011.

[47] Ton van Deursen, Sjouke Mauw, and Saša Radomirović. Untraceability of RFID Protocols. In Jose Antonio Onieva, Damien Sauveron, Serge Chaumette, Dieter Gollmann, and Constantinos Markantonakis, editors, *2nd IFIP WG 11.2 International Workshop on Information Security Theory and Practices – WISTP 2008*, volume 5019 of *Lecture Notes in Computer Science*, pages 1–15, Sevilla, Spain, May 2008. Springer.

[48] Klaus Dietz and J. A. P. Heesterbeek. Daniel Bernoulli's Epidemiological Model Revisited. *Mathematical Biosciences*, 180(1-2):1–21, 2002.

[49] Tassos Dimitriou. A Lightweight RFID Protocol to protect against Traceability and Cloning attacks. In *Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm 2005*, Athens, Greece, 2005. IEEE.

[50] Tassos Dimitriou. RFID-DOT: RFID Delegation and Ownership Transfer made simple. In *4th International Conference on Security and Privacy for Communication Networks – SecureComm 2008*, pages 1–8, Istanbul, Turkey, September 2008. IEEE.

[51] Danni Dolev and Andrew C. Yao. On the Security of Public-Key Protocols. *IEEE Transaction on Information Theory*, 29(2):198–208, March 1983.

[52] Danny Dolev, Cynthia Dwork, and Moni Naor. Non-Malleable Cryptography. In Cris Koutsougeras and Jeffrey Scott Vitter, editors, *23rd Annual ACM Symposium on Theory of Computing – STOC 1991*, pages 542–552, New Orleans, LA, USA, May 1991. ACM.

[53] Paolo D'Arco, Alessandra Scafuro, and Ivan Visconti. Revisiting DoS Attacks and Privacy in RFID-Enabled Networks . In *5th International Workshop on Algorithmic Aspects of Wireless Sensor Networks – ALGOSENSORS 2009*, volume 5804 of *Lecture Notes in Computer Science*, pages 76–87, Rhodes, Greece, July 2009. Springer.

[54] Jim Eagle. RFID: The Early Years 1980-1990. http://members.surfbest.net/eaglesnest/rfidhist.htm, 2001.

[55] Kaoutar Elkhiyaoui. *Security and Privacy in RFID Systems*. PhD thesis, TELECOM ParisTech, Paris, France, September 2012.

[56] EPC Global Inc. Class-1 Generation 2 UHF Air Interface Protocol Standard Version 1.2.0. http://www.epcglobalinc.org/standards/, October 2008.

[57] European Commission. Report from the Commission - First Report on the implementation of the Data Protection Directive 95/46/EC. http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:52003DC0265:EN:NOT, 2003.

[58] European Commission. Privacy and Data Protection Impact Assessment Framework for RFID Applications, January 2011.

[59] European Commission. Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL on the protection of individuals with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation). http://eur-lex.europa.eu/smartapi/cgi/sga_doc?smartapi!celexplus!prod!CELEXnumdoc&lg=en&numdoc=52012PC0011, 2012.

[60] European Commission (Viviane Reding). Commission Recommendation of 12 May 2009 on the implementation of privacy and data protection principles in applications supported by radio-frequency identification. *Official Journal of the European Union*, L(122):47–51, 2009.

[61] European Union. Council Directive 92/75/EEC of 22 September 1992 on the indication by labelling and standard product information of the consumption of energy and other resources by household appliances. *Official Journal of the European Union*, L(297):16–19, 1992.

[62] European Union. Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data. *Official Journal of the European Union*, L(281):31–50, 1995.

[63] European Union. Directive 2002/58/EC of the European Parliament and of the Council of 12 July 2002 concerning the processing of personal data and the protection of privacy in the electronic communications sector. *Official Journal of the European Union*, L(201):37–47, 2002.

[64] European Union. Directive 2006/24/EC of the European Parliament and of the Council of 15 March 2006 on the retention of data generated or processed in connection with the provision of publicly available electronic communications services or of public communications networks and amending Directive 2002/58/EC. *Official Journal of the European Union*, L(105):54–63, 2006.

[65] 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 1986*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194, Santa Barbara, CA, USA, August 1986. Springer.

[66] Dmitry Frumkin and Adi Shamir. Un-Trusted-HB: Security Vulnerabilities of Trusted-HB. In *5th International Workshop on RFID Security – RFIDSec 2009*, Leuven, Belgium, July 2009.

[67] Flavio D. Garcia, Ichiro Hasuo, Wolter Pieters, and Peter van Rossum. Provable Anonymity. In *ACM Workshop on Formal Methods in Security Engineering – FMSE 2005*, pages 63–72, Alexandria, VA, USA, 2005. ACM.

[68] Flavio D. Garcia and Peter van Rossum. Modeling Privacy for Off-Line RFID Systems. In Dieter Gollmann, Jean-Louis Lanet, and Julien Iguchi-Cartigny, editors, *9th IFIP WG 8.8/11.2 International Conference on Smart Card Research and Advanced Applications – CARDIS 2010*, volume 6035 of *Lecture Notes in Computer Science*, pages 194–208, Passau, Germany, April 2010. Springer.

[69] Henri Gilbert, Matthew Robshaw, and Yannick Seurin. Good Variants of HB$^+$ are Hard to Find. In Gene Tsudik, editor, *12th International Conference on Financial Cryptography and Data Security – FC 2008*, volume 5143 of *Lecture Notes in Computer Science*, pages 156–170, Cozumel, Mexico, January 2008. Springer.

[70] Henri Gilbert, Matthew Robshaw, and Hervé Sibert. An Active Attack Against HB$^+$ - A Provably Secure Lightweight Authentication Protocol. Manuscript, July 2005.

[71] Henri Gilbert, Matthew J.B. Robshaw, and Yannick Seurin. HB$^\#$: Increasing the Security and Efficiency of HB$^+$. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 361–378, Istanbul, Turkey, April 2008. Springer.

[72] Henri Gilbert, Matthew J.B. Robshaw, and Yannick Seurin. How to Encrypt with the LPN Problem. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *35th International Colloquium on Automata, Languages and Programming – ICALP 2008*, volume 5126 of *Lecture Notes in Computer Science*, pages 679–690, Reykjavik, Iceland, July 2008. Springer.

[73] Shafi Goldwasser and Silvio Micali. Probabilistic Encryption. *Journal of Computer and System Sciences*, 28(2):270–299, 1984.

[74] Shafi Goldwasser, Silvio Micali, and Ronald Rivest. A "Paradoxical" Solution to the Signature Problem (Extended Abstract). In *25th Annual Symposium on Foundations of Computer Science – FOCS 1984*, pages 441–448. IEEE, 1984.

[75] Zbigniew Golebiewski, Krzysztof Majcher, Filip Zagórski, and Marcin Zawada. Practical Attacks on HB and HB$^+$ Protocols. In Claudio Agostino Ardagna and Jianying Zhou, editors, *5th IFIP WG 11.2 International Workshop on Information Security Theory and Practice – WISTP 2011*, volume 6633 of *Lecture Notes in Computer Science*, pages 244–253, Heraklion, Crete, Greece, June 2011. Springer.

[76] Seda Gürses, Carmela Troncoso, and Claudia Diaz. Engineering Privacy by Design. In *4th Conference on Computers, Privacy & Data Protection – CPDP 2011*, Brussels, Belgium, January 2011.

[77] JungHoon Ha, SangJae Moon, Jianying Zhou, and JaeCheol Ha. A New Formal Proof Model for RFID Location Privacy. In Sushil Jajodia and Javier López, editors, *13th European Symposium on Research in Computer Security – ESORICS 2008*, volume 5283 of *Lecture Notes in Computer Science*, pages 267–281, Malaga, Spain, October 2008. Springer.

[78] Tzipora Halevi, Nitesh Saxena, and Shai Halevi. Using HB Family of Protocols for Privacy-Preserving Authentication of RFID Tags in a Population. In *5th International Workshop on RFID Security – RFIDSec 2009*, Leuven, Belgium, July 2009.

[79] Ghaith Hammouri and Berk Sunar. PUF-HB: A Tamper-Resilient HB Based Authentication Protocol. In Steven M. Bellovin, Rosario Gennaro, Angelos D. Keromytis, and Moti Yung, editors, *6th International Conference on Applied Cryptography and Network Security – ACNS 2008*, volume 5037 of *Lecture Notes in Computer Science*, pages 346–365, New York, NY, USA, May 2008. Springer.

[80] Gerhard Hancke. Practical Eavesdropping and Skimming Attacks on High-Frequency RFID Tokens. *Journal of Computer Security*, 19(2):259–288, March 2011.

[81] Khaled A. Harras, Kevin C. Almeroth, and Elizabeth M. Belding-Royer. Delay Tolerant Mobile Networks (DTMNs): Controlled Flooding in Sparse Mobile Networks. In Raouf Boutaba, Kevin C. Almeroth, Ramòn Puigjaner, Sherman X. Shen, and James P. Black, editors, *4th International IFIP-TC6 Networking Conference – NETWORKING 2005*, volume 3462 of *Lecture Notes in Computer Science*, pages 1180–1192, Waterloo, Canada, May 2005. Springer.

[82] Jens Hermans, Andreas Pashalidis, Frederik Vercauteren, and Bart Preneel. A New RFID Privacy Model. In Vijay Atluri and Claudia Diaz, editors, *16th European Symposium on Research in Computer*

Security – ESORICS 2011, volume 6879 of *Lecture Notes in Computer Science*, pages 568–587, Leuven, Belgium, September 2011. Springer.

[83] Herbert W. Hethcote. The Mathematics of Infectious Diseases. *SIAM Review*, 42(4):599–653, 2000.

[84] Nicholas J. Hopper and Manuel Blum. A Secure Human-Computer Authentication Scheme. Technical report, Computer Science Department, School of Computer Science, Carnegie Mellon University, May 2000.

[85] Nicholas J. Hopper and Manuel Blum. Secure Human Identification Protocols. In Colin Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 52–66, Gold Coast, Australia, December 2001. Springer.

[86] Infineon. Contactless SLE 66 Family. `http://www.infineon.com/`.

[87] Innovatron. Calypso Electronic Ticketing Standard. `http://www.calypsonet-asso.org/`, 1993.

[88] International Civil Aviation Organization. Machine Readable Travel Documents, Doc 9303, Part 1, Machine Readable Passports, Fifth Edition (2003).

[89] International Organization for Standardization. ISO/IEC 9798: Information technology - Security techniques - Entity authentication, 1991–2010.

[90] International Organization for Standardization. ISO/IEC 11784 Radio frequency identification of animals - Code structure, 1996.

[91] International Organization for Standardization. ISO/IEC 11785 Radio frequency identification of animals - Technical concept, 1996.

[92] International Organization for Standardization. ISO/IEC 15693: Identification cards - Contactless integrated circuit(s) cards - Vicinity cards, 2000–2009.

[93] International Organization for Standardization. ISO/IEC 14443: Identification cards - Contactless integrated circuit cards - Proximity cards, 2001–2008.

[94] International Organization for Standardization. ISO/IEC 18092: Information technology - Telecommunications and information exchange between systems - Near Field Communication - Interface and Protocol, 2004.

[95] International Organization for Standardization. ISO/IEC 18000: Information technology - Radio frequency identification for item management, 2008.

[96] International Organization for Standardization. ISO/IEC 15408 (Common Criteria): Information technology - Security techniques - Evaluation criteria for IT security, 2008–2009.

[97] Evan P. C. Jones, Lily Li, and Paul A. S. Ward. Practical Routing in Delay-Tolerant Networks. *IEEE Transactions on Mobile Computing*, 6(8):943–959, 2007.

[98] Ari Juels. RFID Security and Privacy: A Research Survey. *IEEE Journal on Selected Areas in Communications*, 24(2):381–394, February 2006.

[99] Ari Juels and Stephen Weis. Defining Strong Privacy for RFID. In *5th International Conference on Pervasive Computing and Communications – PerCom 2007*, pages 342–347, New York City, NY, USA, March 2007. IEEE.

[100] Ari Juels and Stephen A. Weis. Authenticating Pervasive Devices with Human Protocols. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 293–308, Santa Barbara, CA, USA, August 2005. Springer.

[101] Jonathan Katz, Ji Sun Shin, and Adam Smith. Parallel and Concurrent Security of the HB and HB$^+$ Protocols. *Journal of Cryptology*, 23(3):402–421, July 2010.

[102] Jonathan Katz and Adam Smith. Analyzing the HB and HB$^+$ Protocols in the "Large Error" Case. Cryptology ePrint Archive, Report 2006/326, 2006.

[103] Jonathan Katz and Ji Sun Shin. Parallel and Concurrent Security of the HB and HB$^+$ Protocols. In Serge Vaudenay, editor, *Advances in Cryptology – EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 73–87, Saint Petersburg, Russia, May–June 2006. Springer.

[104] Jeffrey O. Kephart and Steve R. White. Directed-Graph Epidemiological Models of Computer Viruses. In *IEEE Symposium on Security and Privacy – S&P 1991*, pages 343–359. IEEE, 1991.

[105] Marek Kumpost and Vashek Matyás. The Real Value of Private Information - Two Experiment Studies. *ERCIM News, Cybercrime and Privacy Issues*, 2012(90):44–45, July 2012.

[106] Junzuo Lai, Robert H. Deng, and Yingjiu Li. Revisiting Unpredictability-Based RFID Privacy Models. In Jianying Zhou and Moti Yung, editors, *8th International Conference on Applied Cryptography and Network Security – ACNS 2010*, volume 6123 of *Lecture Notes in Computer Science*, pages 475–492, Beijing, China, June 2010. Springer.

[107] Tri Van Le, Mike Burmester, and Breno de Medeiros. Universally Composable and Forward-secure RFID Authentication and Authenticated Key Exchange. In Feng Bao and Steven Miller, editors, *2nd ACM Symposium on Information, Computer and Communications Security – ASIACCS 2007*, pages 242–252, Singapore, Republic of Singapore, March 2007. ACM.

[108] Xuefei Leng, Keith Mayes, and Konstantinos Markantonakis. HB-MP$^+$ Protocol: An Improvement on the HB-MP Protocol. In *IEEE International Conference on RFID*, pages 118–124, April 2008.

[109] Chae Hoon Lim and Taekyoung Kwon. Strong and Robust RFID Authentication Enabling Perfect Ownership Transfer. In Peng Ning, Sihan Qing, and Ninghui Li, editors, *8th International Conference on Information and Communications Security – ICICS*

2006, volume 4307 of *Lecture Notes in Computer Science*, pages 1–20, Raleigh, NC, USA, December 2006. Springer.

[110] Anders Lindgren, Avri Doria, and Olov Schelén. Probabilistic Routing in Intermittently Connected Networks. In Petre Dini, Pascal Lorenz, and José Neuman de Souza, editors, *1st International Workshop on Service Assurance with Partial and Intermittent Resources – SAPIR 2004*, volume 3126 of *Lecture Notes in Computer Science*, pages 239–254, Fortaleza, Brazil, August 2004. Springer.

[111] Changshe Ma, Yingjiu Li, Robert H. Deng, and Tieyan Li. RFID Privacy: Relation Between Two Notions, Minimal Condition, and Efficient Construction. In Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis, editors, *16th ACM Conference on Computer and Communications Security – ACM CCS 2009*, pages 54–65, Chicago, IL, USA, November 2009. ACM.

[112] Mukundan Madhavan, Andrew Thangaraj, Yogesh Sankarasubramaniam, and Kapali Viswanathan. NLHB : A Non-Linear Hopper Blum Protocol. arXiv.org, February 2010.

[113] Sjouke Mauw, Jan Verschuren, and Erik P. de Vink. A Formalization of Anonymity and Onion Routing. In Pierangela Samarati, Peter Y. A. Ryan, Dieter Gollmann, and Refik Molva, editors, *9th European Symposium on Research in Computer Security – ESORICS 2004*, volume 3193 of *Lecture Notes in Computer Science*, pages 109–124, Sophia Antipolis, France, September 2004. Springer.

[114] Robert M. May and Roy M. Anderson. Transmission Dynamics of HIV Infection. *Nature*, 326:137–142, 1987.

[115] Alfred Menezes. Evaluation of Security Level of Cryptography: RSA-OAEP, RSA-PSS, RSA Signature. Technical Report 1011, Cryptography Research and Evaluation Committees – CRYPTREC, 2001.

[116] Alfred J. Menezes, Paul C. Van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, Inc., 1996.

[117] David Molnar and David Wagner. Privacy and Security in Library RFID: Issues, Practices, and Architectures. In Vijayalakshmi Atluri, Birgit Pfitzmann, and Patrick Drew McDaniel, editors, *11th ACM Conference on Computer and Communications Security – ACM CCS 2004*, pages 210–219, Washington, DC, USA, October 2004. ACM.

[118] Daisuke Moriyama, Shin'Ichiro Matsuo, and Miyako Ohkubo. Relation among the Security Models for RFID Authentication Protocol. In *ECRYPT Workshop on Lightweight Cryptography*, Louvain-la-Neuve, Belgium, 2011.

[119] Jorge Munilla and Alberto Peinado. HB-MP: A Further Step in the HB-Family of Lightweight Authentication Protocols. *Computer Networks*, 51(9):2262–2267, 2007.

[120] Delphine Nain, Noshirwan Petigara, and Hari Balakrishnan. Integrated Routing and Storage for Messaging Applications in Mobile Ad Hoc Networks. *Mobile Networks and Applications*, 9(6):595–604, 2004.

[121] Moni Naor and Moti Yung. Public-Key Cryptosystems Provably Secure against Chosen Ciphertext Attacks. In Harriet Ortiz, editor, *22nd Annual ACM Symposium on Theory of Computing – STOC 1990*, pages 427–437, Baltimore, MD, USA, May 1990. ACM.

[122] Roger M. Needham and Michael D. Schroeder. Using Encryption for Authentication in Large Networks of Computers. *Communications of the ACM*, 21(12):993–999, 1978.

[123] M. E. J. Newman. Spread of Epidemic Disease on Networks. *Physical Review*, 66(1), 2002.

[124] NIST (Erika McCallister, Timothy Grance, Karen A. Scarfone). Special Publication 800-122: Guide to Protecting the Confidentiality of Personally Identifiable Information (PII). *Recommendations of the NIST*, 2010.

[125] Karsten Nohl and David Evans. Quantifying Information Leakage in Tree-Based Hash Protocols. In Peng Ning, Sihan Qing,

and Ninghui Li, editors, *8th International Conference on Information and Communications Security – ICICS 2006*, volume 4307 of *Lecture Notes in Computer Science*, pages 228–237, Raleigh, NC, USA, December 2006. Springer.

[126] NXP Semiconductors. DESFire Tags. `http://www.nxp.com/products/identification_and_security/smart_card_ics/mifare_smart_card_ics/mifare_desfire/`.

[127] NXP Semiconductors. JCOP Family. `http://www.nxp.com/`.

[128] NXP Semiconductors. MIFARE Smartcards ICs. `http://www.nxp.com/products/identification_and_security/smart_card_ics/mifare_smart_card_ics/`.

[129] Philippe Oechslin. Making a Faster Cryptanalytic Time-Memory Trade-Off. In Dan Boneh, editor, *Advances in Cryptology – CRYPTO 2003*, volume 2729 of *Lecture Notes in Computer Science*, pages 617–630, Santa Barbara, CA, USA, August 2003. Springer.

[130] Supreme Court of Canada. R. v. Duarte, [1990] 1 S.C.R. 30. `http://www.canlii.org/en/ca/scc/doc/1990/1990canlii150/1990canlii150.html`, 1990.

[131] Supreme Court of Canada. Doe 1 v. AOL LLC, [2009]. `http://classactiondefense.jmbm.com/aol_class_action_defense_opn.pdf`, 2009.

[132] Australian Government Office of the Privacy Commissioner. Privacy Impact Assessment Guide. `http://privacy.gov.au/`, May 2010.

[133] Office of Management and Budget. M-03-22, OMB Guidance for Implementing the Privacy Provisions of the E-Government Act of 2002, September 2003.

[134] Miyako Ohkubo, Koutarou Suzuki, and Shingo Kinoshita. Cryptographic Approach to "Privacy-Friendly" Tags. In *RFID Privacy Workshop*, MIT, MA, USA, November 2003.

[135] Jasmina Omic, Robert E. Kooij, and Piet Van Mieghem. Virus Spread in Complete Bi-Partite Graphs. In *2nd International ICST Conference on Bio-Inspired Models of Network, Information and Computing Systems – BIONETICS 2007*, pages 49–56. ICST, 2007.

[136] Khaled Ouafi. *Security and Privacy in RFID Systems*. PhD thesis, Swiss Federal Institute of Technology (EPFL), Lausanne, Switzerland, December 2011.

[137] Khaled Ouafi, Raphael Overbeck, and Serge Vaudenay. On the Security of HB$^{\#}$ against a Man-in-the-Middle Attack. In Josef Pieprzyk, editor, *Advances in Cryptology – ASIACRYPT 2008*, volume 5350 of *Lecture Notes in Computer Science*, pages 108–124, Melbourne, Australia, December 2008. Springer.

[138] Khaled Ouafi and Raphael C.-W. Phan. Traceable Privacy of Recent Provably-Secure RFID Protocols. In Steven M. Bellovin, Rosario Gennaro, Angelos D. Keromytis, and Moti Yung, editors, *6th International Conference on Applied Cryptography and Network Security – ACNS 2008*, volume 5037 of *Lecture Notes in Computer Science*, pages 479–489, New York City, NY, USA, June 2008. Springer.

[139] Radu-Ioan Paise and Serge Vaudenay. Mutual Authentication in RFID: Security and Privacy. In Masayuki Abe and Virgil D. Gligor, editors, *3rd ACM Symposium on Information, Computer and Communications Security – ASIACCS 2008*, pages 292–299, Tokyo, Japan, March 2008. ACM.

[140] Privacy Rights Clearinghouse. Empowering Consumers, Protecting Privacy. `https://www.privacyrights.org/`.

[141] Charles Rackoff and Daniel R. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. In Joan Feigenbaum, editor, *Advances in Cryptology –CRYPTO 1991*, volume 576 of *Lecture Notes in Computer Science*, pages 433–444, Santa Barbara, CA, USA, August 1991. Springer.

[142] RFIDea. Engineering & Applications in Electronic Traceability. `http://www.rfidea.com/`.

[143] Melanie R. Rieback. *Security and Privacy of Radio Frequency Identification.* PhD thesis, Vrije Universiteit, 2008.

[144] Melanie R. Rieback, Bruno Crispo, and Andrew S. Tanenbaum. Is Your Cat Infected with a Computer Virus? In *Pervasive Computing and Communications*, pages 169–179. IEEE, 2006.

[145] Panagiotis Rizomiliotis. HB-MAC: Improving the Random-HB$^{\#}$ Authentication Protocol. In Simone Fischer-Hübner, Costas Lambrinoudakis, and Günther Pernul, editors, *6th International Conference on Trust, Privacy and Security in Digital Business – TrustBus 2009*, volume 5695 of *Lecture Notes in Computer Science*, pages 159–168, Linz, Austria, August 2009. Springer.

[146] Panagiotis Rizomiliotis and Stefanos Gritzalis. GHB$^{\#}$: A Provably Secure HB-Like Lightweight Authentication Protocol. In Feng Bao, Pierangela Samarati, and Jianying Zhou, editors, *10th International Conference on Applied Cryptography and Network Security – ACNS 2012*, volume 7341 of *Lecture Notes in Computer Science*, pages 489–506, Singapore, Republic of Singapore, June 2012. Springer.

[147] A. Sania, D.P. Kroesea, and P.K. Pollett. Stochastic Models for the Spread of HIV in a Mobile Heterosexual Population. *Mathematical Biosciences*, 208(1):98–124, 2007.

[148] Sanjay Sarma, Stephen Weis, and Daniel Engels. RFID Systems and Security and Privacy Implications. In Burton Kaliski, Çetin Kaya ço, and Christof Paar, editors, *4th International Workshop on Cryptographic Hardware and Embedded Systems – CHES 2002*, volume 2523 of *Lecture Notes in Computer Science*, pages 454–469, Redwood Shores, CA, USA, August 2002. Springer.

[149] U.S. Securities and Exchange Commission. Privacy Impact Assessment (PIA) Guide. `http://www.sec.gov/`, January 2007.

[150] Giuseppe Serazzi and Stefano Zanero. Computer Virus Propagation Models. In Maria Carla Calzarossa and Erol Gelenbe, editors, *Performance Tools and Applications to Networked Systems*, volume

2965 of *Lecture Notes in Computer Science*, pages 26–50. Springer, 2004.

[151] Victor Shoup. Sequences of Games: A Tool for Taming Complexity in Security Proofs. Cryptology ePrint Archive, Report 2004/332, 2004.

[152] Tara Small and Zygmunt J. Haas. The Shared Wireless Infostation Model: A New Ad Hoc Networking Paradigm (or Where There is a Whale, There is a Way). In *4th ACM International Symposium on Mobile Ad Hoc Networking & Computing – MobiHoc 2003*, pages 233–244. ACM, 2003.

[153] Mohammad Reza Sohizadeh Abyaneh. On the Security of Non-Linear HB (NLHB) Protocol Against Passive Attack. Cryptology ePrint Archive, Report 2010/402, 2010.

[154] Sarah Spiekermann. The Challenges of Privacy-by-Design. *Communications of the ACM*, 55(7):38–40, July 2012.

[155] Sarah Spiekermann and Lorrie Faith Cranor. Engineering Privacy. *IEEE Transaction on Software Engineering*, 35(1):67–82, January–February 2009.

[156] Thrasyvoulos Spyropoulos, Konstantinos Psounis, and Cauligi S. Raghavendra. Single-Copy Routing in Intermittently Connected Mobile Networks. In *1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks – SECON 2004*, pages 235–244. IEEE, 2004.

[157] Chunhua Su, Yingjiu Li, Yunlei Zhao, Robert H. Deng, Yiming Zhao, and Jianying Zhou. A Survey on Privacy Frameworks for RFID Authentication. *IEICE Transactions on Information and Systems*, E95.D(1):2–11, 2012.

[158] Latanya Sweeney. k-Anonymity: A Model for Protecting Privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, October 2002.

[159] Chiu C. Tan, Bo Sheng, and Qun Li. Serverless Search and Authentication Protocols for RFID. In *International Conference on*

Pervasive Computing and Communications – PerCom 2007, New York City, NY, USA, March 2007. IEEE.

[160] Pierre-Henri Thevenon. *Sécurisation de la Couche Physique des Communications Sans Contact de Type RFID et NFC*. PhD thesis, CEA-LETI, France, July 2012.

[161] Amin Vahdat and David Becker. Epidemic Routing for Partially-Connected Ad Hoc Networks. Technical report, Duke University, 2000.

[162] Serge Vaudenay. On Privacy Models for RFID. In Kaoru Kurosawa, editor, *Advances in Cryptology – ASIACRYPT 2007*, volume 4833 of *Lecture Notes in Computer Science*, pages 68–87, Kuching, Malaysia, December 2007. Springer.

[163] Bob Violino. The History of RFID Technology. `http://www.rfidjournal.com/articles/view?1338`, 2005.

[164] Charles A. Walton. Electronic Identification & Recognition System. U.S. patent 3,752,960, issue date 14 august 1973, 1973.

[165] Bongno Yoon. HB-MP^{++} Protocol: An Ultra Light-Weight Authentication Protocol for RFID System. In *IEEE International Conference on RFID*, Orlando, FL, USA, April 2009.

[166] Zhensheng Zhang. Routing in Intermittently Connected Mobile Ad Hoc Networks and Delay Tolerant Networks: Overview and Challenges. *IEEE Communications Surveys and Tutorials*, 8(1):24–37, 2006.