# Security and Privacy in RFID Systems

THÈSE N$^{O}$ 5283 (2012)

PRÉSENTÉE LE 10 FÉVRIER 2012

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS<br>LABORATOIRE DE SÉCURITÉ ET DE CRYPTOGRAPHIE<br>PROGRAMME DOCTORAL EN INFORMATIQUE, COMMUNICATIONS ET INFORMATION

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Khaled OUAFI

acceptée sur proposition du jury:

Prof. A. Lenstra, président du jury
Prof. S. Vaudenay, directeur de thèse
Prof. G. Avoine, rapporteur
Prof. D. Naccache, rapporteur
Dr Ph. Oechslin, rapporteur

EPFL
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE
Suisse
2012

إلى والدي، تقديرا لكل ما قدماه لي.

# ABSTRACT

This PhD thesis is concerned with authentication protocols using portable lightweight devices such as RFID tags. These devices have lately gained a significant attention for the diversity of the applications that could benefit form their features, ranging from inventory systems and building access control, to medical devices. However, the emergence of this technology has raised concerns about the possible loss of privacy carrying such tags induce in allowing tracing persons or unveiling the contents of a hidden package. This fear led to the appearance of several organizations which goal is to stop the spread of RFID tags. We take a cryptographic viewpoint on the issue and study the extent of security and privacy that RFID-based solutions can offer.

In the first part of this thesis, we concentrate on analyzing two original primitives that were proposed to ensure security for RFID tags. The first one, $\mathrm{HB}^{\#}$, is a dedicated authentication protocol that exclusively uses very simple arithmetic operations: bitwise AND and XOR. $\mathrm{HB}^{\#}$ was proven to be secure against a certain class of man-in-the-middle attacks and conjectured secure against more general ones. We show that the latter conjecture does not hold by describing a practical attack that allows an attacker to recover the tag's secret key. Moreover, we show that to be immune against our attack, HB#'s secret key size has to be increased to be more than 15 000 bits. This is an unpractical value for the considered applications.

We then turn to SQUASH, a message authentication code built around a public-key encryption scheme, namely Rabin's scheme. By mounting a practical key recovery attack on the earlier version of SQUASH, we show that the security of *all* versions of SQUASH is unrelated to the security of Rabin encryption function.

The second part of the thesis is dedicated to the privacy aspects related to the RFID technology. We first emphasize the importance of establishing a framework that correctly captures the intuition that a privacy-preserving protocol does not leak any information about its participants. For that, we show how several protocols that were supported by simple arguments, in contrast to a formal analysis, fail to ensure privacy. Namely, we target ProbIP, MARP, Auth2, YA-TRAP, YA-TRAP+, O-TRAP, RIPP-FS, and the Lim-Kwon protocol. We also illustrate the shortcomings of other privacy models such as the LBdM model.

The rest of the dissertation is then dedicated to our privacy model. Contrarily to most RFID privacy models that limit privacy protection to the inability of linking the identity of two participants in two different protocol instances, we introduce a privacy model for RFID tags that proves to be the exact formalization of the intuition that a private protocol should not leak any information to the adversary. The model we introduce is a refinement of Vaudenay's one that invalidates a number of its limitations. Within these settings, we are able to show that the strongest notion of privacy, namely privacy against adversaries that have a prior knowledge of all the tags' secrets, is realizable. To instantiate an authentication protocol that achieves this level of privacy, we use plaintext-aware encryption schemes. We then extend our model to the case of mutual authentication where, in addition to a tag authenticating to the reader, the reverse operation is also required.

**Key words:** Cryptography, Cryptanalysis, RFID, Authentication protocols, Message Authentication Codes, HB, SQUASH.

# RESUMÉ

Cette thèse de doctorat s'intéresse aux protocoles d'authentification utilisant des marqueurs tels que des puces à radio-identification, plus communément désignés par marqueurs RFID. Ces puces ont récemment connus un intérêt grandissant grâce à leur versatilité et les avantages qu'elles présentent pour plusieurs applications telles que la gestion d'inventaire, le contrôle d'accès et les dispositifs médicaux embarqués. Cependant, l'émergence de cette technologie induit certaines réserves quant à l'éventuelle fuite de données privées telle que la possibilité de suivre une personne d'une manière automatisée ou de reconnaître la nature d'un paquet dissimulé en accédant à distance à la puce RFID qui le caractérise. Ces craintes ont conduit à la constitution d'un certain nombre d'organisations dont le but avoué est de contrer le développement de la technologie RFID. Tout au long de cette thèse, nous étudions le problème du point de vue de la cryptographie et analysons les aspects de sécurité et de vie privée liés aux solutions basées sur des systèmes RFID.

Dans la première partie de ce document, nous nous concentrons sur l'analyse de deux solutions cryptographiques dédiées aux puces RFID et censées garantir leur sécurité. La première de ces solutions est un protocole d'authentification, appelé HB$^{\#}$, qui présente la particularité de n'utiliser que deux très simples opérations booléennes, le ET et XOR (OU exclusif) logiques. Malgré que HB$^{\#}$ soit prouvé sûr contre certaines attaques de l'intermédiaire, sa sécurité contre l'ensemble des attaques de l'homme du milieu n'est supportée que par une conjecture. Nous démontrons que cette conjecture n'est pas valide en illustrant une attaque qui permet à un attaquant de retrouver la clé secrète stockée dans un marqueur RFID. Par ailleurs, nous montrons que pour être immunisé contre notre attaque, les secrets partagés de HB# doivent avoir une taille supérieure à 15 000 bits. Pour des appareils aussi simples que les marqueurs RFID, ceci n'est pas envisageable.

Après cela, nous nous tournons vers SQUASH, un code d'authentification de message bâti autour du système de chiffrement à clé publique de Rabin. En dépit de l'hypothèse communément admise que les chiffrements à clé publique sont plus lourds à implémenter que les solutions classiques, SQUASH présente la singularité de pouvoir être implémenté sur des puces ayant des capacités aussi restreintes que celles des marqueurs RFID. Cependant, nous demontrons que la sécurité de SQUASH est indépendante de celle du chiffrement de Rabin. Afin d'arriver à ce résultat, nous illustrons une stratégie contre la version antérieure de SQUASH permettant à un adversaire de retrouver la clé secrète et cela sans avoir recourt à factoriser le modulus hérité de la fonction de Rabin. Bien que notre attaque ne s'applique pas à la version finale de SQUASH, toutes les versions de ce dernier reposent sur la même analyse de sécurité. En conséquence, notre attaque, étant indépendante du problème de factorisation qui est étroitement lié au chiffrement de Rabin, invalide les arguments de sécurité accompagnant SQUASH.

La seconde partie de cette thèse est dédiée aux aspects de vie privée liés aux marqueur RFID. Dans un premier temps, nous motivons l'importance d'étudier le degré de protection de vie privée qui est offert par un protocole RFID dans un formalisme qui comporte une définition reflétant l'étendue de ce concept. Pour cela, nous demontrons que plusieurs protocoles, plus explicitement ProbIP, MARP, Auth2, YA-TRAP, YA-TRAP+, O-TRAP, RIPP-FS et celui de Lim-Kwon, échouent à prévenir des attaques de traçage, où le but de l'adversaire est de pouvoir suivre un tag donné entre plusieurs sessions d'authentification. En parallèle, nous montrons les limites du modèle LBdM en illustrant des attaques de traçage réalistes sur des protocoles pourtant prouvés respectueux de la vie privée dans ce dernier modèle.

Le reste de ce document est dédié à notre modèle de vie privée dans les systèmes RFID. Contrairement aux modèles précédents qui réduisent le respect de la vie privée à un certain nombre de propriétés telle que l'incapacité de tracer un marqueur, nous estimons qu'un protocole RFID respecte la vie privée si aucune information ne peut être déduite par un adversaire interagissant avec des marqueurs et le lecteur. Le modèle que nous développons est une correction de celui proposé par Vaudenay, affranchi de certaines de ses limites. Concrètement, notre modèle admet la possibilité d'obtenir la forme la plus absolue de respect de la vie privée en faisant appel à des chiffrements "plaintext-aware". Dans un souci de complétude, nous proposons aussi une extension de notre modèle pour le cas de protocoles avec authentification mutuelle, dans lesquels le lecteur doit aussi s'authentifier auprès des marqueurs.

**Mots-clés :** Cryptographie, Cryptanalyse, RFID, code d'authentification de message, Protocoles d'authentification. HB, SQUASH.

# CONTENTS

# REMERCIEMENTS

*« Je n'avais d'abord projeté qu'un mémoire de quelques pages ; mon sujet m'entraînant malgré moi, ce mémoire devint insensiblement une espèce d'ouvrage trop gros, sans doute, pour ce qu'il contient, mais trop petit pour la matière qu'il traite. J'ai balancé longtemps à le publier ; et souvent il m'a fait sentir, en y travaillant, qu'il ne suffit pas d'avoir écrit quelques brochures pour savoir composer un livre. Après de vains efforts pour mieux faire, je crois devoir le donner tel qu'il est, jugeant qu'il importe de tourner l'attention publique de ce côté-là ; et que, quand mes idées seraient mauvaises, si j'en fais naître de bonnes à d'autres, je n'aurai pas tout à fait perdu mon temps. »*

Jean-Jacques Rousseau, Préface de Émile, ou De l'éducation.

Je ne saurais exprimer toute ma gratitude envers Serge Vaudenay, mon directeur de thèse. Talentueux et rigoureux dans la recherche, intègre et sincère dans ses relations, Serge n'a cessé d'être un modèle et une source d'apprentissage. Accomplir ma thèse sous sa direction fut un immense honneur et je ne pourrais jamais assez l'en remercier.

Je tiens également à remercier tous les membres du jury pour le temps qu'ils ont consacré à étudier ce manuscript. Merci à Gildas Avoine d'avoir accepté mon invitation et d'être la raison pour laquelle cette thèse traite des puces RFID. Merci à Philippe Oechslin pour tout ce qu'il m'a enseigné et pour ses conseils. Merci à David Naccache pour sa sympathie, son sens de l'humour et sa disponibilité. Merci à Arjen Lenstra pour avoir accepter de présider ce jury malgré un emploi du temps chargé et pour son humour et franc-parler.

Merci au Fond National Suisse d'avoir financé la plupart de mes travaux de recherche. Merci à ORIDAO de m'avoir permis de découvrir une autre facette de la cryptographie.

Ces années de thèse au LASEC n'auraient pas été aussi passionnantes et enrichissantes sans tout ses membres avec qui j'ai partagé d'inoubliables moments, qu'ils soient à la cafétéria d'électricité d'à coté ou à Seoul, à l'autre bout du monde. Merci à Thomas Baignères, Sylvain Pasini et Martin Vuagnoux pour leur accueil au LASEC. Merci à Raphael Phan et Jorge Nakahara

d'avoir partagé leur bureau avec moi et d'avoir supporté mes travers. Merci à Rafik Chaabouni d'être plus qu'un collègue, cela fait plus de 10 ans que nous nous sommes rencontré et il a su rester fidèle à lui-même. Merci pour ces années passées ensemble. Merci à Atefeh Mashatan pour avoir su être à l'écoute, pour ses conseils avisés, pour ses commentaires pertinents, pour le travail que nous avons accompli ensemble et pour avoir contribué à améliorer ce mémoire. Merci à Pouyan Sepehrdad pour avoir été un formidable collègue de bureau et pour tout le temps passé à relire ce mémoire. Merci à la relève : Aslı Bay et Petr Susil.

Bien entendu, je ne saurais oublier "l'âme vivante du labo", Martine Corval, et je la remercie pour sa présence et l'aide précieuse qu'elle a su me fournir.

A maintes occasions, France Faille a fait preuve d'une gentillesse incomparable et m'apporté une aide inestimable. Merci pour tout France.

Ces années de thèse furent aussi l'occasion de voyager et faire le tour du monde à assister à des conférences, présenter des résultats et écouter d'autres personnes parler. Ces voyages m'ont permis de découvrir la communauté des cryptographes avec qui j'ai eu la chance de partager des moments inoubliables, qu'ils soient au sommet de l'Eureka Tower à Melbourne ou dans la cave d'un club de Jazz à New-York. En particulier, un grand merci à Yannick Seurin, Henri Gilbert, Pascal Paillier, Nicolas Gama, Sébastien Zimmer, Kenny Patterson, Martijn Stam et Dominique Unruh.

Merci à Marijan et Cendrine pour leur accueil à Singapour. Merci en particulier à Cendrine pour toutes les fois où l'on a refait le monde.

Merci à Blaise, Marc, Urs et Yann sans qui je n'aurais jamais autant apprécié de vivre en Suisse.

Merci à Antoine et Hanane pour leur amitié.

Merci à Jérémi pour 25 ans d'amitié.

Merci à mes oncles et tantes pour leur aide et affection continue et soutenue. En particulier, merci à Azzedine et Fouzia pour leurs encouragements. Merci à Souhila d'être la grande soeur que je n'ai jamais eu.

Merci à Billel d'être un frère et un ami. Merci à Lémia pour son affection et de rester la même petite soeur.

Merci à mes parents pour leur soutien inconditionnel et pour avoir toujours cru en moi. Je ne saurais exprimer ici l'amour, l'admiration et la gratitude que je leur porte. Merci à ma mère de m'avoir fait grandir et transmis son amour pour les mathématiques, cette réussite est surtout la sienne. Merci à mon père d'avoir été présent lorsque le besoin s'en ressentait.

Merci à Nesrine de m'accompagner, de me supporter, et de me soutenir. Merci de partager mes joies et mes tristesses, mes réussites et mes échecs. Merci d'être toi.

Merci à Elias d'être là et d'enchanter ma vie.

# PERSONAL BIBLIOGRAPHY

The scientific papers that were published while completing the current thesis are listed hereafter in reverse chronological order. The entries written in bold are included in the dissertation.

[1] Atefeh Mashatan and Khaled Ouafi. Efficient fail-stop signatures from the factoring assumption. In Xuejia Lai, Jianying Zhou, and Hui Li, editors, *Information Security, 14th International Conference, ISC 2011, Xi'an, China, October 26-29, 2011. Proceedings*, volume 7001 of *Lecture Notes in Computer Science*, pages 372–385. Springer, 2011.

[2] Stéphane Badel, Nilay Dagtekin, Jorge Nakahara, Khaled Ouafi, Nicolas Reffé, Pouyan Sepehrdad, Petr Susil, and Serge Vaudenay. ARMADILLO: A multi-purpose cryptographic primitive dedicated to hardware. In *CHES 2010. Proceedings*, volume 6225 of *Lecture Notes in Computer Science*, pages 398–412. Springer, 2010.

[3] **Khaled Ouafi, Raphael C.-W. Phan, Doug Stinson, and Jiang Wu. Privacy analysis of forward and backward untraceable RFID authentication schemes. *Wireless Personal Communications*, pages 1–13, 2010.**

[4] Khaled Ouafi and Serge Vaudenay. Pathchecker: an RFID application for tracing products in suply-chains. In *RFIDSec'09 - The 5th Workshop on RFID Security, June 30 - July 2, Leuven, Belgium*, 2009.

[5] **Khaled Ouafi and Serge Vaudenay. Smashing SQUASH-0. In *EUROCRYPT 2009. Proceedings*, volume 5479 of *Lecture Notes in Computer Science*, pages 300–312. Springer, 2009.**

[6] **Khaled Ouafi, Raphael Overbeck, and Serge Vaudenay. On the security of HB# against a man-in-the-middle attack. In *ASIACRYPT 2008. Proceedings*, volume 5350 of *Lecture Notes in Computer Science*, pages 108–124. Springer, 2008.**

[7] **Khaled Ouafi and Raphael C.-W. Phan. Privacy of recent RFID authentication protocols. In *ISPEC 2008, Proceedings*, volume 4991 of *Lecture Notes in Computer Science*, pages 263–277. Springer, 2008.**

[8] **Khaled Ouafi and Raphael C.-W. Phan. Traceable privacy of recent provably-secure RFID protocols. In *ACNS 2008. Proceedings*, volume 5037 of *Lecture Notes in Computer Science*, pages 479–489. Springer, 2008.**

# 1 INTRODUCTION

The invention and subsequent widespread of the Internet and its underlining World Wide Web has provoked the development of new cryptographic applications well beyond its historical purpose of concealing transmitted messages. In this context, the problem of authentication, i.e., being ensured of the party one is communicating with, became a very central issue. The seminal work of Diffie and Hellman [DH76] paved the way for even further development. Besides introducing public-key encryption, Diffie and Hellman showed how to build a confidential channel from an authenticated one, named after them. Not only that but they also gave the idea of using digital signature schemes to construct that authenticated channel. All these primitives were combined to construct protocols dedicated to specific tasks. Among those tasks, protocols for authentication and identification arguably represent one of the most used cryptographic protocols. In short, they provide one party, called the verifier, a way to decide whether another entity with whom she is communicating is who it claims to be, or assure that a device is a trusted one. That other party is usually called the prover.

Authentication protocols are used in many contexts that essentially apply in all cases a human being needs to prove his identity. This can happen at the entrance of a sensitive building, at the borders of a country, on the phone for a secure bank transaction, or online to access some features of a website. Fortunately, several technologies were introduced to simplify these operations such as fingerprint recognition, smartcards, and RFID tags. While the first technology uses features that are beyond pure cryptography, as it calls for image capturing and pattern recognition, cryptographers considered two-factor authentication methods where, for example, a password has to be given in conjunction with the biometric data.

Despite being invented in 1968, research on cryptography for smartcards only started in the late 80's [Sch90b, Sch90a] with the development of processing units in smartcard chips. However, these devices attained their maturity and have most of their original limitations waived as they became capable of carrying rather heavy computations such as public-key encryption [PKC07]. Some of them even embed a Java Virtual Machine to execute Java bytecode [Jav08].

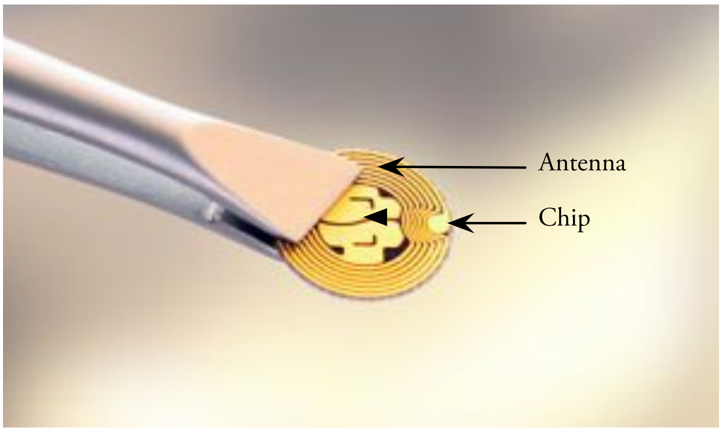

**Figure 1.1:** An RFID tag composed of an integrated circuit for storing and processing information, modulating and demodulating a radio-frequency signal and an antenna for receiving and transmitting radio signals.

Quite surprisingly, the RFID technology dates from World War II, where it was used to identify aircrafts, vehicles, or forces as friendly, making it more than 20 years older than smartcards, their wired equivalent. However, this technology did not get a practical impact until recently with the introduction of biometric passports and the industry's move towards replacing barcodes by a more efficient and equally cheap alternative. RFIDs are also widely used in animal identification, electronic vehicle registration, and public transportation payments. Other applications of the technology are also commonly found in supply-chains and addressing counterfeiting, especially in the pharmaceutical and luxury industries.

An RFID system is usually composed of a set of RFID tags that communicate with readers which in their turn are connected to a back-end database server. Tags are devices whose size can range from hundreds of $\mu$m$^2$'s to a few cm$^2$'s. Internally, they comport a radio-frequency antenna for emitting and receiving, most commonly Ultra-High Frequency (UHF), radio-signals and a chip that includes a modulator and demodulator for the signal, along with another circuit for memory, processing information, and possibly functionalities dedicated to specific tasks such as measuring the scale of a physical phenomenon. Most of today's massively deployed RFID tags do not carry a battery and are powered by a magnetic resonance induction field generated by the reader's signal. These are known as passive tags and represent the large majority of currently deployed tags. Active tags are the ones that carry a battery and operate independently from the reader's signal. Being expensive, these type of tags are reserved for high-end applications, such as the US Departement of Defense's tracking system for its inventory [oD]. Hybrid tags, refered to as semi-active, only use their battery to perform internal operations but rely on the reader's signal to power their antenna and modulator.

The current dissertation consists of two parts, corresponding to two different subjects related to RFIDs, security and privacy. In the first one, we study the security of two dedicated

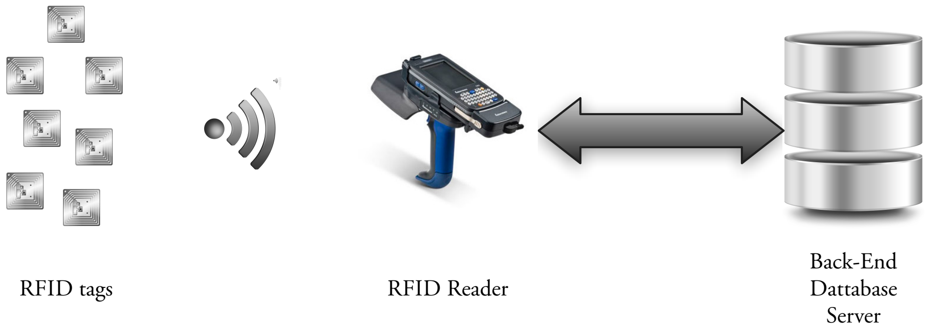

Figure 1.2: An RFID System composed of a set of RFID tags communicating wirelessly with a reader that is in itself connected to a back-end database server.

proposals for RFID tags, namely, the authentication protocol HB# and the message authentication code SQUASH. Concretely, we exhibit cryptographic attacks that enable an attacker to recover the tag's secret key. The second part of the dissertation is then dedicated to assess the need of a privacy model. We contribute to this edifice by proposing a privacy model for RFID tags that admits several levels depending on the needs of the application and the type of tags used in it.

## 1.1 The Need for Dedicated Cryptographic Primitives for RFID Tags

To ensure the protection of an RFID system, the first thing to do is to look back in the literature of cryptographic primitives and protocols and test whether any of the already proposed solutions can be applied. Unfortunately, it turns out that implementing classical algorithms is far beyond the capabilities of most encountered types of RFID tags. The standard unit of measure for this capability is the Gate Equivalent (GE). The Gate Equivalent is a unit of measure which allows to specify manufacturing-technology-independent complexity of digital electronic circuits. In today's Complementary metal–oxide–semiconductor (CMOS) technology, one GE refers to (the area taken by) one NAND Gate with two inputs and one output. For example, the circuit implementing the boolean function $a \wedge b$ needs 2 NAND gates, one for NAND and one for boolean inversion, and therefore costs 2 GE. For low-end RFID tags, it is commonly agreed that no more than 2 000 GE can be dedicated to security [JW05a]. However, to date, the best implementation of the Advanced Encryption Standard (AES) needs 3 600 GE [FDW04], which can be reduced to 3 100 GE in an encryption-only architecture [HAHH06]. Similarly, a variant of the Digital Encryption Standard (DES), in which all S-boxes are identical, can be implemented within 1 848 GE [LPPS07]. On another hand, currently deployed hash functions proved too heavy to be considered for RFID tags [FR06]. For instance, having SHA-1 on an RFID chip requires 8 120 GE, SHA-256 requires 10 868 GE, MD4 demands 7 350 GE, and MD5 fits in 8 400 GE. In a recent work, Hutter, Feldhofer, and

Wolkerstorfer [HFW11] achieved an implementation of AES and SHA-1 restricted to 128-bit messages for a processor dedicated to producing ECDSA signatures in 2 387 GE. Concerning stream ciphers, the eStream portfolio for hardware oriented ciphers, currently consisting of the two ciphers MICKEY [BD08a] and Trivium [CP08] require approximatively 3 400 GE and 2 300 GE respectively (we do not include Grain [HJMM08] in the list since it was broken by Dinur and Shamir [DS11]). While we are not aware of any cryptanalysis result on MICKEY, Trivium has been able to resist Dinur and Shamir's cube attack [DS09].

Having said that, implementing public-key cryptography is an even greater challenge. Two different directions were taken in this line of research. On one side, elliptic curve cryptography was shown to be feasible on an RFID tag, although it requires more than 10 000 GE to be implemented [HWF09]. Similar results were obtained for hyperelliptic curve cryptography using approximatively 14 500 GE [FBV09]. On the other side, the lattice-based NTRU public-key encryption scheme [HPS98] can have its encryption algorithm implemented in 2 800 GE [ABF+08]. A different approach was taken by Calmels, Canard, Girault, and Sibert [CCGS06]. In an effort to circumvent the issue of the cost of implementing modular arithmetic in hardware, they proposed to adapt the GPS identification scheme [GPS06] by loading the tag with a set of precomputed data, called coupons. This trick allowed them to reduce the amount of operations a tag needs to carry out for authentication to a small number of arithmetic additions. However, besides needing a large memory to retain all the coupons, the GPS-with-coupon scheme is susceptible to availability issues and vulnerable against Denial of Service (DoS) attacks (Calmels et al.'s proof of concept used 2 600 GE for security and stored eight coupons that would be used for eight authentications).

The shortcomings of classical cryptographic primitives to meet the needs of RFIDs led to the rise of new proposals with innovative designs. Among those proposals, we mention the block ciphers KATAN and KTANTAN [CDK09] which can be implemented in less than 800 GE. Another interesting proposal was made by Guo, Peyrin, and Poschman who simplified the AES design to propose a block cipher, called a Led [GPPR11], and a one-way hash function, called Photon [GPP11] that can fit in less than 1 000 GE. Other proposals include the hash functions Quark [AHMNP10] and Spongent [BKL+11]. However, relying on simplified designs is error prone as the confidence in the security of several newly proposed primitives is put under question. That was the case for schemes such as Present [BKL+07] for which the most recently published cryptanalysis succeeds in performing a key recovery attack on reduced rounds of the cipher [CS10, KCS11], and PRINTcipher, a block cipher proposed in [KLPR10] and fully cryptanalyzed in [LAAZ11]

In the first part of this thesis, we analyze the security of two of the most innovative proposals for RFID tags, namely, HB# [GRS08b] and SQUASH [Sha07, Sha08].

### 1.1.1 *The HB Family of Authentication protocols*

One of the most interesting directions that was taken in finding well suited primitives for RFID chips is the HB family of protocols. Contrary to other proposals which focused on designing a general-purpose lightweight cryptographic primitive such as an encryption scheme, a MAC, or a hash function, HB-like protocols do not serve any other purpose than authenticating one prover to a verifier. Moreover, by relying on a hard computational problem called LPN for learning parity with noise, the protocols offer a well-defined security guarantee.

Increasingly secure versions of the protocol were successively proposed. The original HB protocol, due to Hopper and Blum [HB01], was meant for human communicating over channels that an attacker can only eavesdrop, such as a phone communication channel, and therefore requires very simple computations that can be performed by pen and paper. Motivated by its simplicity, Juels and Weis had the idea to use the protocol in RFID tags and proposed a stronger variant called HB+ that is secure against adversaries who can directly communicate with the two parties [JW05a]. However, the protocol was shown to be insecure if an attacker could alter messages going from the reader to the tags [GRS05]. This led to the proposal of HB# which is provably secure in that model [GRS08b]. Not only that, but HB# was conjectured to be secure against adversaries who can interact with both parties at the same time, i.e., modify messages going in both directions: from the reader to the tag and vice versa. Such adversaries are generally referred to as man-in-the-middle adversaries.

Our main contribution regarding these protocols is to show that the latter conjecture does not hold. That is, we illustrate an attack that is carried out by a man-in-the-middle attacker who succeeds in recovering the parties' shared secret. Our attack is practical as the adversary only needs to trigger $2^{20}$ or $2^{35}$ protocol sessions, depending on the parameter set by Gilbert et al. proposed. We further study possible fixes of the protocol, such as limiting the number of errors the problem is putting in its answers, but it turns out that even these variants do not stand against other variants of the attack. These results were published at AsiaCrypt 2008 [OOV08].

Regarding the structure, we divided this analysis into two chapters. In Chapter 3, we give a more detailled presentation of the LPN problem and its properties. We also present the main protocols of the HB protocols, namely HB and HB+. We leave HB# and its security analysis to Chapter 4.

### 1.1.2 *From Public-Key Cryptography to MACs: The SQUASH Proposal*

Another proposal we study is rather unusual as it builds a symmetric-key cryptographic primitive, a MAC, from a function that represents the core of one of the oldest and most studied cryptosystems: The Rabin encryption scheme [Rab79]. This MAC, called SQUASH, for SQUAre haSH, was proposed by Shamir [Sha07, Sha08]. SQUASH was claimed to provide at least the same level of security Rabin's scheme provides as it consists of encrypting a mixing function of a message and the key using Rabin's function and only releasing a small number of consecutive bits from the ciphertext. Although it strips the MAC from an inversion property, which is mandatory in an encryption scheme, this last feature allowed SQUASH to be suitable for constrained environments. Pushing this reasoning even further, Shamir proposed to use Mersenne numbers of unknown factorization for the Rabin function as the special form of such numbers induces a significant simplification of Rabin's modular reduction. All these optimizations allowed Shamir to estimate the number of gate equivalents needed for the final proposal SQUASH-128 to be around half the number of gates needed by the hardware-oriented stream cipher GRAIN-128 [HJMM08], i.e., around 850 GE. Nevertheless, all those simplifications were claimed to not affect the security of the MAC. As a proof, Shamir gave a "blame-game" argument which consists of saying that any successful attack against SQUASH could be translated in an attack against the Rabin cryptosystem. Therefore, if there was any weakness in SQUASH's design, it is Rabin's scheme that should be blamed for. With the latter's security having tight bounds with the factorization problem, SQUASH was backed up with solid arguments.

Besides the practical SQUASH-128 proposal, two theoretical versions were proposed, the first one which we call SQUASH-0, proposed to use a Linear Feedback Shift Register (LFSR) that is loaded with the XOR of the key and the message to MAC. This version was proposed in [Sha07]. As it was quickly shown to be insecure when no window truncation was used, i.e., if all the Rabin ciphertext was released, Shamir proposed to replace the LFSR by a non-linear function [Sha08]. Still, both versions stood on the same security arguments. Note that SQUASH-128 is an aggressive proposal for which the blame-game argument does not hold.

In Chapter 5, we challenge SQUASH's blame-game argument by mounting a key recovery attack on the first version, SQUASH-0. We show that when using the recommended Mersenne number $2^{1\,277} - 1$ for the modulus of Rabin's function, the secret key can be recovered after $2^{10}$ pairs consisting of messages and their MACs. Of course, our attack does not rely on factoring the modulus and works by manipulating the entries of the mixing function. Unfortunately, the attack does not extend to the version in which the mixing function is a non-linear mapping. Despite that, our attack leads to the conclusion that the security guarantees behind SQUASH do not hold. So, although there is no concrete attack on the final SQUASH proposal, its exact level of security is unknown.

These results were part of an earlier research paper published at EuroCrypt 2009 [OV09].

## 1.2 Privacy Issues in RFID Systems

Besides needing dedicated cryptographic primitives to ensure security, the massive deployment of contactless devices such as RFID tags introduced a whole set of new threats related to the privacy of their wielders. Indeed, the particularity of these devices of communicating over the air presents several attractive advantages. Unfortunately, it also makes them much more vulnerable as this feature permits any entity in a reasonably close distance to monitor all their communications. Not only that, but those attackers can also access the device and interact with it at will. In spite of the specifications that may claim otherwise, this treat is even more serious to consider as the distance from which a tag can be accessed ranges from around 20cm for passive tags to more than 100m for active ones.

As Avoine and Oechslin noted [AO05], the privacy of RFIDs is a problem exceeding a single layer and needs to be addressed in every layer of the Open Systems Interconnection (OSI) model. In this dissertation, we concentrate on the higher ones and study privacy at the level of protocols.

The traditional cryptographic requirement was limited to security, which roughly summarizes in prohibiting the adversary from having access to sensitive content she should not have had access to. The nowadays availability of contactless devices introduced the possibility for a malicious adversary to trace or track an RFID tag. Being able to track such a tag constites a mean to automatically trace its holder. It was the threat caused by this leakage of privacy that led to the constitution of several organizations devoted to thwart the spread of this technology such as the Boycott Benetton campaign [Ben] and the CASPIAN groups's protest against the introduction of RFID chips in supermarkets [CAS].

In the second part of this thesis, we discuss and study how privacy can be protected in RFID authentication protocols. For that, we study the problem of formalizing what is a privacy leakage. We claim that this formalism is needed and is the reason of the failure of many authentication protocols. After that, we consider the protocols that can be used to obtain privacy preserving protocols that can be used in RFID systems.

### 1.2.1 *The Need of a Privacy Model*

To motivate the need of a privacy model, we show in Chapter 6 how several RFID protocols, whose security are either based on adhoc arguments or were proven in an inadequate model, fail to protect the privacy of the tag. More explicitly, we mount tracing attacks on ProbIP [CS07], MARP [KYK06], Auth2 [TSL07], and Tsudik's YA-TRAP [Tsu06] along with its variants YA-TRAP+ and O-TRAP [LBdM06]. We stress that all these protocols have the common property of not being supported by a rigorous security proof, but were based on rather informal arguments. The point of these cryptanalysis is mainly to demonstrate how crucial it is for protocols in general, and lightweight ones in this context, to be supported by a sound proof that quantifies the expected security.

Still, using an inappropriate model for assessing the privacy of schemes opens the door to attacks leading to privacy leakage. To illustrate this point, we give tracing attacks for the Lim-Kwon protocol [LK06] which used an adhoc model presented in the same paper to prove that the scheme is both forward and backward private (Forward privacy deals with the privacy of the scheme before the tag's secrets leak to the adversary while, backward privacy looks at sessions occurring after that leakage). We also show limitations of corruption in the model proposed by Van Le, Burmester and de Medeiros [LBdM07]. For that, we prove that O-FRAP and O-FRAKE, proposed in the same paper as an illustration of how the framework would apply to prove privacy, do not provide Forward privacy. The conclusion from this analysis is to emphasize the importance of having a model whose definitions correctly mirrors the requirements of privacy.

These results, with others not covered in this thesis, were published in two papers presented at ISPEC 2008 [OP08a] and ACNS 2008 [OP08b]. Some parts of them also appeared in an article published in the Wireless Personal Communications journal [OPSW10].

### 1.2.2 *Our Privacy Model*

Chapters 7 to 10 are devoted to present our privacy model. Our starting point is Vaudenay's work [Vau07] as we retain its underlying intuition that privacy is the inability for any adversary in extracting any information from protocol messages. We start by recalling and adapting some of its definitions in Chapter 8. We also compare the model with other proposals we describe in Chapter 7, namely Juels and Weis model [JW07] and the zero-knowledge privacy model introduced in [DLYZ10]. Our conclusion from this comparison is that any privacy leakage detected in the previous two models is detected in Vaudenay's model.

However, Vaudenay's definitions induce one unnatural result in the impossibility of designing a protocol that provides privacy protection against adversaries who have an a priori knowledge of the tag's secrets and have access to the result of protocol sessions. In Chapter 9, we argue that this impossibility is the result of a mismatch between the actual definition of privacy and the notion it aims to implement. Therefore, we update the definition to fill that gap. Moreover, we show that using a plaintext-aware public-key encryption scheme leads to a protocol achieving this level of privacy. On a side note, we show that the same level of privacy cannot be achieved by a public-key encryption scheme secure against chosen ciphertext attacks (IND-CCA2), hence, giving one of the sole applications of plaintext-awareness that is independent from IND-CCA.

That impossibility result also had implications for protocols with mutual authentication, i.e., in which the reader is also required to authenticate to the tag. In a critical work of Paise and Vaudenay's model, Armknecht, Sadeghi, Scafuro, Visconti, and Wachsmann [ASS$^+$10] showed that no protocol with mutual authentication achieves security and privacy with respect to adversaries who have knowledge of all the tags' secrets, but do not see the result of protocol instances (Such a level of security and privacy is achievable in unilateral authentication protocols by an IND-CCA2 secure encryption scheme). This result comes as a direct contradiction to Paise and Vaudenay's IND-CCA2 based protocol that was supposed to achieve it. Although we agree with Armknecht et al. on their results, we still show that Paise and Vaudenay's scheme is Forward private. We also argue why their results do not hold under our corrected model and demonstrate that the strongest form of privacy is achievable in conjonction with security for mutual authentication protocols by proposing a concrete RFID protocol. Again, we rely on plaintext-aware encryption schemes to instantiate this protocol. This extension is the subject of Chapter 10.

2

# PRELIMINARIES

Contents

## 2.1 Notations

In all this dissertation, we define a probabilistic algorithm to be an interactive Turing machine running on two tapes, one containing its inputs and the other one its randomness. An algorithm is said to be polynomial or to run in polynomial-time if it stops after a polynomial number of steps in the size of it entry tape. Algorithms can also be deterministic: Those are the ones that can be modeled by a Turing machine that only runs on a tape that contains its explicit inputs.

We use the notation $\mathcal{A}(x, y) \to z$ to refer to running the algorithm $\mathcal{A}$ with input $x$ and $y$ and obtaining $z$ as an output. When the algorithm is interactive and has access to an oracle $\mathcal{O}$, we shall denote it $\mathcal{A}^{\mathcal{O}}$. Finally, we define the view of an interactive algorithm to be its random tape and all the answers that it got from interacting with the oracles it had at its disposal. All the other messages can be computed from this view and the algorithm's description. For an algorithm $\mathcal{A}$, its view is denoted $\mathsf{view}_{\mathcal{A}}$.

For a discrete set $X$, $|X|$ refers to its cardinality, i.e., the number of elements it contains. A vector $v$ whose components are bits is called a binary vector. We also define the Hamming weight of a binary vectors as the number of 1's that it contains.

Finally, we let $\mathbf{N}$ denote the set of natural numbers, 0 inclusive, and $\mathbf{N}^\star$ denote the set of natural numbers greater than 0. Likewise, $\mathbf{Z}$ is the set of integers. $\mathbf{Z}_p$ denotes the set of positive integers smaller than $p$ and $\mathbf{Z}_p^\star$ is a subset of the former that only includes integers that a coprime with $p$.

## 2.2 Probabilities and Negligible Functions

We first start by recalling some basic definitions for probabilities. The probability mass function of a discrete probability distribution is a function $f$ such that $f(x) = \Pr[\mathbf{x} = x]$. We also recall the definition of the cumulative distribution function $F(x) = \Pr[\mathbf{x} \leq x]$.

Throughout this dissertation, we will explicitly use four probability distribution. For the sake of completeness, we describe them inhere.

- **The Uniform Distribution.** Over a discrete set, the uniform distribution assign to every entry an equal probability. That is, its probability mass function is a constant function that sums to 1 over all elements of $X$, i.e.,

  $$\forall x \in X : f(x) = |X|^{-1}.$$

  Even if it constitutes an abuse of notation, we write $x \in_R X$ to express the fact that $x$ is chosen from $X$ according to the uniform distribution.
- **The Bernoulli Distribution.** This distribution is defined over the binary set $\{0, 1\}$ and models the success of an experiment that is controlled with a probability $p$. That

is, a random variable $x$ following the Bernoulli distribution with parameter $\nu$, denoted $x \sim \mathsf{Ber}(\nu)$, takes the value 1 with success probability $\nu$ and the value 0 with probability $1 - \nu$.

- **The Binomial Distribution.** Simply put, the binomial distribution is counting how many Bernoulli trials succeed: It consists of repeating $n$ times an experiment that succeeds with probability $\nu$ and counting how many of those experiments succeeded. Therefore, the law admits two parameters, $n$ and $\nu$, and is noted $\mathsf{Binom}(n, \nu)$. For this distribution, the mean and variance compute as $n\nu$ and $m\nu(1 - \nu)$ respectively. The probability mass function is given by

$$f_{n,\mu}(x) = \binom{n}{x} \nu^x (1 - \nu)^{n-x}$$

- **The Gaussian Distribution.** Also known as the normal distribution, this distribution is often used as a first approximation to describe random variables that tend to cluster around a single mean value $\mu$ resulting in a bell-shaped distribution curve which width depends on another paramater $\sigma^2$ called the variance.

$$\varphi_{\mu,\sigma^2}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Its corresponding cumulative distribution function is denoted $\Phi(x)$.

## 2.3 Classical Cryptography

Let us consider two parties, traditionally refered to as Alice and Bob, who share a secret bitstring $K$, called the key. Alice sends a message $m$ to Bob through a communication channel that may be under the control of a malicious entity, that we call Malice. Bob receives a message $\hat{m}$ that may be different from $m$. Depending on the needs of Alice and Bob, they may want to materialize some of the properties below.

- **Integrity.** A communication channel is integer if whenever $\hat{m} \neq m$, Bob detects it. That is, no one can modify messages transiting through a noiseless channel without being detected. Note that this property does not prevent Malice from inserting or deleting messages.
- **Authenticity.** In an authenticated channel, Bob only receives messages that come from Alice and whenever Malice inserts or modifies a message transiting through the channel it is detected.
- **Confidentiality.** This property aims at preserving the secrecy of the message. That is, it ensures that no one except Bob deduces any information about the message.

In the following sections, we describe classical cryptographic tools for achieving each one of these properties.

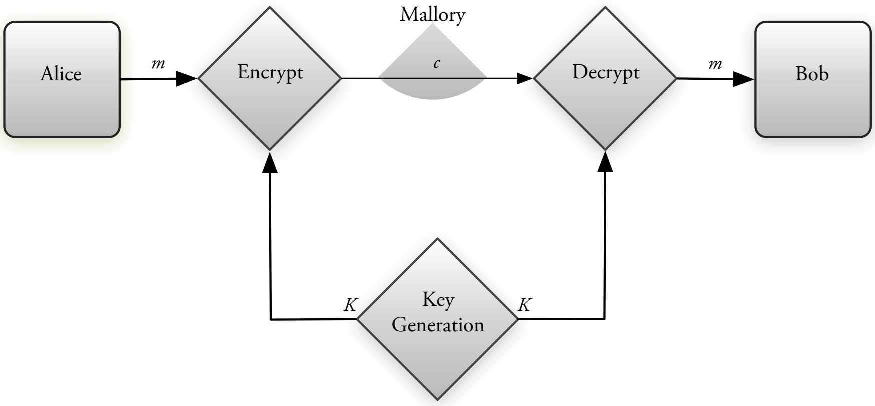

**Figure 2.1:** Components of a symmetric encryption scheme.

### 2.3.1 *Symmetric Encryption*

Depicted in Figure 2.1, symmetric encryption schemes are used to achieve confidentiality.

**Definition 2.1 (Symmetric-Key Encryption)**
*A symmetric-key encryption system is a set of three algorithms defined as follow.*

- ***Key Generation.*** $\mathsf{KeyGen} \to K$ *is an algorithm for generating the symmetric key* $K$ *that will be used by Alice and Bob to communicate privately.*
- ***Encryption.*** *For a message* $m$*, denoted plaintext,* $\mathsf{Enc}_K(m) \to c$ *produces a ciphertext. This ciphertext is sent to Bob.*
- ***Decryption.*** *Decryption is the inverse operation of encryption. That is,* $\mathsf{Dec}_K(c) \to m$ *produces the plaintext that was encrypted to* $c$ *using the same key* $K$*.*

Symmetric encryption schemes are divided into two categories: stream ciphers and block ciphers. Stream ciphers are inspired by the one-time pad, the only perfectly secure encryption scheme in the Shannon model [Sha49], with the key difference between the two being that in a stream cipher the stream that is XORed with the plaintext is only pseudo-random. Consequently, stream ciphers can encrypt messages of virtually arbitrary length. Popular stream ciphers include RC4, that is used in many protocols such as SSL, WEP and WPA, and A5/1, used in the GSM cellular telephone standard. On the other side, block ciphers impose a fixed length for the plaintext so that even shorter messages need to padded before encryption. Today's most used block ciphers include DES, 3DES, both standardized in [NIS99], AES [DR02], IDEA [LM91], and IDEA-NXT [JV04]. However, techniques, known as modes of operations, were proposed to extend the maximal length of plaintexts. The CBC mode is an example of such mode of operation.

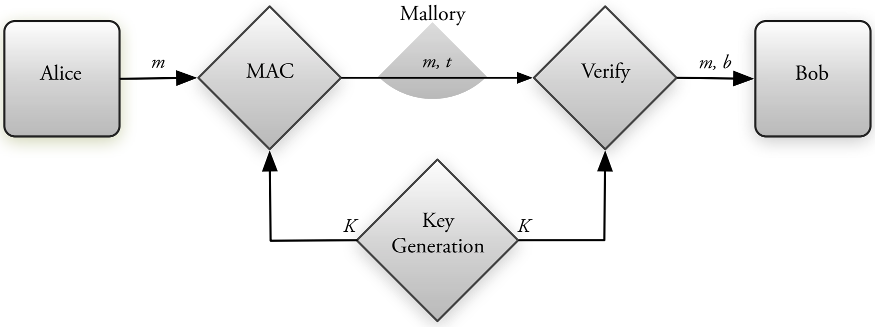

**Figure 2.2:** Components of a MAC.

Regarding their security, a stream cipher is secure if the generated bit sequence is indistinguishable from a truly random sequence. More detail on this definition will be given in Section 2.3.5. The security of block ciphers is a more elaborated case which has been subject of several definitions, ranging from Vaudenay's decorrelation theory [Vau03] to Bellare-Desai-Jokipii-Rogaway's indistinguishability based definitions [BDJR97]. In this work, we keep the most general and simple definition for its security, namely indistinguishability from a random permutation. That is, a block cipher is said to be secure if for a randomly chosen permutation $C^\star$, a uniformly distributed key $K$ and every distinguisher $\mathcal{D}$, we have

$$\left|\Pr[\mathcal{D}^{C^\star(\cdot)} \to 1] - \Pr[\mathcal{D}^{\mathsf{Enc}_K(\cdot)} \to 1]\right| \leq 2^{-\kappa}$$

### 2.3.2 *Message Authentication Codes*

While an encryption scheme ensures the confidentiality of a communication channel, it does not guarantee that Mallory cannot manipulate ciphertexts that would induce a transformation of the underlying plaintext (Note that Mallory only changes the content of the message in a certain way but this does not mean that she learnt any information about it). Message authentication codes (MACs) are tasked with achieving authentication in a communication channel . As it is depicted in Figure 2.2, a MAC is composed of the following three algorithms.

**Definition 2.2 (Message Authentication Codes - MAC)**
*A MAC is a triplet of algorithms (*KeyGen, MAC, Verify*) defined as follow*

- ***Key Generation.*** *The setup is delegated to an algorithm* KeyGen *that outputs the key $K$ that will be used by Alice and Bob.*
- ***MAC.*** *Using the secret key $K$, this algorithm generates a tag $t$ for a message $m$ given as input, i.e.,* $\mathsf{MAC}_K(m) \to t$.
- ***Verify.*** *This last algorithm is used by the recipient to assert whether a tag $t$ authenticates a message $m$. In other words,* $\mathsf{Verify}_K(m, t)$ *outputs $1$ if $t$ is a tag corresponding to the*

*message* $m$ *for the secret key* $K$*, otherwise, it outputs* $0$*. Note that it should be that every tag generated by* $\mathsf{MAC}$ *passess* $\mathsf{Verify}$ *keyed with the same key.*

The standard security requirement for MACs is called existential unforgeability under chosen message attacks. In short, it assumes that the adversary has access to an oracle to which she can submit a set of adaptively chosen messages $M = \{m_1, \ldots, m_n\}$ to a MAC oracle that when queried with $m_i$ returns $t \leftarrow \mathsf{MAC}_K(m_i)$. (In this context, adaptive refers to the adversary's ability to choose the $(i+1)^{\text{th}}$ message after receiving the oracle answer regarding the $i^{\text{th}}$ message.) In parallel, the adversary may also access an oracle for MAC verification, i.e., an oracle to which she can submit $(m, t)$ pair and learn about the bit $\mathsf{Verify}_K(m, t)$. In the end, the adversary wins if she manages to produce a pair $(m^\star, t^\star)$ such that $m^\star$ was not submitted to the MAC oracle and $\mathsf{Verify}_K(m, t)$. A MAC is then said to be secure if every such adversary limited to $2^\kappa$ basic operations does not win with a probability better than $2^{-\kappa}$. Equivalently, the MAC security experiment can be written as follow.

$$\Pr\left[\mathsf{Verify}_K(m^\star, t^\star) = 1 \,\middle|\, \begin{array}{c} K \in_R \{0,1\}^\kappa \\ (m^\star, t^\star) \leftarrow \mathcal{A}^{\mathsf{MAC}_K(\cdot), \mathsf{Verify}_K(\cdot,\cdot)} \end{array}\right] \leq 2^{-\kappa}$$

A few dedicated MACs were proposed. Instead, MACs are generally built from other symmetric-key primitives using standard transformations. For instance, the HMAC [BCK96] and UMAC [BHK$^+$99] constructions allow to build a MAC from a hash function. Other proposals, such as OMAC [IK03] and PMAC [BR02], build a MAC from a block cipher. Finally, some constructions even allow to combine encryption and MAC in a single primitive called authenticated encryption. Examples of such constructions include the EAX mode of operation [BRW04].

### 2.3.3 *Cryptographic Hash Functions*

Besides ensuring data integrity, cryptographic hash functions are cryptography's Swiss army knife, serving many purposes and appearing in almost all constructions.

A hash function family is a set of functions mapping arbitrary large strings to a fixed size output called the hash. Mathematically, for a key $K \in \{0,1\}^\kappa$, we consider a set of functions $H_K : \{0,1\}^\star \to \{0,1\}^n$. The usual security requirements for hash functions are listed below in increasing strength order.

- **First Pre-image Resistance.** A hash function is said to be (first) preimage resistant if for a randomly chosen $K$, given a hash $y$, it is infeasible to find a $x$ such that $y \leftarrow H_K(x)$. In other words,

$$\Pr\left[H_K(\mathcal{A}(y)) = y \,\middle|\, \begin{array}{c} K \in_R \{0,1\}^\kappa \\ y \in_R \mathsf{Range}(H_K) \end{array}\right] \leq 2^{-n}$$

- **Second Pre-image Resistance.** This definition is similar to the first pre-image resistance except that the adversary is already given one pre-image of the hash $y$ and is tasked with finding another one. That is, a hash function is secure against second pre-image attacks if

$$\Pr\left[ H_K(x) = H_K(x_0) \wedge x \neq x_0 \;\middle|\; \begin{array}{c} K \in_R \{0,1\}^\kappa \\ x_0 \in_R \mathsf{Domain}(H_K) \\ x \leftarrow \mathcal{A}(x_0) \end{array} \right] \leq 2^{-n}$$

- **Collision Resistance.** This is the strongest attack as it grants to the adversary the power to choose the hash for which she has to provide two different values $x_1 \neq x_2$ such that $H_K(x_1) = H_K(x_2)$. Contrarily to the two attacks before, there exists an attack in $2^{n/2}$ due to the birthday paradox (This attack consists in picking two random values and checking whether they are mapped to the same value and repeating until the condition is satisfied). A hash function family is then said to be collision resistant if no adversary can do better than the birthday attack.

Although considering a family of functions is essential for the correctness of the definition of collision resistance, widely used hash functions such as SHA-1, SHA-2 [NIS02], and even the current SHA-3 finalists, consist of a single function. Clearly, when only pre-image resistance is required those hash functions can fit the

### 2.3.4 *Universal Hash Functions*

In many situations, the hardness of finding collisions is a too strong requirement and we only need that collisions for two random values happen with a small enough probability. This is the case for example in hash tables and for extracting randomness [ILL89]. The functions satisfying this notion are called universal hash functions.

**Definition 2.3 (Universal Hash Function Family)**
*A universal hash function family is a family of functions $H_{K\in\{0,1\}^\kappa} : \{0,1\}^\ell \to \{0,1\}^n$ that satisfies the following property.*

$$\forall x, y \in \{0,1\}^\ell, \text{s.t.} x \neq y : \Pr_{h \in H}[H_K(x) = H_K(y) | K \in_R \{0,1\}^\kappa] \leq 2^{-n}.$$

Note that contrarily to most cryptographic primitives universal hash functions does not need to rely on any assumption. Instead, they can be instantiated using simple modular arithmetic as in the Carter-Wegman construction [CW77].

### 2.3.5 *Pseudo-Random Functions*

As true randomness is difficult to obtain, it is essential to have a mean to generate sequences that look random to the adversary. Two cryptographic primitives can be used to implement

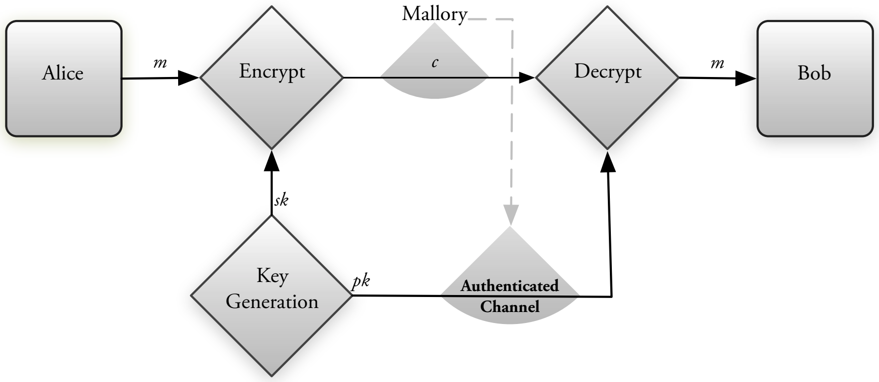

**Figure 2.3:** Components of a Public-key encryption scheme.

such functionality: Pseudo-random functions (PRF) and pseudo-random number generators (PRNG).

**Definition 2.4 (Pseudo-Random Function - PRF)**
*Let $F_{K \in \{0,1\}^\kappa} : \{0,1\}^\ell \to \{0,1\}^n$ be a family of functions indexed by a key $K$. We say that $F$ is a family of pseudo-random functions if it satisfies*

- *For every $K \in \{0,1\}^\kappa$ and every $x \in \{0,1\}^\ell$, $F_K(x)$ is computable in polynomial-time.*
- *For a randomly chosen $K$, $F_K$ is indistinguishable from a function $R$, chosen randomly among the set of functions from $\{0,1\}^\ell$ to $\{0,1\}^n$. In other words, for every distinguisher $\mathcal{D}$, we have that*

## 2.4 Public-Key Encryption Schemes

Public-key encryption, also known as asymmetric encryption, is one of the greatest achievements of modern cryptography. It allows one sender to encrypt messages that only a particular receiver can decrypt and that with only having a public key. On the other hand, the receiver is given a secret key that is used for decryption. That is, anyone with access to the public key can encrypt messages that only the receiver can decrypt.

**Definition 2.5 (Public-Key Encryption Scheme)**
*A public-key encryption scheme is a triplet of algorithms (*$\mathsf{KeyGen}$, $\mathsf{Enc}$, $\mathsf{Dec}$*) defined as follows*

- $\mathsf{KeyGen}(1^k) \to (sk, pk)$. *This is the key generation algorithm. On input a security parameter $k$, written in unary, this algorithm generates in polynomial-time a secret key $sk$ that is securely transmitted to its intended owner, while a public key $pk$ is published. The latter also characterises a, usually finite, message space $M$. Note that this algorithm has to be probabilistic.*

| **Adversary**<br>$(\mathcal{A}_1, \mathcal{A}_2)$ | **Common Input:**<br>$k \in \mathbf{N}$ | **Challenger**<br>$\mathcal{C}$ |
|---|---|---|
|  | $\xleftarrow{\quad pk \quad}$ | $(sk, pk) \leftarrow \mathsf{KeyGen}(1^k)$ |
| $(m_0, m_1) \leftarrow \mathcal{A}_1(pk)$ | $\xrightarrow{\quad m_0, m_1 \quad}$ | $b \in_R \{0, 1\}$ |
| $\hat{b} \leftarrow \mathcal{A}_2(c)$ | $\xleftarrow{\quad c \quad}$ | $c \leftarrow \mathsf{Enc}_{pk}(m_b)$ |
|  | **Win if** $\hat{b} = b$ |  |

**Figure 2.4:** The IND-CPA Security Experiment.

- $\mathsf{Enc}_{pk}(m) \rightarrow c$. *This, usually probabilistic, polynomial-time algorithm is used to encrypt a message* $m \in M$ *under the public key* $pk$ *by forming a ciphertext* $c$*.*
- $\mathsf{Dec}_{sk}(c) \rightarrow m$. *This last algorithm is used to decrypt a ciphertext* $c$*. That is, given* $c$ *and the secret key* $sk$*, the algorithm recovers* $m$ *in a polynomial-time number of steps.*

*Obviously, these algorithms have to be consistent with each other in the sense that using a secret key* $sk$ *to decrypt a ciphertext* $c$ *that is the encryption of a message* $m$ *under the corresponding public key* $pk$ *yields* $m$*. In other words,*

$$\forall k \in \mathbf{N}: \quad \Pr\left[ \mathsf{Dec}_{sk}(c) = m \;\middle|\; \begin{array}{c} \mathsf{KeyGen}(1^k) \rightarrow (sk, pk) \\ m \in_R M \\ \mathsf{Enc}_{pk}(m) \rightarrow c \end{array} \right] = 1$$

Regarding the security of encryption schemes, we retain the two classical notions of semantic security and non-malleability. Semantic security formalizes the fact that ciphertexts conceal all information about their underlying plaintexts. This property is captured by indistinguishability under chosen message attacks, commonly abbreviated IND-CPA [GM82].

**Definition 2.6 (IND-CPA Security)**
*We consider the IND-CPA security experiment shown in Figure 2.4. A scheme is called IND-CPA secure, if no probabilistic polynomial-time adversary wins the IND-CPA experiment with an advantage greater than a negligible function of the security parameter. In other words, for every probabilistic polynomial-time two-stage algorithm* $(\mathcal{A}_1, \mathcal{A}_2)$*, IND-CPA security requires that*

$$\left| \Pr\left[ \mathcal{A}_2(c, \mathsf{st}) = b \;\middle|\; \begin{array}{c} \mathsf{KeyGen}(1^k) \rightarrow (sk, pk) \\ \mathcal{A}_1(pk) \rightarrow (m_0, m_1, \mathsf{st}) \\ b \in_R \{0, 1\} \\ \mathsf{Enc}_{pk}(m_b) \rightarrow c \end{array} \right] - \frac{1}{2} \right| = \mathsf{negl}(k)$$

IND-CPA secure encryption schemes include classical examples such as the Goldwasser-Micali cryptosystem [GM82], Elgamal's encryption scheme [Elg85], and Paillier's encryption scheme [Pai99].

| Adversary | Common Input: | Challenger |
|---|---|---|
| $(\mathcal{A}_1, \mathcal{A}_2)$ | $k \in \mathbf{N}$ | $\mathcal{C}$ |
| | $\xleftarrow{\quad pk \quad}$ | $(sk, pk) \leftarrow \mathsf{KeyGen}(1^k)$ |
| | $\xrightarrow{\quad m_0, m_1 \quad}$ | $b \in_R \{0, 1\}$ |
| $\hat{b} \leftarrow \mathcal{A}_2(c)$ | $\xleftarrow{\quad c \quad}$ | $c \leftarrow \mathsf{Enc}_{pk}(m_b)$ |
| | **Win if** $\hat{b} = b$ | |

**Figure 2.5:** The IND-CCA2 Security Experiment.

Non-malleability is a stronger notion, not only requiring that no adversary can learn any information on the message but also mandating that it is not possible to transform the encrypted plaintext by applying some operations. This level of security is clearly higher and more difficult to attain than IND-CPA security. More formally, it was shown to correspond to indistinguishability against chosen-ciphertext attacks, IND-CCA for short [RS92]. In short, this notion is similar to the IND-CPA property except that the adversary, in both phases, can query a decryption oracle on every ciphertext but $c$. A weaker notion in which only $\mathcal{A}_1$ is granted that access has been considered by Naor and Yung [NY90]. To distinguish both variants, the former notion is commonly refered to as IND-CCA2 security and the later one by IND-CCA1 security. We give the formal definition of IND-CCA2 security.

**Definition 2.7 (IND-CCA Security)**
*Let us consider the IND-CCA2 security experiment depicted in Figure 2.5. A public-key encryption scheme is called IND-CCA2 secure if no probabilistic polynomial-time adversary wins the IND-CCA2 experiment with an advantage greater than a negligible function of the security parameter. In other words, for every probabilistic polynomial-time two-stage algorithm* $(\mathcal{A}_1, \mathcal{A}_2)$*, IND-CCA2 security requires that*

$$\left| \Pr\left[ \mathcal{A}_2^{\mathcal{O}_{\mathsf{Dec}}}(c, \mathsf{st}) = b \;\middle|\; \begin{array}{c} \mathsf{KeyGen}(1^k) \to (sk, pk) \\ \mathcal{A}_1^{\mathcal{O}_{\mathsf{Dec}}}(pk) \to (m_0, m_1, \mathsf{st}) \\ b \in_R \{0, 1\} \\ \mathsf{Enc}_{pk}(m_b) \to c \end{array} \right] - \frac{1}{2} \right| = \mathsf{negl}(k)$$

*When the probability above only holds for adversaries that are such that* $\mathcal{A}_2$ *does not query* $\mathcal{O}_{\mathsf{Dec}}$*, then the scheme is said to be IND-CCA1 secure.*

RSA-OAEP [BR95a] and Rabin-SAEP [Bon01] are two examples of systems that achieve IND-CCA2 security (in the Random Oracle model explained in Section 2.6). The Cramer-Shoup [CS98] cryptosystem is also IND-CCA2 secure, but in the standard model.

## 2.5 Hybrid Encryption

Public-key encryption has contributed to the simplification of key management, reducing the number of keys in a network of $n$ users from $n(n-1)$ to $n$. Despite that, the price to pay for this simplification is that encrypting with a public-key scheme is much slower than doing so with a symmetric encryption scheme. Another limitation of public-key encryption relates to the fixed length of plaintexts: While it is possible to encrypt arbitrary long message using modes of operations, there is no generic way to extend the message space of a public-key encryption scheme.

Therefore, to combine the advantages of both types of encryptions, a dual mechanism of key encapsulation mechanism (KEM) coupled with a data encapsulation mechanism (DEM) was proposed. Simply put, this system works by having a random symmetric key encrypted in a ciphertext which is used in a symmetric scheme to encrypt the data to send. More explicitly, we give the following definition.

**Definition 2.8 (The KEM/DEM Paradigm of Hybrid Encryption)**
*A KEM consists of the following algorithms*

- $\mathsf{KeyGen}(1^\lambda) \to (sk, pk)$. *This first probabilistic polynomial-time algorithm generates the pair of keys.*
- $\mathsf{KEM.Enc}_{pk}() \to (K, C)$. *Taking no input, this algorithm produces a symmetric key $K$ for the DEM defined below and outputs its corresponding ciphertext $C$.*
- $\mathsf{KEM.Dec}_{sk}(C) \to K$. *Via this algorithm the receiver makes use of his secret key $sk$ to recover the symmetric key $K$.*

*A DEM is then defined as for a symmetric encryption scheme without key generation.*

- $\mathsf{DEM.Enc}(K, m) \to c$. *Using this algorithm, the sender encrypts the message $m$.*
- $\mathsf{DEM.Dec}(K.c) \to m$. *By this algorithm, the receiver decrypts $c$ and gets the underlying ciphertext $m$.*

From a very high level, a KEM/DEM can be seen as a public-key encryption scheme. Since they mimic public-key and symmetric-key encryption schemes, it is rather easy to define similar security properties for KEMs and DEMs.

## 2.6 The Random Oracle Model

The Random Oracle (RO) model consists of replacing hash functions by black-box oracles that produces uniformly distributed outputs [BR93]. The RO model has been useful in proving the security of many schemes with fairly simpler design than comprable ones with security proofs in the standard model. There is however a separation between the two as Canetti, Goldreich, and Halevi demonstrated [CGH98]. Nevertheless, their construction is rather artificial and the impact of replacing the random oracle by a traditional hash function in a more conventional design is yet to be clearly outlined.

**Definition 2.9 (Random Oracle)**
*A random oracle over $\{0,1\}^n$ is an algorithm managing a table $\mathcal{T}$, initially empty, which receives bit-strings $x$ of arbitrary length as queries and answers as follow:*

- *If $\mathcal{T}$ already contains an entry $(x, y)$, then it simply returns $y$.*
- *Otherwise, it picks a random $y \in \{0,1\}^n$, inserts the pair $(x, y)$ in $\mathcal{T}m$ and finally returns $y$.*

## 2.7 Proof Techniques

### 2.7.1 *Hard Problems*

Proofs of security in cryptography are usually reductions. That is, they transform an algorithm performing an attack on a system in a certain model to an adversary against a believed-to-be-hard computational problem, in the sense that no probabilistic polynomial-time adversary can solve it. However, in some sense cryptographic hard problems are harder than $\mathcal{NP}$-Complete problems in the sense that they require that a randomly chosen instance of the problem is hard to solve where $\mathcal{NP}$-Completeness deals with the same issue but for all problems. It then follows that $\mathcal{P} \neq \mathcal{NP}$ is not a sufficient condition for the existence of these problems. However, if $\mathcal{P} = \mathcal{NP}$ then no such problem would exist.

Typical conjectured hard problems include

- **The Factorization Problem.** For two prime numbers $p$ and $q$, define $n = pq$. The factoring problem is to recover $p$ and $q$ from $n$. Rabin's cryptosystem is based on the assumption [Rab79]. Recovering the secret key in RSA [RSA78] is also as hard as solving this problem.
- **The RSA Problem.** This is also known as the $e^{\text{th}}$-root problem. Given a hard to factor integer $n$, an integer $e < n$ such that $\mathsf{gcd}(e, \varphi(n)) = 1$, and $y \in_R \mathbf{Z}_n^\star$, compute $x$ such that $x^e = y \pmod{n}$. The reason this problem is named this way is because it is the computational assumption on which the security of the RSA cryptosystem stands.
- **The Discrete Logarithm Problem.** Given a generator $g$ of a cyclic group $G$ (typically the multiplicative group of a finite field or an elliptic curve group) and $y \in_R G$, find $x$ such that $g^x = y$. The security of the Elgamal cryptosystem against key recovery attacks rests on this problem.
- **The Diffie-Hellman Problem.** This is the problem induced by the Diffie-Hellman key agreement protocol. In this problem $g$ a generator of a cyclic group $G$ and $x, y$ are two integers. The problem is, given $g$, $g^x$, and $g^y$, to recover $g^{xy}$.

- **The Decisional Diffie-Hellman Problem.** The DDH problem is to distinguish between a Diffie-Hellman triplet, i.e., $(g^x, g^y, g^{xy})$, where $g$ is a generator of a cyclic group $G$ and $x$, $y$ are randomly chosen integers, from a triplet $(g^x, g^y, R)$, where $R$ is a random uniformly distributed over $G$. The semantic security of the Elgamal cryptosystem relies on the hardness of this problem.

In the next two sections, we review classical techniques to reduce the security of a cryptosystem to a certain mathematical problem.

### 2.7.2 *The Simulation Paradigm and Hybrid Arguments*

Intuitively, the best way to express the property that a secure cryptographic functionality is required to not leak any information to an attacker interacting with it in a non-predictable way. The classical way to approach the problem is due to Goldwasser and Micali [GM82] who formalized this requirement by saying that the adversary does not learn any information from interacting with a system then it should be possible to replace all the responses computed by the system by adequate "fake" messages without effectively disturbing the output of the adversary. Being independent from the cryptographic system, these fake messages can be generated by a third party that is called the simulator. For instance, we expect from a secure block cipher that ciphertexts are indistinguishable from random bit-strings so that we can define a simulator that replaces those ciphertexts by random elements.

This simulation is sufficient when the adversary has only one access to the functionality, e.g., to an encryption oracle. However, the situation may be more complicated when the adversary produces adaptively chosen queries. The common technique to deal with these issues is to consider intermediate adversaries.

Now assume that the adversary is making a polynomially-bounded $q$ queries to that encryption oracle and the goal is to obtain an adversary who only gets access to random elements and produces an indistinguishable output. We define $q+1$ intermediate adversaries, denoted $\mathcal{A}_0, \ldots, \mathcal{A}_i, \ldots, \mathcal{A}_q$, called hybrids, such that the $i$ first queries of $\mathcal{A}_i$ are handled by the encryption oracle and the rest of them are processed by the simulator. The idea is to show that if $\mathcal{A}_i$ and $\mathcal{A}_{i+1}$ produce indistinguishable distributions, in the sense that the distance between the two output distributions is negligible, then by triangle inequality $\mathcal{A}_0$ and $\mathcal{A}_q$ produce indistinguishable distributions. To conclude, the latest two algorithms respectively correspond to the one that only accesses the "real" oracle and to the one that accesses the simulator. The proof technique in its whole is called a hybrid argument.

### 2.7.3 *The Game Proof Methodology*

It is often the case that a cryptosystem relies on more than one assumption to be proven secure. For these kind of systems, the simulation paradigm shows its limitations as it results in proofs that are often complex to follow and verify.

Instead, the game proof methodology allows to treat each case at once by considering a number of "intermediate games". The proof starts by considering a game, denoted Game 0, played by an adversary $\mathcal{A}$ against a challenger that simulates the environment for $\mathcal{A}$ and ensures that she follows the game description. The adversary then wins at the end if an event $S_0$ occur. The proof consists of iteratively tweaking the game until we obtain an adversary literally attacking a mathematical problem. $S_i$ denotes the event that the adversary wins in Game $i$. Three different types of transitions are usually considered [Sho04].

- **Transitions based on indistinguishability.** In such a transition, a small change is made that, if detected by the adversary, would imply an efficient method of distinguishing between two distributions that are indistinguishable (either statistically or computationally).
- **Transitions based on failure events.** In such a transition, we argue that Games $i$ and $i + 1$ proceed identically unless a certain "failure event" $E$ occurs. Using the difference Lemma, it can be shown that the statistical distance between the two games, i.e., $\Pr[S_i] - \Pr[S_{i+1}]$, can be bounded by $\Pr[E]$. Therefore, as long as $E$ occurs with negligible probability, the transition goes unnoticed to the adversary.
- **Bridging Games.** These transitions are generally used to make the proof simpler by reformulating how certain quantities or variables are computed.

# Part I

## THE SECURITY OF RFID PRIMITIVES

3

# THE LPN PROBLEM AND THE HB FAMILY

Contents

With the limitations of theoretically secure cryptography, it was natural to try to design cryptographic primitives that rely on problems that are computationally hard to solve. The choice of NP-Complete problem.

In this Chapter, we review the

## 3.1 The LPN Problem

### 3.1.1 *Definition of the Problem*

We consider the problem of recovering a secret $k$-bit vector $x$. For that purpose, we are given an oracle $\mathcal{O}_x$ which knows the vector $x$ and, on each request, answers with a uniformly chosen $k$-bit vector $a$ and a bit equal to $a \cdot x$, where the operation $\cdot$ is the scalar product. In other words, the output distribution of the oracle is

$$\{(a, a \cdot x) : a \in_R \{0,1\}^k\}.$$

This problem is simple to solve using algebraic techniques such as Gauss elimination. All that is needed for such an algorithm to recover $x$ is $k$ linearly independent vectors $a$. The cost of an unoptimized algorithm implementing Gaussian elimination is roughly $\mathcal{O}(k^3)$. As the vectors are chosen uniformly and independently by $\mathcal{O}_x$, the probability that $k$ vectors returned by this latter are linearly independent is equal to

$$\prod_{i=1}^{k-1}(1 - \frac{1}{2^i}).$$

According to Euler's Pentagonal number theorem, this probability tends to the number $({}^1\!/\!_2)_\infty \simeq 0.2887$ when $k$ tends to infinity. From this, it results that the attack consisting of querying $k$ times the oracle $\mathcal{O}_x$ and solving the resulting linear system using Gaussian elimination succeeds with a probability bounded by $({}^1\!/\!_2)_\infty$ in time complexity $\mathcal{O}(k^3)$.

Suppose now that the oracle $\mathcal{O}_x$ adds noise to the bit it outputs. That is, instead of outputting $a \cdot x$, it may flip that bit and output $a \cdot x \oplus 1$. When this flipping occurs for every returned bit, the attack above can still be applied by adding an extra step. Namely, the algorithm has to flip all the bits that it obtained before using Gaussian elimination. The case where the decision to flip each bit is random is more difficult to deal with. A popular variant consists in flipping the answer according to a probability that follows a Bernoulli distribution.

From now on, we will consider the following variant conditioned by a parameter $\nu \in ]0, {}^1\!/\!_2[$: On each query, $\mathcal{O}_{x,\nu}$ picks a uniformly chosen $k$-bit vector $a$ outputs a pair of the form $(a, a \cdot x \oplus \eta)$, where $\eta$ is a bit chosen following a Bernoulli distribution of parameter $\nu$, denoted

$\mathsf{Ber}(\eta)$. In this particular case, the answers of the oracle $\mathcal{O}_{x,\nu}$ follow the probability distribution

$$\{(a, a \cdot x \oplus \epsilon) : a \in_R \{0,1\}^k, \Pr[\epsilon = 1] = \nu\}.$$

This problem has proven to be very hard to solve and lies in the $\mathcal{NP}$-hard class of complexity as it will be detailled later. Although it has many formulations, the problem, as it is stated above, is commonly known as the *Learning Parity with Noise* problem, abbreviated LPN [Hås97].

**Definition 3.1 (The LPN problem)**
*Let $x$ be a binary vector of length $k$ and $\eta \in ]0, 1/2[$ a real number. We define $\mathcal{O}_{x,\eta}$ to be an oracle that outputs independent samples according to the distribution*

$$\{(a, a \cdot x \oplus \epsilon) : a \in_R \{0,1\}^k, \Pr[\epsilon = 1] = \nu\}.$$

*We say that an algorithm $\mathcal{A}$ solves the LPN problem with parameters $(k, \eta)$ with probability $\rho$ if*

$$\Pr[x \leftarrow \mathcal{A}^{\mathcal{O}_{x,\nu}}(1^k) | x \in_R \{0,1\}^k] \geq \rho.$$

*Here, the probability is taken over the random choice of $x$ and the random tape of $\mathcal{A}$.*

The LPN problem can be equivalently reformulated as a pure computational instance in which the problem is to find an assignment for a $q$-bit vector $x$ in a system of $q$ linear equations that satisfy $q_0 \leq q$ equations. From this perspective, the problem is best known as the minimum disagreement problem, or its abbreviation MDP problem [CKS03].

**Definition 3.2 (The MDP Problem)**
*Let $q$ and $k$ two positive integers, $A$ an $q \times k$ binary matrix, and $z$ a binary vector of length $q$. If $q_0$ denotes a positive integer smaller than or equal $q$, find a $k$-bit vector $x$ satisfying $q_0$ equations of the system $A \cdot x = z$.*

### 3.1.2 *The Average Hardness of the LPN Problem*

As a special case of a general decoding problem for linear codes, the NP-hardness of the LPN problem follows from the work of Berlekamp, McEliece, and van Tilborg [BMT78]. In short, they reduced the general decoding problem for linear codes to the three-dimensional matching problem, the 17th NP-Complete problem in Karp's list [Kar72]. A stronger result was found by Håstad [Hås01]: He proved that it is NP-hard to find an algorithm that succeeds in finding solutions to the general decoding problem for linear codes better than the trivial algorithm which tests random values.

Table 3.1: Complexity of Solving the LPN problem for different values of $\eta$ and $k$. Values taken from Leveil's thesis [Lev08].

|  | $k$ |  |  |  |
|---|---|---|---|---|
| $\eta$ | 128 | 256 | 512 | 768 |
| 0.1 | $2^{19}$ | $2^{38}$ | $2^{72}$ | $2^{97}$ |
| 0.125 | $2^{24}$ | $2^{43}$ | $2^{73}$ | $2^{105}$ |
| 0.25 | $2^{32}$ | $2^{51}$ | $2^{85}$ | $2^{121}$ |
| 0.4 | $2^{40}$ | $2^{62}$ | $2^{101}$ | $2^{143}$ |

However, NP-Completness only considers the worst-case hardness of solving decisional problems and does not guarantee that a randomly chosen instance of the problem cannot be solved by a polynomial-time algorithm. Unfortunately, it is the latter property that is required in cryptography. Therefore, we need to consider the average-case complexity of solving the LPN problem. Of course, no proof regarding the average hardness of the LPN problem was found. (This would prove the existence of one-way function!) Still, some arguments acts in favor of its hardness. Among them, we mention Regev's result concerning the the self-reducibility of the problem with respect to $x$ [Reg05] : The complexity of solving the LPN problem is independent from the choice of the secret vector $x$. (This property is shared with the discrete logarithm problem.) Another result due to Kearns [Kea98] relates the LPN problem to a learning problem where the solver is restricted to "statistical queries". In this paper, Kearns demonstrated that the class of parity functions cannot be efficiently learned by statistical queries. As learning algorithms that comply to the restriction of "statistical queries" form the majority of learning problems, this result rules out a large class of learning algorithm that can be used to attack the LPN problem. We finally mention a surprising result concerning the hardness of solving the LPN problem when the adversary does not know $\eta$: Laird [Pfi88] showed a technique that allows to revert in polynomial-time to the case where the adversary is given $\eta$. Therefore, from a complexity classification point of view, both variants are equivalent.

**Algorithms to Solve the LPN Problem** To date, the best method for solving the LPN problem is by using the BKW algorithm, named after its authors Blum, Kalai, and Wasserman [BKW03]. From a high level point of view, this algorithm implements the following idea: by picking carefully a few well-chosen vectors in a quite large set of samples and computing the xor of these vectors, we can find basis vectors, i.e., vectors of Hamming weight equal to 1. The advantage of finding this vector is that it readily yields a bit of the LPN's secret vector when the number of errors introduced in the answers is even. Therefore, the algorithm relies on finding enough independent combinations of vectors equals to the same basis vector, and use a majority vote enables to recover the correct value of the bit at the same position of the vector's 1. Note that this vote can only be efficient when the number of vectors to sum is small as the error bias of the final equation becomes too small. For that the number of these vectors is set in practice to be equal to $2^4$ or $2^6$.

Compared to exhaustive search algorithms on the correct equations [CTIN08], or on the errors introduced in the equations [GMZZ08], which run in strict exponential time, the BKW algorithm has the advantage of running in (slight) sub-exponential time. However, Levieil and Fouque [LF06] noted that the BKW algorithm makes unnecessary queries to the LPN oracle and proposed to use a Walsh transform to reduce the number of these queries. Independently, Lyubashevsky [Lyu05] adapted the BKW algorithm to produce a strict polynomial number of requests to the LPN oracle at the cost of a slightly greater overall complexity.

Table 3.1, that was compiled by Leveil for his PhD thesis [Lev08] gives the best attack complexities from all the previously mentioned algorithms to attack the LPN problem with various parameters.

### 3.1.3 *Extensions of the LPN Problem*

Another branch of research was started by Regev [Reg05] from generalizing the LNP problem to the ring $\mathbf{Z}_p^\star$, for a prime $p$, and called the generalized problem the learning with error problem (LWE). It turned out that this problem enjoys tight relations with lattice reduction problems. On one hand, Regev showed that the decision version of LWE is hard assuming quantum hardness of the gap shortest vector problem GapSVP and the shortest independent vector problem SIVP. On the other hand, Peikert [Pei09] proved a similar result assuming only the classical hardness of an easier version of the GapSVP problem.

The LWE problem proved to be very useful in serving as the basis for secure public-key encryption under both chosen-plaintext [Reg05, PVW08] and chosen-ciphertext [PW08, Pei09] attacks, oblivious transfer [GPV08], identity-based encryption [CHKP10], leakage-resilient encryption [AGV09, ACPS09], and more.

More recently, Lyubashevsky, Peikert and Regev [LPR10] extended the LWE problem to the ring of integer polynomials modulo a cyclotomic, irreductible over the rationals, polynomial, and used its hardness to propose the first truly practical lattice-based public-key cryptosystem with an efficient security reduction.

Another variation of the LPN and the LWE problems, known as the subspace LPN and LWE problem has been introduced by Pietrzak [Pie10]. Among others, these problems served to construct a MAC from the LPN problem [KPC$^+$11].

## 3.2 Security Models for the HB Family

Before going into the description of HB-like protocols, we review the main security models for these protocols. All these protocols are symmetric-key based. That is, the prover and the verifier receive a key $K$ uniformly distributed over the set of all possible secret keys.

As many probabilistic protocols, protocols from the HB family admit a false rejection rate. That it, it is possible that a legitimate prover gets rejected by the verifier even if the instance went undisturbed. We shall refer to the probability of this event happening by $P_{\mathsf{FR}}$. Of course, for practical reasons, we will require this probability to be negligible.

Conversely, it is also possible that a trivial adversary who only produces randomly generated protocol messages succeeds in authenticating as the prover. We denote the probability of this event occurring by $P_{\mathsf{FR}}$. Again, for obvious security reasons, this probability has to be negligible in the security parameter.

For simplicity, we assume the most devastating attack in which the adversary's goal is to recover the shared key. For these adversaries, we differentiate multiple attack scenarios.

- **Passive Adversaries.** This is the commonly assumed weakest adversarial model. A passive adversary can only eavesdrop on communications between two parties. This is usually formalized by giving to the adversary the access to an oracle $\mathcal{O}_{\mathsf{T}}$ that returns honestly generated protocol transcripts.
- **The DET Model.** Better known as the active adversarial model, it assumes that the adversary is able to interact with the two parties independently. That is, the adversary is given a black-box access to one oracle implementing the prover's and the verifier's strategies with the secret key as input. A first INIT message specifies which party the adversary wants the oracle to simulate. Note that the adversary cannot concurrently launch two sessions with the oracle.
- **The MIM Model.** This model considers the most powerful type of adversaries. Attackers in this model are called man-in-the-middle for their ability to "sit" between the prover and the verifier and have complete control the communication channel. Concretely, a man-in-the-middle has the power to insert a message in the channel (as an active adversary would do), but can also modify any message sent by one of the parties.
- **The GRS-MIM Model.** For reasons that will be made clearer in Section 3.5, a restricted man-in-the-middle adversary in which the adversary can only modify messages going from the verifier to the prover.

Finally, we say that a scheme is secure in a certain model if every probabilistic polynomial-time adversary who belongs to the associated class of adversaries does not recover the key $K$ with a probability better than $P_{\mathsf{FA}} + \mathsf{negl}(k)$.

## 3.3 The HB Protocol

The first protocol based on the LPN problem is due to Hopper and Blum, who proposed the HB protocol in 2001 [HB01]. Contrarily to its descendants, the aim of the HB protocol is to reach extreme simplicity to be used by humans for authentication. Along with this imposed

| **Prover** | | **Verifier** |
|---|---|---|
| Secret: $x$ | | Secret: $x$ |
| | $\xleftarrow{\ a\ }$ | Choose $a \in_R \{0,1\}^k$ |
| Choose $\nu \sim \mathsf{Ber}(\eta)$ | | |
| Compute $z = a \cdot x \oplus \nu$ | $\xrightarrow{\ z\ }$ | |
| | | Accept if $a \cdot x = z$ |

**Figure 3.1:** One round of the HB protocol. The protocol consists of $r$ such rounds.

simplicity, introducing a human parties induced substantial limitations for the adversarial model because, as in SAS-based cryptography [Vau05b, PV06, LP08] that is also intended for humans, the existence of an authenticated channel, such as the voice of the participants is much easier materialize than for electronic devices.

The HB protocol assumes that a prover and a verifier share a $k$-bit secret vector $x$. The authentication procedure, depicted in Figure 3.1, consists of repeating $r$ times the following operation: the prover first picks a random $k$-bit vector $a$ and sends it the verifier. This latter picks a bit $\nu$ according to the bernoulli distribution of parameter $\eta$, i.e., $\Pr[\nu = 1] = \eta$ and computes the answer $z = a \cdot x \oplus \nu$ to be sent back to the prover. At last, the prover verifies whether the equality $z = a \cdot x$ holds. If, after the $r$ repetitions, the equality $z = a \cdot x$ was satisfied at least $t$ times, for a threshold $t \in [\eta r, {}^{r}\!/_{2}[$, then the verifier acknowledges the prover. Otherwise, authentication fails. Hence, a legitimate prover gets rejected if he introduced at least $t + 1$ errors in its answers. This event, known as false rejection, happens with probability

$$P_{\mathsf{FR}} = \sum_{i=t+1}^{k} \binom{r}{i} \eta^i (1-\eta)^{r-i}$$

On the another side, the probability that a random answer $z$ gets accepted by the verifier has to be low to guarantee security. This probability, called the false acceptance rate, is given by

$$P_{\mathsf{FA}} = 2^{-r} \sum_{i=0}^{t} \binom{r}{i}$$

In the original paper, Blum and Hopper proved that, as long as the LPN assumption holds, the HB protocol is secure against passive adversaries. The proof comes from the observation that a adversary has only access to the transcript of different protocol instances and get pairs of the form $(a, z = a \cdot x \oplus \nu)$. As pairs correspond exactly to the output of the $O_{x,\eta}$ oracle from the LPN problem, any adversary deducing information on the shared secret of HB can be used to deduce information on the LPN secret.

The formal security reduction runs as follows. Given a passive adversary $\mathcal{A}_{\mathsf{HB}}$ against the HB protocol, we construct an adversary $\mathcal{A}_{\mathsf{LPN}}$ against the LPN problem that succeeds with

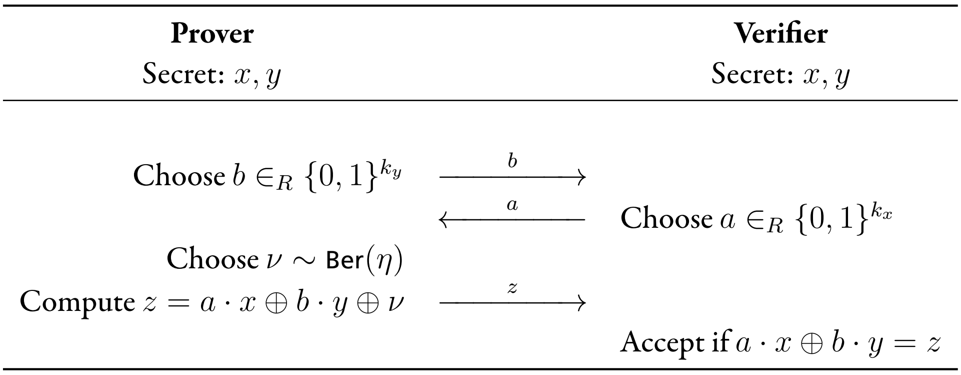
Prover
Secret: $x, y$
Verifier
Secret: $x, y$
Choose $b \in_R \{0,1\}^{k_y}$
$\xrightarrow{b}$
$\xleftarrow{a}$
Choose $a \in_R \{0,1\}^{k_x}$
Choose $\nu \sim \mathsf{Ber}(\eta)$
Compute $z = a \cdot x \oplus b \cdot y \oplus \nu$
$\xrightarrow{z}$
Accept if $a \cdot x \oplus b \cdot y = z$

**Figure 3.2:** One round of the HB$^+$ protocol. The protocol consists of $r$ such rounds.

the same probability. That is, $\mathcal{A}_{\mathsf{HB}}$ interacts with a prover and a verifier, relaying messages between the two and $\mathcal{A}_{\mathsf{LPN}}$ interacts with an oracle $\mathcal{O}_{x,\eta}$.

## 3.4 HB$^+$

Starting from the idea that RFID protocols, like human protocols, should be as simple as possible, Juels and Weis proposed to use the HB protocol as an RFID protocol [JW05a]. However, HB's security properties are insufficient in front of adversaries able to access RFID tags and perform the attack described in the end of the previous section. For this purpose, they proposed the HB$^+$ protocol whose goal was to design an HB-related protocol secure against active adversaries.

To thwart the attack against the HB protocol, Juels and Weis used a randomization technique in HB$^+$ consisting of an extra message added to each round of the protocol, sent by the prover at the beginning, and denoted by $b$. The shared secret between the prover and the verifier is then composed of two vectors $x$ and $y$ of size $k_x$ and $k_y$ respectively. Like HB, HB$^+$ consists of repeating $r$ times the following procedure: The prover first sends a uniformly chosen $k_y$-bit vector $b$ and sends it to the verifier. This latter also generates a random $k_x$-bit vector $a$ and sends it to the verifier. Then, after generating a bit $\nu \sim \mathsf{Ber}(\eta)$, the verifier computes $z = a \cdot x \oplus b \cdot y \oplus \nu$. Upon reception of $z$, the verifier checks the equality $z = a \cdot x \oplus b \cdot y$. In the end, the verifier authenticates the prover if a least $t$ authentication rounds succeeded, for $t \in [\eta r, {}^{r}\!/_{2}[$.

Not only HB$^+$ fulfills its purpose of denying the active attack against HB, but it is provably immune to attacks performed by active adversaries as that was demonstrated in the paper of Juels and Weis [JW05a]. However, their result only hold in the sequential case, i.e., when the adversary has to terminate a session before provoking another one. A later paper by Katz and Shin [KS06a] showed that the reduction holds when the adversary is allowed to launch parallel instances with the parties of the protocol if $\eta < {}^{1}\!/_{4}$. This result was further generalized for any $\eta < 1/2$ by Katz and Smith [KS06b]. It should be noted that in all those security proofs, $k_y$ and $\eta$ are only the parameters on which depend the resulted adversary against the LPN problem, the other parameters only affect the gap of the reduction.

## 3.5 The GRS Attack

Soon after its publication, Gilbert, Robshaw and Sibert exhibited an attack against HB$^+$ that can be carried by a man-in-the-middle adversary [GRS05]. Before going any further, it is worth noting that such an adversary is stronger than the active adversary considered in the security analysis of HB$^+$. So, the GRS attack does not contradict the security claims mentioned in [JW05a, KS06a, KS06b]. In the same time, the adversary considered by Gilbert et al. is also weaker than the usual man-in-the-middle adversary as, to carry out the attack, the adversary is only required to manipulate messages going from the verifier to the prover. For this reason, such adversaries are called GRS adversaries and a protocol secure against any GRS adversary is said to be secure in the GRS model.

The GRS attack against HB$^+$ consists in changing the $a$ value sent by the verifier. Taking a constant $k_x$-vector $\delta$, the adversary replaces each $a$ of the $r$ rounds of the protocol by $a \oplus \delta$. Upon reception of each of these message, the prover computes

$$z_\delta = a \cdot x \oplus \delta \cdot x \oplus b \cdot y \oplus \nu.$$

Hence, the verifier accepts when $\delta \cdot x \oplus \nu = 0$. In the case where $\delta \cdot x = 0$, the success probability is not altered and the prover authenticates successfully with probability $1 - P_{\mathsf{FR}}$. In the other case, i.e., when $\delta \cdot x = 1$, all the noise bits become flipped and the probability that the verifier acknowledges the prover is

$$\sum_{i=0}^{t} \binom{r}{i} \eta^{r-i}(1-\eta)^i = \sum_{j=r-t}^{r} \binom{r}{j} \eta^j (1-\eta)^{r-j} < P_{\mathsf{FR}}.$$

Depending on the outcome of the protocol, the adversary deduces one linear equation in $x$. That equation is correct with a probability of at least $(1 - P_{\mathsf{FR}})$. The secret vector $x$ can then be recovered by launching $k_x$ instances of the protocol and altering the $a$ messages with different $\delta_i$ to obtain $k_x$ linear equations and then solve the linear system. This way, the adversary is able to recover $x$ with probability $(1 - P_{\mathsf{FR}})^k \prod_{i=1}^{k-1}(1 - 2^{-i})$, that tends to $(1/2)_\infty$ when $k \to \infty$, in time complexity $\mathcal{O}(k_x^3)$, corresponding to the complexity of solving the linear system. Note that this attack can be optimized by carefully choosing the vectors $\delta_i$. For instance, the adversary can choose $\delta_i$ as being the vector with only the bit at position $i$ set to 1 and the others set to $0$. In this case, $\delta_i \cdot x = x_i$ so the adversary deduces this bit from the outcome of the protocol. The complexity of the attack becomes linear in $k_x$ and the success probability increases to $(1 - P_{\mathsf{FR}})^k$ that tends to $1$ when $k \to \infty$.

Once the adversary has obtained $x$, she can interact with the verifier, impersonating the prover and provoking the launch of protocol instances by sending the same vector $b$ during the $r$ rounds. In the last step of each round, the adversary sends $z = a \cdot x$. As the verifier checks that $z = a \cdot x \oplus b \cdot y$, the adversary deduces that $b \cdot y = 0$ when the verifier accepts. In contrary, she deduces that $b \cdot y = 1$ when the verification fails. As before, this deduction is wrong with probability $(1 - P_{\mathsf{FR}})$. The adversary can then recover $y$ by the same method as $x$. Hence, she succeeds in time complexity $\mathcal{O}(k_y)$ and probability $(1 - p_{\mathsf{FR}})^k$. We remark that this second phase is not mandatory for all attack scenarios. If, for example, the goal of the adversary is to impersonate the prover to the verifier, then recovering $y$ is unnecessary: Once the adversary learns $x$, she only has to choose $b = 0$ so that the final response $z$ does not depend on $y$.

As Juels and Weis [JW05b] noted, it may be possible to counter this attack by introducing extra defense mechanisms. For instance, the verifiers can have a detection mechanism that consists of stopping the system from performing authentications or revoking the shared key after a fixed number of authentication failures is reached. As the manipulations of the adversary induce an acceptance rate of $1/2$ for the $k_x$ protocol sessions needed to recover $x$, setting a threshold lower than $k_x/2$ for the number of failed authentications would limit the success probability of the attack.

## 3.6 Attempts to Thwart the GRS Attack

After the publication of the GRS attack on HB$^+$, several HB-related candidates were proposed to obtain a lightweight protocol based on the LPN problem secure in the GRS model.

In this section, we review a certain number of these protocols.

### 3.6.1 *HB$^{++}$*

HB$^{++}$ was proposed by Bringer, Chabanne, and Dottax [BCD06] to yield an HB-related protocol that is secure in the GRS model.

The protocol is depicted in Figure 3.3. It works in two phases. In the first phase, two parties who share a secret key $Z$ agree on a session key consisting of a quadruplet $(x, x', y, y')$. For that, both participants exchange $k$-bit vectors $A$ and $B$ and compute $(x, x', y, y') \leftarrow h(A, B, Z)$, where $h$ is a universal hash function. The second phase of the protocol is similar to HB$^+$, i.e., the two parties exchange $k$-bit vectors $a$ and $b$. Then the prover picks two bit noises $\nu, \nu'$ following the bernoulli distribution with parameter $\eta$. He then sends to the verifier $z = a \cdot x \oplus b \cdot y \oplus \nu$, as in HB$^+$, and $z' = f(a)^{\ll i} \cdot x' \oplus f(b)^{\ll i} \cdot y' \oplus \nu'$. The verifier then accepts the session of both relations hold. As for HB$^+$, both participants run the protocol for $r$ rounds and authentication succeeds if more than $t$ sessions get accepted. HB$^{++}$ can

| **Prover**<br>Secret: $Z$ | | **Verifier**<br>Secret: $Z$ |
|---|---|---|
| | **Session Key Derivation** | |
| Choose $B \in_R \{0,1\}^k$ | $\xrightarrow{B}$ | |
| | $\xleftarrow{A}$ | Choose $A \in_R \{0,1\}^k$ |
| $(x, x', y, y') \leftarrow h(A, B, Z)$ | | $(x, x', y, y') \leftarrow h(A, B, Z)$ |
| | $i^{\text{th}}$ **Authentication Round** | |
| Choose $b \in_R \{0,1\}^k$ | $\xrightarrow{b}$ | |
| Choose $\nu, \nu' \sim \mathsf{Ber}(\eta)$ | $\xleftarrow{a}$ | Choose $a \in_R \{0,1\}^k$ |
| Compute $z = a \cdot x \oplus b \cdot y \oplus \nu$ | | |
| $z' = f(a)^{\lll i} \cdot x' \oplus f(b)^{\lll i} \cdot y' \oplus \nu'$ | $\xrightarrow{z,z'}$ | Accept if<br>$a \cdot x \oplus b \cdot y = z$ and<br>$f(a)^{\lll i} \cdot x' \oplus f(b)^{\lll i} \cdot y' = z'$ |

Figure 3.3: The HB$^{++}$ protocol. One complete protocol instance consists of one session key derivation protocol and $r$ authentications. such rounds. $f$ is a permutation and $f(\cdot)^{\lll i}$ refers to the bit rotation by $i$ bits to the left in little endian notations.

be seen as two parallel instances of the HB protocol with independent secrets but correlated challenges.

Concerning its security in the GRS model, the authors of HB$^+$ gave arguments regarding adversaries who modify the $a$ message of all rounds of a protocol instance. Despite this, Gilbert, Robshaw, and Seurin [GRS08a] proposed an adversary who only disturbed the first $s$ rounds by adding a constant vector $\delta$ to each vector $a$ in a way similar to the GRS attack against HB$^+$. Their subsequent analysis proved that this attack succeeds in producing a linear equation in the session secrets $x$ and $x'$ if the attacker disturbs $s \in [\![\frac{t-\eta r}{1-2\eta}, 2\frac{t-\eta r}{1-2\eta}]\!]$ rounds. As for the sake of correctness, false rejections have to happen with small probability, it must be that $t - \eta r$ is large so that we can find values of $s$ in the interval. Therefore, HB$^{++}$ is insecure when no session key is established, i.e., if $(x, x', y, y')$ is the long term key. Nevertheless, Gilbert, Robshaw, and Seurin were able to extend the attack to the case in which session keys are used.

### 3.6.2 *HB*$^\star$

HB$^\star$ was proposed by Duc and Kim [DK07]. Again, this protocol consists of running $r$ rounds and authentication succeeds when at least $t$ of these rounds succeed. However, the protocol differs for HB$^+$ in that participants are given an extra $k$-bit secret vector $s$ that is used in the beginning of the protocol to securely send a bit $\gamma$. This is done by having the

| **Prover** | | **Verifier** |
|---|---|---|
| Secret: $x, y, s$ | | Secret: $x, y, s$ |
| Choose $b \in_R \{0,1\}^k$ | | |
| $\gamma \sim \mathsf{Ber}_{\nu'}$ | | |
| $w = b \cdot y \oplus \gamma$ | $\xrightarrow{b}$ | |
| | $\xleftarrow{a}$ | Choose $a \in_R \{0,1\}^k$ |
| Choose $\nu \sim \mathsf{Ber}(\eta)$ | | |
| if $\gamma = 0$ then | | |
| $z = a \cdot x \oplus b \cdot y \oplus \nu$ | | |
| else | | |
| $z = b \cdot x \oplus a \cdot y \oplus \nu$ | $\xrightarrow{z}$ | if $w = b \cdot s$ then |
| | | Accept if $z = a \cdot x \oplus b \cdot y \oplus \nu$ |
| | | else |
| | | $z = b \cdot x \oplus a \cdot y \oplus \nu$ |

**Figure 3.4:** One round of the HB$^\star$ protocol. The protocol consists of $r$ such rounds.

prover generate a random bit $\gamma$ following a Bernoulli distribution of parameter $\nu'$ along with a $k$-bit vector $b$ and sending $b$ with $w = b \cdot s \oplus \gamma$. With the knowledge of $s$, the verifier can easily recover $\gamma$. After that, the verifier generates its $k$-bit challenge $a$ and send to the prover. Depending on $\gamma$, the latter either computes $z$ as $a \cdot x \oplus b \cdot y \oplus \nu$ or as $z = a \cdot y \oplus b \cdot x \oplus \nu$. The verifier then verifies whether the $z$ he receives is consistent using the appropriate equation.

Regarding its security, Duc and Kim provided a heuristic analysis. Concretely, they argued that since no adversary can recover $\gamma$ (This leads to a direct attack against the LPN problem), the GRS attack does not apply as the adversary cannot know whether it is adding $\delta \cdot x$ or $\delta \cdot y$ to $z$. However, Gilbert, Robshaw, and Seurin [GRS08a] demonstrated that the success probability of the GRS manipulation applied to HB$^\star$ is dependent on $x$ and $y$. In short, every value for the pair $(\delta \cdot x, \delta \cdot y)$ induce a different success probability which can be easily computed. Hence, it is sufficient to run the same attack several times to deduce the success probability of the protocol with the manipulation, and therefore deducing the value of $\delta \cdot x$ and $\delta \cdot y$. Repeating the whole attack $k$ times, the adversary obtains two well defined systems of linear equations which she can solve to retrieve $x$ and $y$.

### 3.6.3 *PUF-HB*

Hammouri and Sunar [HS08] proposed to combine HB$^+$ and physically unclonable functions (PUF) to obtain a tamper-resilient authentication protocol secure against GRS-like attacks.

| **Prover** | | **Verifier** |
|---|---|---|
| Secret: $y$ | | Secret: $y$ |
| Choose $b \in_R \{0,1\}^{k_y}$ | $\xrightarrow{b}$ | |
| | $\xleftarrow{a}$ | Choose $a \in_R \{0,1\}^{k_x}$ |
| Choose $\nu \sim \mathsf{Ber}(\eta)$ | | |
| Compute $z = \mathsf{PUF}(a) \oplus b \cdot y \oplus \nu$ | $\xrightarrow{z}$ | |
| | | Accept if $\mathsf{PUF}(a) \oplus b \cdot y = z$ |

**Figure 3.5:** One round of the PUF-HB protocol. The protocol consists of $r$ such rounds.

A PUF is a function that is embodied in a physical structure and is easy to evaluate but hard to predict [GCvDD03]. Moreover, an individual PUF device must be easy to make but practically impossible to duplicate, even given the exact manufacturing process that produced it. Nevertheless for some types of PUFs, it is possible to obtain a good approximation of the PUF, up to an error rate of 3% for delay-based PUFs [GCvDD03], which in the case of an HB-like protocol can incorporated in the LPN problem's Bernoulli noise.

PUF-HB is very similar to HB$^+$ and differs in the use of the PUF to compute the prover's answer. As it is shown in Figure 3.5, instead of computing $a \cdot x$ like in the HB$^+$ protocol, the prover computes $\mathsf{PUF}(a)$. In other words, the secret in PUF-HB consists of a single binary vector $y$. The prover initiates a protocol instance by sending a nonce $b$ to which the verifier replies with a challenge $a$. The prover's answer is then computed by picking a bit $\nu \sim \mathsf{Ber}(\eta)$ and $z = \mathsf{PUF}(a) \oplus b \cdot y \oplus \nu$. Having a function $\mathsf{PUF}_\epsilon(\cdot)$ approximating the $\mathsf{PUF}$, the verifier checks that the equality $z = \mathsf{PUF}_\epsilon(a) \oplus b \cdot y$ holds. If after the procedure is repeated $r$ times, the verifier has accepted at least $t$ rounds, the protocol succeeds.

Using the non-linearity of the PUF function, the authors of PUF-HB argued that the GRS attack does not apply to their protocol. They however do not show whether it can resist other variants of the GRS attack and only proved that PUF-HB is resistant to active adversaries, just like HB$^+$. As it was already pointed at in [OOV08] and[Seu09], PUF-HB can easily be shown to be vulnerable to man-in-the-middle attacks. That is, a man-in-the-middle adversary who can modify messages going from the prover to the verifier can change all $b$ messages to $b \oplus \delta$ and check whether the authentication succeeds or not. When authentication succeeds, the attacker learns that $\delta \cdot y = 0$ with probability $1 - P_{\mathsf{FR}}$. Otherwise, she deduces that $\delta \cdot y = 1$, which holds with probability $1 - P_{\mathsf{FA}}$. Running this attack $k$ times yields enough equations to solve a linear system and recover $y$.

Having said that, we can even show that PUF-HB is not secure in the GRS model for most PUFs. We only cover the case of delay-based PUFs but the attack generalizes to other types of PUFs that are vulnerable to modeling attacks [RSS$^+$10]. (In fact, the PUFs considered in the latter paper cover most proposed PUFs.) Delay-based PUFs can be modeled by a linear equation of the form.

$$\mathsf{PUF}(a) = \begin{cases} 1 & \text{when } \sum_{i=1}^{k}(-1)^{a_i}y_i + y_{k+1} > 0 \\ b \in_R \{0,1\} & \text{when } \sum_{i=1}^{k}(-1)^{a_i}y_i + y_{k+1} = 0 \\ 0 & \text{when } \sum_{i=1}^{k}(-1)^{a_i}y_i + y_{k+1} < 0 \end{cases}$$

We remark that the

The first HB-related protocol that proved to be immune in the GRS model is HB# [GRS08b] for which we dedicate the next Chapter.

# 4

# HB# AND ITS (IN)SECURITY AGAINST MAN-IN-THE-MIDDLE ATTACKS

Contents

As we have seen in the previous chapter, numerous attempts to propose an alternative to $\text{HB}^+$ secure against GRS-like attacks have been made. It is was not until three years after the proposal of Juels and Weis, in 2009, that Gilbert, Robshaw, and Seurin proposed Random-HB# as the first HB-like protocol with a formal proof of security in the GRS model. Moreover, they conjectured that Random-HB# is secure against general man-in-the-middle adversaries. However, the cost of such security is the size of the secret bits needed for the random secret matrices that replaced the vectors of $\text{HB}^+$. For this sake, a variant, which has less memory requirements, called HB#, was also proposed in the same paper.

In this chapter, we first describe Random-HB# and HB# and then present their security properties. After that, we challenge the conjecture of Gilbert, Robshaw and Seurin about the security of Random-HB# and HB# against man-in-the-middle attacks. Concretely, we devise a strategy that allows a man-in-the-middle adversary that is given the result of all protocol instances to recover the secret shared between a prover and a verifier. Depending on the parameter set considered, our strategy succeeds with the adversary provoking either $2^{20}$ or $2^{35}$ authentications. We further bound the minimal size required for Random-HB# and HB# to prevent our attack and show that it needs to exceed $5\,000$ bits in any case and $15\,000$ bits if we take into account the necessity of having legitimate tags authenticated with probability close to $1$.

Moreover, we look at possible fixes to Random-HB# and HB#, including lowering the acceptance threshold or excluding the possibility of having false negatives. Unfortunately, we demonstrate that these also vulnerable to variants of our attack.

The results presented in this chapter are part of a paper published at AsiaCrypt 2008 [OOV08].

## 4.1 Random-HB# and HB#

### 4.1.1 *Description*

In Random-HB#, a prover and a verifier are assumed to share two secret matrices $X$ and $Y$ of dimension $k_x \times m$ and $k_y \times m$ respectively. These matrices are assumed to be randomly picked from the set of matrices with binary components, hence the prefix Random.

As shown in Figure 4.1, Random-HB# runs in one round as follows: At first, the prover generates a $k_y$-bit vector $b$ and sends it to the verifier. This latter then generates a random $k_x$-bit vector $a$ that he sends back to the prover. Upon reception of $a$, the prover picks an $m$-bit vector $\nu$ following the binomial distribution with parameters $(m, \eta)$, i.e., every bit of $\nu$ is set to 1 with probability $\eta$, independently from the other bits. After that, he computes $z = a \cdot X \oplus b \cdot Y \oplus \nu$, and transmits it to the verifier. At last, the verifier compares the $z$ he receives with $aX \oplus bY$ and accepts the prover if they differ in $t$ positions at most, i.e., he accepts if $\mathsf{wt}(aX \oplus bY \oplus z) \leq t$.

| **Prover** | | **Verifier** |
|---|---|---|
| Secret: $X, Y$ | | Secret: $X, Y$ |
| Choose $b \in_R \{0,1\}^{k_y}$ | $\xrightarrow{\quad b \quad}$ | |
| | $\xleftarrow{\quad a \quad}$ | Choose $a \in_R \{0,1\}^{k_x}$ |
| Choose $\nu \sim \mathsf{Binom}(m, \eta)$ | | |
| Compute $z = aX \oplus bY \oplus \nu$ | $\xrightarrow{\quad z \quad}$ | |
| | | Accept if |
| | | $\mathsf{wt}(aX \oplus bY \oplus z) \leq t$ |

**Figure 4.1:** The Random-HB# and HB# protocols. In Random-HB#, $X$ and $Y$ are random matrices, in HB# they are Toeplitz matrices. wt denotes the Hamming weight.

We remark that Random-HB# can be seen as the compression of $m$ rounds of $\text{HB}^+$ with the same challenge pair $(b, a)$ and different secret vectors, represented by the columns of the matrices $X$ and $Y$. The size of secret is then increased from $k_x + k_y$ for $\text{HB}^+$ to $(k_x + k_y)m$ for Random-HB# and the same bounds for the probabilities of false accepts and false rejects can be derived by replacing $r$, the number of authentication rounds in $\text{HB}^+$ by $m$, the number of rows in the matrices $X$ and $Y$

$$P_{\mathsf{FA}} = 2^{-m} \sum_{i=0}^{t} \binom{m}{i}, \qquad P_{\mathsf{FR}} = \sum_{i=t+1}^{k} \binom{m}{i} \eta^i (1-\eta)^{m-i}.$$

While Random-HB# possess strong security guarantees, as this will be detailled in the next section, it needs a huge amount of memory to store all the secret bits of the secret matrices $X$ and $Y$. Such an amplification is not acceptable when dealing with constrained devices such as RFIDs or humans. To obtain a practical protocol, the authors of Random-HB# proposed a practical variant to Random-HB# in which the random secret matrices $X$ and $Y$ have a special structure: all values on the same diagonal are equal. These matrices are known as Toeplitz matrices.

**Definition 4.1 (Toeplitz Matrix)**
*An $k \times m$ Toeplitz matrix $X$ is a matrix in which all values on the same diagonal are equal, i.e., it has the form:*

$$A = \begin{pmatrix} a_1 & a_2 & a_3 & \cdots & \cdots & a_m \\ a_{m+1} & a_1 & a_2 & \ddots & \ddots & \vdots \\ a_{m+2} & a_{m+1} & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & a_2 & a_3 \\ \vdots & \ddots & \ddots & a_{m+1} & a_1 & a_2 \\ a_{m+k-1} & \cdots & \cdots & a_{m+2} & a_{m+1} & a_1 \end{pmatrix}$$

**Table 4.1:** Practical parameter sets for HB# matching 80-bit security against GRS adversaries. In the set III, the Hamming weight of the error vector $\nu$ generated by the prover is always smaller than or equal $t$.

| Parameter set | $k_x$ | $k_y$ | $m$ | $\eta$ | $t$ | $P_{\mathsf{FR}}$ | $P_{\mathsf{FA}}$ |
|---|---|---|---|---|---|---|---|
| I | $80$ | $512$ | $1164$ | $0.25$ | $405$ | $2^{-45}$ | $2^{-83}$ |
| II | $80$ | $512$ | $441$ | $0.125$ | $113$ | $2^{-45}$ | $2^{-83}$ |
| III | $80$ | $512$ | $256$ | $0.125$ | $48$ | $0$ | $2^{-81}$ |

*Such a matrix can be uniquely defined through a $(k + m - 1)$-bit vector $s$ that determines the bits on the first row and the first column of the matrix. Hence, we denote $A_s$ the Toeplitz matrix defined by the vector $s$.*

So, HB# is obtained by replacing the random matrices of Random-HB# $X$ and $Y$, by two Toeplitz matrices that we denote $X_x$ and $Y_y$. The main benefit from this trick is that number of secret bits needed for HB# becomes as low as $k_x+k_y+2m-2$, compared to the $(k_x+k_y)m$ bits needed for Random-HB#. Unfortunately, the price to pay for this optimization is that, as it will be shown in the next section, the computational hypothesis on which the security of HB# stands is not as well studied as the LPN problem is, and thus possibly weaker.

### 4.1.2 *Proposed Parameter Sets*

When defining parameter values for HB#, The first thing to consider is the targeted false acceptance and rejection rates. As the former only depends on $m$ and $t$, setting a negligible value for $P_{\mathsf{FA}}$ yields a relation involving those two parameters. Setting the rejection rate, a function of $m$, $t$, and $\eta$, yields another equation in these three parameters.

The overall security of the scheme then depends on $k_X$, $k_Y$ and $\eta$. However, the security proof of Random-HB# shows that $k_X$ and $k_Y$ play two different roles: only $k_Y$ is related to the difficulty of the underlying LPN problem, while $k_X$ needs only be $k$-bit long to achieve $k$-bit security.

In the end, two parameter sets, matching 80-bit security against GRS-like attacks, corresponding to the two most popular values for $\eta$ in the LPN problem,1/4 and 1/8, were proposed for HB# by Gilbert et al. [GRS08b]. A third parameter set was proposed for a variant of the protocol in which the prover never uses error vectors of Hamming weight greater than the threshold $t$. These parameters are shown in Table 4.1.

Please note that no parameter set was proposed for Random-HB# as it was meant to be a protocol of theoretical value only. However, throughout the analysis that follows in Sec 4.3, we use the parameter values intended for HB# in Random-HB# to provide numerical examples.

## 4.2 The Security of Random-HB# and HB# in the GRS Model

In this section, we give a sketch on the proof of security of Random-HB# and HB# in the GRS Model. We kindly refer the reader to the PhD thesis of Yannick Seurin [Seu09, Chapter 5] for a complete analysis.

### 4.2.1 *The MHB Puzzle*

To reduce the security of Random-HB# to the LPN problem, Gilbert et al. introduced an intermediate problem, in the form of a weakly-verifiable puzzle, whose hardness is equivalent to the hardness the LPN problem; this puzzle is called the MHB puzzle. Hence, we introduce the definition of weakly-verifiable puzzles, as they were defined by Canetti et al. [CHS05].

**Definition 4.2 (Weakly-Verifiable Puzzle, [CHS05])**
*A weakly-verifiable system puzzle is a pair of algorithms $(\mathcal{G}, \mathcal{V})$ defined as follow:*

- *$\mathcal{G}$, called the puzzle generator, is a probabilistic polynomial-time algorithm that, on input a security parameter $1^k$, generates a random "puzzle" $p$ and some "check information", i.e.,*

$$(p, c) \leftarrow \mathcal{G}(1^k).$$

- *$\mathcal{V}$, called the puzzle verifier, is a deterministic polynomial-time algorithm that on input a puzzle $p$, check-information $c$, and answer $a$, outputs a bit, i.e.,*

$$\{0, 1\} \leftarrow \mathcal{V}(p, c, a).$$

*A solver $\mathcal{S}$ for this puzzle is a polynomial-time algorithm, in $k$, that takes as input a puzzle $p$ generated by $\mathcal{G}$ and outputs an answer $a$. We define the wining probability of such a solver as the probability that the puzzle verifier $\mathcal{V}$, on input $p$, the check information $c$, and $a$, outputs $1$.*

*A system puzzle is said to be $(1-\epsilon)$-hard if for any efficient solver $\mathcal{S}$, we have*

$$\forall\, k \in \mathbf{N}: \qquad \left| \Pr\left[ \mathcal{V}(p, c, a) = 1 \,\middle|\, \begin{array}{c} (p, c) \leftarrow \mathcal{G}(1^k) \\ a \leftarrow \mathcal{S}(p) \end{array} \right] - \epsilon \right| = \mathsf{negl}(k)$$

*where the probability is taken over the random coins of $\mathcal{G}$ and $\mathcal{S}$.*

The LPN problem can be equivalently reformulated in terms as a puzzle, known as the HB puzzle. This puzzle is defined as a game in which the solver has access to a polynomially bounded number of samples $(a_i, z_i)$, that are identical to the pairs computed by an LPN Oracle $\mathcal{O}_{x,\eta}$. The goal of the solver if then to find one linear equation involving a random vector $a$, supplied by the puzzle generator, and the secret vector $x$.

**Definition 4.3 (The HB Puzzle)**
*Let $\eta \in ]0, 1/2[$ and let $q : \mathbf{N} \rightarrow \mathbf{N}$ be a polynomial function. The HB puzzle is defined as follows:*

- *The puzzle generator $\mathcal{G}$, which takes the security parameter $k$ as input, first picks a random $k$-bit vector $x$. It then generates $q(k)$ random binary vectors $a_1, \ldots, a_{q(k)}$ of length $k$ and $q(k)$ bits $\nu_1, \ldots, \nu_{q(k)}$ following the Bernoulli distribution with parameter $\eta$. After that, it computes, for $i = 1 \ldots q(k)$, $z_i = a_i \cdot x \oplus \nu_i$. At last, it generates a random $k$-bit vector $a$ and returns $\{(a_i, z_i)\}_{1 \leq i \leq q(k)}$ and $a$ that compose the puzzle. The check information is the vector $x$ while the answer of the puzzle, that the solver has to compute in order to win, is a bit denoted $z$.*
- *The puzzle verifier, $\mathcal{V}$, outputs $1$ if and only if the equality $a \cdot x = z$ holds.*

The security of the HB protocol in the passive settings can be expressed in terms of the HB puzzle as it is shown hereafter.

**Lemma 4.1 (LPN Problem ⇔ HB Puzzle [GRS08b])**
*If the LPN problem is hard, then, for every polynomial function $q : \mathbf{N} \to \mathbf{N}$, the HB puzzle is $(1 - 1/2)$-hard.*

As it was already noted, Random-HB# can be seen as the parallel repetition of $m$ instances of HB$^+$ with different secret vectors and the same pair. It becomes then natural to consider the extension of the HB puzzle to $m$ parallel instances. For this, Gilbert et al. used a result from the paper of Canetti et al. [CHS05, Lemma 1] relative to the hardness of a puzzle composed of $m$ independent instances of a $(1 - \epsilon)$-hard puzzle. Essentially, they showed that the puzzle composed of $m$ independent instances of the former puzzle is $(1 - \epsilon^m)$-hard with respect to solvers who have to solve *all* the $m$ instances. The extension of the HB puzzle, named the MHB puzzle, can be defined as follows:

**Definition 4.4 (The MHB Puzzle)**
*Let $\eta \in ]0, 1/2[$ and let $m, q : \mathbf{N} \to \mathbf{N}$ be two polynomial functions. The MHB puzzle is defined as follows:*

- *The puzzle generator $\mathcal{G}$, which takes the security parameter $k$ as input, first picks a random $(k \times m)$-binary matrix $X$. It then generates $q(k)$ random binary vectors $a_1, \ldots, a_{q(k)}$ of length $k$ and $q(k)$ binary vectors $\nu_1, \ldots, \nu_{q(k)}$ of length $m$ that follow the Binomial distribution with parameters $m$ and $\eta$. After that, it computes, for $i = 1 \ldots q(k)$, $z_i = a_i \cdot X \oplus \nu_i$. At last, it generates a random $k$-bit vector $a$ and returns $\{(a_i, z_i)\}_{1 \leq i \leq q(k)}$ and $a$ that compose the puzzle. The check information is the matrix $X$ while the answer of the puzzle, that the solver has to compute in order to win, is an $m$-bit vector denoted $z$.*
- *The puzzle verifier, $\mathcal{V}$, outputs $1$ if and only if the equality $a \cdot X_x = z$ holds.*

**Theorem 4.1 (Hardness of the MHB Puzzle)**
*If the LPN problem is hard, then the MHB puzzle with parameter $m$, that is polynomially bounded in the security parameter, is $(1 - 2^{-m})$-hard for every polynomial function $q(\cdot)$.*

To mirror the difference between Random-HB# and HB#, Gilbert et al. introduced a puzzle similar to the MHB one in which the random matrix $X$ of the MHB puzzle is replaced by a random Toeplitz matrix. That puzzle is called the Toeplitz-MHB puzzle.

**Definition 4.5 (The Toeplitz-MHB Puzzle)**
*Let $\eta \in ]0, 1/2[$ and let $m, q : \mathbf{N} \to \mathbf{N}$ be two polynomial functions. The Toeplitz-MHB puzzle is defined as follows:*

- *The puzzle generator $\mathcal{G}$, which takes the security parameter $k$ as input, first picks a random $(k+m-1)$-bit vector $x$ that uniquely characterise the $k \times m$ Toeplitz matrix $X_x$. It then generates $q(k)$ random binary vectors $a_1, \ldots, a_{q(k)}$ of length $k$ and $q(k)$ binary vectors $\nu_1, \ldots, \nu_{q(k)}$ of length $m$ that follow the Binomial distribution with parameters $m$ and $\eta$. After that, it computes, for $i = 1 \ldots q(k)$, $z_i = a_i \cdot X_x \oplus \nu_i$. At last, it generates a random $k$-bit vector $a$ and returns $\{(a_i, z_i)\}_{1 \leq i \leq q(k)}$ and $a$ that compose the puzzle. The check information is the matrix $X_x$ while the answer of the puzzle, that the solver has to compute in order to win, is an $m$-bit vector denoted $z$.*
- *The puzzle verifier, $\mathcal{V}$, outputs $1$ if and only if the equality $a \cdot X_x = z$ holds.*

Unfortunately, no result about the equivalence between the MHB and Toeplitz-MHB puzzles is known so far. At the same time, no separation result between the two instances has been demonstrated either. So, while the hardness of the MHB puzzle is tightly related to the one of the LPN problem, the hardness of the Toeplitz-MHB puzzle is an open conjecture.

### 4.2.2 *The Security Reduction*

The proof of security of Random-HB# presented by Gilbert et al. runs in two steps. The first step is to prove that Random-HB# is secure in the DET model, i.e., against active adversaries. This result is not surprising by itself, as it is an adaptation of the proof of Juels and Weis concerning the security of HB$^+$ in the DET model. As the reduction proves that any active adversary against Random-HB# winning with a non-negligible probability reduces to an efficient solver for the MHB puzzle, which hardness in its turn reduces to the LPN problem. Unfortunately, the security reduction of HB# in the DET model relies on the conjectured hardness of the Toeplitz-MHB puzzle, which is not known to be equivalent to the LPN problem. Hence, the security of Random-HB# in the GRS model is based on the assumption that the LPN problem is hard while the security of HB# in the DET model is only based on the conjectured hardness of the Toeplitz-MHB puzzle.

The second step of the proof is to reduce any GRS adversary against Random-HB# to an active adversary. The idea of the proof is to show that almost every time the adversary modifies the $a$ message of the protocol, the verifier refuses the authentication. The reason for that is that, by doing this manipulation, the adversary is introducing errors in the triplet. For carefully chosen $m$ and $t$, the resulting error vector computed by the verifier will have, with overwhelming probability, a Hamming weight greater than $t$. We remark that this result is also applicable to HB#. In consequence, HB# is only proven to be secure in the DET model if the Topelitz-MHB puzzle is hard. However, if it is secure in the DET model, then it is secure in the GRS model.

| **Prover** | **Man in the middle** | **Verifier** |
|---|---|---|
| Secret: $X, Y$ | Target triplet: $(\bar{a}, \bar{b}, \bar{z})$ | Secret: $X, Y$ |
| Choose $b \in_R \{0,1\}^{k_y}$ | $\xrightarrow{\hat{b}=b\oplus\bar{b}}$ | |
| | $\xleftarrow{\hat{a}=a\oplus\bar{a}}$ | Choose $a \in_R \{0,1\}^{k_x}$ |
| Choose $\nu \sim \mathsf{Binom}(m, \eta)$ | | |
| Compute $z = \hat{a}X \oplus bY \oplus \nu$ | $\xrightarrow{\hat{z}=z\oplus\bar{z}}$ | |
| | | Accept if<br>$\mathsf{wt}(aX \oplus \hat{b}Y \oplus \hat{z}) \leq t$ |

**Figure 4.2:** The Man-in-the-Middle Attack against Random-HB# and HB#. The goal of the man-in-the-middle is to learn the Hamming weight of the error vector introduced in the triplet $(\bar{a}, \bar{b}, \bar{z})$.

The full security analysis of Random-HB# and HB# can be found in the original paper of Gilbert et al. [GRS08b], its full version from the Cryptology Eprint Archive [GRS08c], or the PhD thesis of Seurin [Seu09] (in French).

## 4.3 The Insecurity of Random-HB# and HB# in the MIM Model

The previous section described how the security of Random-HB# can be reduced to the LPN problem when dealing with GRS adversaries, through the MHB puzzle. However, Gilbert et al. could not extend that result to general man-in-the-middle adversaries. Instead, they proposed an analysis in favor of the immunity of Random-HB# to such attack scenarios. Concretely, they studied the possible perturbation a man-in-the-middle adversary can perform on one protocol instance $(a, b, z)$. In a first time, they studied the information an adversary may get by replacing the last message $z$ by $z \oplus \zeta$. As the outcome of this disturbed instance only depends on the Hamming weight of $\nu \oplus \zeta$, it is independent from the secret matrices. Hence, the adversary does not learn any information from such an attack. The other case considered by Gilbert et al. is when the adversary modifies the three messages, i.e., the adversary replaces the messages $a$ by $a \oplus \alpha$, $b$ by $b \oplus \beta$, and $z$ by $z \oplus \zeta$. Due to the perfect balancing of the function $a \mapsto a \cdot X$, when $X$ is a random matrix, the error vector computed by the verifier at the end is $\nu' = \alpha X \oplus \beta Y \oplus \zeta$, which also follow a uniform distribution. The probability that authentication succeeds is then bounded by $P_{\mathsf{FA}}$, which should be negligible for secure parameters of Random-HB#. We note that this reasoning can also be applied to HB# using a earlier result due to Carter and Wegman [CW79] that proves that the mapping $a \mapsto aX$ is perfectly balanced when $X$ is a random Toeplitz matrix (in fact, Carter and Wegman showed that it is a family of universal hash functions). After suggesting that none of these two cases reveal any information to the adversary, they conjectured that Random-HB# is secure against man-in-the-middle attacks.

The rest of this chapter is dedicated to show that this conjecture does not hold and to analyze possible variants of Random-HB#. Concretely, we will propose a man-in-the-middle attack against Random-HB#. As it will detailled, this attack also applies to HB# since this latter is only a simplification of Random-HB# to make it more practical. For the sake of clarity, we present the attack in three steps.

### 4.3.1 *Step 1: Computing the Hamming Weight of the Error Vector*

**Algorithm 1** Approximating $\bar{w}$

```
Input: $\bar{a}, \bar{b}, \bar{z}, n$
Output: $P^{-1}\left(\frac{c}{n}\right)$, an approximation of $\bar{w}$ where $\bar{w} = \mathsf{wt}(\bar{a}X \oplus \bar{b}Y \oplus \bar{z})$
Processing:
  Initialize $c \leftarrow 0$
  for $i = 1 \dots n$ do
    During a protocol, set $\hat{a} = a \oplus \bar{a}$, $\hat{b} = b \oplus \bar{b}$ and $\hat{z} = z \oplus \bar{z}$
    if reader accepts then
      $c \leftarrow c + 1$
    end if
  end for
```

The core idea of the attack is to perturbate the protocol messages $(a, b, z)$ exchanged during an instance by adding a special triplet: the adversary picks a triplet $(\bar{a}, \bar{b}, \bar{z})$ obtained by eavesdropping on a previous exchange between the prover and the verifier. That is, once the triplet $(\bar{a}, \bar{b}, \bar{z})$ has been obtained, the adversary modifies the communication messages of every other protocol instance: When the prover sends his $b$ message, it is replaced by $\hat{b} = b \oplus \bar{b}$. Similarly, the adversary replaces the messages $a$ and $z$ by $\hat{a} = a \oplus \bar{a}$ and $\hat{z} = z \oplus \bar{z}$. This operation is depicted in Figure 4.2. The goal of that perturbation is to "superpose" the error vector that was embedded in $\bar{z}$, denoted $\bar{\nu}$, on the different error vector contained in the $z$ that is sent by the prover. After a number of repetitions, the adversary would be able to deduce the Hamming weight of $\bar{\nu}$. The following computation shows that at the verifier authenticates the prover if $\mathsf{wt}(\nu \oplus \bar{\nu}) \leq t$

$$\begin{aligned} aX \oplus \hat{b}Y \oplus \hat{z} &= aX \oplus (\bar{b} \oplus b)Y \oplus (\bar{z} \oplus z) \\ &= (\hat{a}X \oplus bY \oplus z) \oplus (\bar{a}X \oplus \bar{b}Y \oplus \bar{z}) \\ &= \nu \oplus \bar{\nu} \end{aligned}$$

Algorithm 1 is a procedure that can be used by the adversary to recover $\bar{w}$, the Hamming weight of $\bar{\nu}$. The justification of the correctness of this algorithm is provided hereafter.

By defining $\hat{\nu} = \nu \oplus \bar{\nu}$ and $\hat{w} = \mathsf{wt}(\hat{\nu})$, we can derive the probability $p$ that the bit of $\hat{\nu}$ at position $i$, denoted $\hat{v}_i$ is equal to $1$:

$$p = \Pr[\hat{\nu}_i = 1] = \begin{cases} \eta & \text{if } \bar{\nu}_i = 0 \\ 1-\eta & \text{if } \bar{\nu}_i = 1. \end{cases}$$

Hence, $m - \bar{w}$ bits of $\hat{\nu}$ follow a Bernoulli distribution of parameter $\eta$ and the other $\bar{w}$ bits follow a Bernoulli distribution of parameter $1 - \eta$. Due to the pairwise-independence of all the bits of $\nu$ and $\bar{\nu}$, the expected value and variance of $\hat{w}$ are respectively given by

$$\mu = (m - \bar{w})\eta + \bar{w}(1-\eta), \qquad \sigma^2 = m\eta(1-\eta).$$

We let $P$ be the function defined as $P(\bar{w}) = \Pr[\hat{w} \leq t]$. If $n$ denotes the number of perturbed rounds and $c$ the number of times the authentication succeeded in those $n$ rounds, then $P(\bar{w})$ tends to $c/n$ when $n$ tends to the infinity. Using the central limit theorem to approximate $P$ by the normal distribution function $\Phi$, we obtain

$$P(\bar{w}) \approx \Phi(u), \qquad u = \frac{t-\mu}{\sigma},$$

The random variable $c/n$ thus follows a normal distribution with expected value $P(\bar{w})$ and variance $\frac{1}{n}P(\bar{w})(1-P(\bar{w}))$. Furthermore, letting $P'(\bar{w}) = P(\bar{w}+1) - P(\bar{w})$ denote the discrete derivative of $P$ at the point $\bar{w}$, we derive the following approximation

$$\begin{aligned} P'(\bar{w}) &= P(\bar{w}+1) - P(\bar{w}) \\ &\approx -\frac{1-2\eta}{\sqrt{m\eta(1-\eta)}}\Phi'(u) \\ &= -\frac{1-2\eta}{\sqrt{m\eta(1-\eta)}} \times \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}. \end{aligned}$$

In order to yield a good approximation of $P(\bar{w})$ by $c/n$, it is sufficient to take

$$n = \frac{\theta^2}{r^2} R(\bar{w}), \qquad R(\bar{w}) = 2\frac{P(\bar{w})(1-P(\bar{w}))}{(P'(\bar{w}))^2},$$

so that

$$\begin{aligned} \Pr\left[\frac{c}{n} - P(\bar{w}) > r|P'(\bar{w})|\right] &= \Pr\left[\frac{c}{n} - P(\bar{w}) < -r|P'(\bar{w})|\right] \\ &= \Phi(-\theta\sqrt{2}) \\ &= \frac{1}{2}\mathsf{erfc}(\theta). \end{aligned}$$

In the previous equation, erfc denotes the complementary error function, i.e., the complement to 1 of the error function $\mathsf{erf}(\cdot)$, defined as follows

$$\begin{aligned}\mathsf{erfc}(\theta) &= 1 - \mathsf{erf}(\theta)\\ &= 1 - \frac{2}{\sqrt{\pi}}\int_0^\theta e^{-t^2}dt\\ &= \frac{2}{\sqrt{\pi}}\int_\theta^{+\infty} e^{-t^2}\,dt.\end{aligned}$$

We deduce that with $\theta$ high enough, i.e. such that it gives an $\mathsf{erfc}(\theta)$ small enough, $^c/_n$ yields a fair estimate of $P(\bar{w})$ with precision $\pm rP'(\bar{w})$. Finally, the adversary can recover the Hamming weight of the error vector introduced in $\bar{z}$ by computing

$$\bar{w} = P^{-1}\left(\frac{c}{n}\right) = \Phi^{-1}$$

This algorithm can be used in different situations, by adjusting its internal parameters:

- With the prior assumption that $\bar{w}$ is an integer close to some value $w_0$, we can call Algorithm 1 and $r = {}^1/_2$ to infer $\bar{w} = \lceil P^{-1}(\frac{c}{n})\rfloor$ with error probability $\mathsf{erfc}(\theta)$.
- If $\bar{w} \in \{w_0 - 1, w_0 + 1\}$, then we can choose $r = 1$ to infer $\bar{w}$ by the closest value to $P^{-1}(\frac{c}{n})$. The error probability is $\frac{1}{2}\mathsf{erfc}(\theta)$.
- With the prior assumption that $\bar{w} \in \{w_0 - 2, w_0, w_0 + 2\}$ we can use $r = 1$ to infer $\bar{w} = \lceil P^{-1}(\frac{c}{n})\rfloor$. The error probability is $\frac{1}{2}\mathsf{erfc}(\theta)$ when $\bar{w} \in \{w_0 - 2, w_0 + 2\}$ and it is $\mathsf{erfc}(\theta)$ when $\bar{w} = w_0$. If $\bar{w}$ comes with an a priori distribution of $((1-\eta)^2, 2\eta(1-\eta), \eta^2)$ over the support $\{w_0 - 2, w_0, w_0 + 2\}$, the error probability is $(\frac{1}{2} + \eta(1-\eta))\mathsf{erfc}(\theta)$.

In all cases, Algorithm 1 can be interpreted as an oracle running with complexity $n = \frac{\theta^2}{r^2}R(w_0)$ that can be used to compute $\bar{w}$ given $\bar{a}, \bar{b}, \bar{z}$, and succeeding with an probability of error smaller than $\mathsf{erfc}(\theta)$.

### 4.3.2 *Step 2: Using the Weight Oracle to Obtain Linear Equations*

Now that the adversary has an oracle to compute the Hamming weight of $\bar{\nu}$ inserted in a triplet $(\bar{a}, \bar{b}, \bar{z})$, we can use Algorithm 2 to recover, one by one, the values of the bits of $\bar{\nu}$. For this, we use a very simple strategy: at first, we recover the Hamming weight of $\bar{\nu}$ with the assumption that is close to $w_0 = \lfloor m\eta\rceil$, its mean value. Then, we iterate on every bit position $i$ of $\bar{z}$ and flip it to obtain $\bar{z}_i$. After that, submit the triplet $(\bar{a}, \bar{b}, \bar{z}_i)$ to the weight oracle, with the knowledge that the Hamming weight of the error vector introduced in $(\bar{a}, \bar{b}, \bar{z}_i)$ is $w_0 \pm 1$. If the weight measured by the oracle is $w_0 - 1$, then we deduce that, by flipping the bit at position, we have removed an error from $\bar{z}$. In the other case, i.e., when the returned value is

**Algorithm 2** Getting linear equations for $X$ and $Y$

```
Input: $\bar{a}, \bar{b}, \bar{z}$ and $\bar{w}_0$ the expected weight of $\bar{\nu} = \bar{a}X \oplus \bar{b}Y \oplus \bar{z}$
Output: A system of $m$ linear equations $\bar{a}X \oplus \bar{b}Y = \bar{c}$
Processing:
  Initialize $m$-bit vector $\bar{c} \leftarrow \bar{z}$
  Call Algorithm 1 on input $(\bar{a}, \bar{b}, \bar{z}, n = 4\theta^2 R(\bar{w}_0))$ to get $\bar{w}$
  for $i = 1 \ldots m$ do
      Flip bit $i$ of $\bar{z}$ to get $\bar{z}'$
      Call Algorithm 1 on input $(\bar{a}, \bar{b}, \bar{z}', n = \theta^2 R(\bar{w}))$ to get $\bar{w}'$
      if $\bar{w}' = \bar{w} - 1$ then
          $\bar{c}_i \leftarrow \bar{c}_i \oplus 1$
      end if
  end for
```

$w_0 + 1$, then no error bit was introduced at position $i$. This way, the adversary can recover all the error positions in $\bar{z}$. In the end, she obtains a correct system of $m$ linear equations of the form $\bar{a}X \oplus \bar{b}Y = \bar{z} \oplus \bar{\nu}$. As the complexity of Algorithm 1 is equal to its input $n$, when $\bar{w}_0$ is the initial guess for the value of $\bar{w}$, the Hamming weight of $\bar{\nu}$, the total complexity of Algorithm 2 is

$$4\theta^2 R(\bar{w}_0) + m\theta^2 R(\bar{w}).$$

Note that this operation can be repeated until the adversary gets enough equations to solve a linear system and recover $X$ and $Y$. Concretely, in order to recover the $\ell$ bits of the secret key, $\ell$ linear equations are necessary. Hence, we need to iterate Algorithm 2 $\lceil \ell/m \rceil$ times on independent $(\bar{a}, \bar{b})$ pairs. Recalling that Algorithm 2 outputs an erroneous equation with probability $\mathsf{erfc}(\theta)$, the expected number of errors in the equation system defining $X$ and $Y$ is then bounded by $\ell \cdot \mathsf{erfc}(\theta)$. In order to be solvable, the system of equations should consist of linearly independent vectors $(\bar{a}, \bar{b})$. For this sake, each time the adversary gets a triplet $(\bar{a}, \bar{b}, \bar{z})$, she verifies if the pair $(\bar{a}, \bar{b})$ is linearly dependent from the $i$ previous ones and dismisses it if this is the case. Recalling that these vectors are of bit-size $k_x$ and $k_y$ respectively, the probability that this event happens is $2^{i-k_x-k_y-1}$. We can then derive the number of sessions on which the adversary has to eavesdrop to get enough equations for the linear system

$$C = \sum_{i=1}^{\lceil \ell/m \rceil} \frac{1}{1 - 2^{i-k_x-k_y-1}} < 2 + \left\lceil \frac{\ell}{m} \right\rceil .$$

### 4.3.3 *Step 3: Solving the Linear System*

Now that the adversary has obtained enough linear equations, it remains to solve the linear system and compute the secret bits of $X$ and $Y$. For this aim, she can use classical algorithms from linear algebra such as Gauss and Gauss-Jordan elimination, LU decomposition, or the square root method which all run in time complexity $\Theta(\ell^3)$. To speed up this phase, other modern methods, asymptotically faster, can be used such as Strassen's formula for matrix multiplication and inversion [Str69], that runs in time complexity $\Theta(\ell^{\log_2 7}) \approx \Theta(\ell^{2.8})$ or the more complex Coppersmith-Winograd algorithm [CW90] which achieve the record complexity of $\Theta(\ell^{2.376})$. However, we note that this last method is only asymptotically faster as it is outperformed by Strassen's formula for our values of interest of $\ell$.

### 4.3.4 *Asymptotic Complexity Analysis*

The complexity of the attack is related to the complexity of Algorithm 2 (with a factor of $\ell C/m$), which, in its turn, is related to the complexity of Algorithm 1 (with a factor of $m+1$). Thus, the main component of the attack affecting the overall complexity is the input $n$ of Algorithm 1.

Recalling that $P(\bar{w}) \in [0,1]$, we have

$$\begin{aligned} n &= \frac{2\theta^2}{r^2} \times \frac{P(\bar{w})(1-P(\bar{w}))}{(P'(\bar{w}))^2} \\ &= \frac{2\theta^2}{r^2} \times P(\bar{w})(1-P(\bar{w})) \left( \frac{\sqrt{2\pi m\eta(1-\eta)}}{1-2\eta} e^{\frac{u^2}{2}} \right)^2 \\ &\in \Theta\left(\theta^2 e^{u^2}\right), \end{aligned}$$

and the minimal value of $n$ is reached when $u = 0$ which happens when

$$\bar{w}_0 = \bar{w}_{\mathsf{opt}} = \frac{t - m\eta}{1-2\eta},$$

and we obtain

$$\begin{aligned} P(\bar{w}_{\mathsf{opt}}) &= \frac{1}{2}, \qquad P'(\bar{w}_{\mathsf{opt}}) = -\frac{1-2\eta}{\sqrt{2\pi m\eta(1-\eta)}}, \\ R(\bar{w}_{\mathsf{opt}}) &= \frac{\pi m}{4}\left(\frac{1}{(1-2\eta)^2} - 1\right) = R_{\mathsf{opt}}. \end{aligned}$$

In order to obtain the minimal complexity, we have to start from a valid triplet $(\bar{a}, \bar{b}, \bar{z})$ obtained from a passive attack (or just by impersonating the verifier to the prover) for which $\bar{w} = \bar{w}_{\mathsf{opt}}$. As it is unlikely that the expected value of $\bar{w}_0$ is equal to $\bar{w}_{\mathsf{opt}}$, we would like to manipulate errors in $\bar{z}$ to reach an expected value of $\bar{w}_{\mathsf{opt}}$. Unfortunately, due to the hardness of the LPN problem, we cannot *remove* errors from $\bar{z}$ if $\bar{w} > \bar{w}_{\mathsf{opt}}$. However, if $\bar{w} \leq \bar{w}_{\mathsf{opt}}$ then we can *inject* errors in $\bar{z}$ so that the resulting vector has an expected weight of $\bar{w}_{\mathsf{opt}}$. When the triplet $(\bar{a}, \bar{b}, \bar{z})$ is known to be valid, in the sense that a session with that transcript was accepted by the verifier, and the false rejection rate of the protocol is negligible, we can use the approximation $\bar{w}_0 \approx m\eta$. In such a case, we can derive

$$m\eta \leq \frac{t - m\eta}{1 - 2\eta} \Longrightarrow t \geq 2m\eta(1 - \eta),$$

As the error probability of the attack should be less than $1$, $\mathsf{erfc}(\theta)$ should be less than the inverse of the number of secret bits $\ell$. Using the approximation $\Phi(-x) \approx \varphi(x)/x$ when $x$ is large (so $\Phi(-x)$ is small), we can set $\theta = \sqrt{\ln \ell}$ for which we obtain

$$\mathsf{erfc}(\theta) = 2\Phi(-\theta\sqrt{2}) \approx 2\frac{\varphi(\theta\sqrt{2})}{\theta\sqrt{2}} = \frac{e^{-\theta^2}}{\theta\sqrt{\pi}} < \frac{1}{\ell}.$$

Thus, from the expression of $n$ given above, we distinguish three cases:

1. $t \geq 2m\eta(1 - \eta)$: as we have $\bar{w} = \bar{w}_{\mathsf{opt}}$ for which $u = 0$, the attack has an asymptotic complexity of $\Theta\left(\ell \ln \ell\right)$.
2. $t = 2m\eta(1 - \eta) - c\sqrt{m\eta(1 - \eta)}$ for $c = \Theta\left(\sqrt{\ln m\eta(1 - \eta)}\right)$: the complexity is multiplied by a $e^{c^2}$ factor. Thus, Algorithm 1 still runs in polynomial time.
3. In the other cases, the complexity varies from sub-exponential to exponential but is clearly not polynomial anymore.

**Strategy for the case $t \geq 2m\eta(1 - \eta)$.** From the hypothesis $t \geq 2m\eta(1 - \eta)$, we have that $\bar{w}_{\mathsf{opt}} \geq \bar{w} = m\eta$. Thus, the best strategy is to optimize the complexity of Algorithm 1 by having a triplet $(\bar{a}, \bar{b}, \bar{z})$ with an error vector of expected Hamming weight $\bar{w}_{\mathsf{opt}}$.

Our strategy is to use a triplet $(\bar{a}, \bar{b}, \bar{z})$ obtained from a passive attack, and introduce some errors in it. That is, we flip any $\lfloor(\bar{w}_{\mathsf{opt}} - m\eta)/(1 - 2\eta)\rceil$ bits of $\bar{z}$ to get $\bar{\nu}$ of expected Hamming weight $\bar{w}_{\mathsf{opt}}$. After that, we can use the attack described previously with optimal complexity.

*Application to parameter vector II.* As these parameters are in the case $t \geq 2m\eta(1 - \eta)$, we can use Algorithm 2 in its optimum complexity to attack Random-HB# and HB#. After computing $\bar{w}_{\mathsf{opt}} = 77.167$, $P'(\bar{w}_{\mathsf{opt}}) = 0.0431$, $R_{\mathsf{opt}} = 269.39$ and the expected value of $\bar{w} = m\eta = 55$, we have to flip $f = 29$ bits to get an expected value close to $\bar{w}_{\mathsf{opt}}$.

For Random-HB# the number of bits to retrieve is $\ell = (k_x + k_y)m = 261\,072$ for which we can use $\theta = 3.164$. The total complexity is $\ell\theta^2 R_{\mathsf{opt}} = 2^{29.4}$. In the case of HB# the number of secret bits is $\ell = k_x + k_y + 2m - 2 = 1\,472$ for which we use $\theta = 2.265$ and end up with complexity of $\ell\theta^2 R_{\mathsf{opt}} = 2^{21}$.

**Strategy for $t$ close to $2m\eta(1 - \eta)$.** The case $t < 2m\eta(1 - \eta)$ is trickier to address since the expected value of $\bar{w}$ becomes *greater* than $w_{\mathsf{opt}}$. To achieve the same complexity as the previous case we would have to reduce the Hamming weight of $\bar{\nu}$ which is infeasible in polynomial time due to the hardness of the LPN problem.

However, if $t$ is a only a little less than $2m\eta(1-\eta)$ then the expected value of $\bar{w}$ is not far from $w_{\text{opt}}$. So, we can use Algorithm 2 without flipping any bit of $\bar{z}$ and the complexity is still polynomial. To further speed up the attack, we can remove errors from $\bar{z}$ in step 9 of Algorithm 2 as they are recovered until we reach $\bar{w} = w_{\text{opt}}$ which we can expect to happen at iteration $i = \left\lceil \frac{\bar{w}_0 - \bar{w}_{\text{opt}}}{\bar{w}_0} \right\rceil$.

*Application to Parameter Set I.* Although the asymptotic complexity of the attack is exponential in $\ell$ when $t < 2m\eta(1-\eta)$, we can nevertheless apply the attack on parameter set I proposed for HB# and Random-HB#. We first compute $\bar{w}_0 = m\eta = 291$, $\bar{w}_{\text{opt}} = 228$, $P'(\bar{w}_{\text{opt}}) = 0.0135$, $R(\bar{w}_0) = 15\,532$ and $R_{\text{opt}} = 2742.6$.

For Random-HB#, the number of key bits is $\ell = (k_x + k_y)m = 689\,088$ and $\theta = 3.308$ is enough to guarantee that $\mathsf{erfc}(\theta) \leq \frac{1}{689\,088}$. Hence, we obtain a total complexity of $\ell\theta^2(\frac{\bar{w}_0 - \bar{w}_{\text{opt}}}{\bar{w}_0} R(\bar{w}_0) + \frac{\bar{w}_{\text{opt}}}{\bar{w}_0} R_{\text{opt}}) = 2^{35.4}$. For HB#, we have $\ell = k_x + k_y + 2m - 2 = 2\,918$ secret bits to retrieve, so $\theta_2 = 2.401$ is enough and we get a total complexity of $\ell\theta^2(\frac{\bar{w}_0 - \bar{w}_{\text{opt}}}{\bar{w}_0} R(\bar{w}_0) + \frac{\bar{w}_{\text{opt}}}{\bar{w}_0} R_{\text{opt}}) = 2^{26.6}$.

### 4.3.5 *Optimizing the Attack*

**Algorithm 3** Finding errors in $|J|$-bit windows

```
Input: $\bar{a}, \bar{b}, \bar{z}, \bar{w} = \mathsf{wt}(\bar{a}X \oplus \bar{b}Y \oplus \bar{z})$, a set $J \subseteq \{0, 1, \cdots m\}$ and $w_J$ the number of
    non-zero $(\bar{a}X \oplus \bar{b}Y \oplus \bar{z})_j, j \in J$
Output: $I \subseteq J$ containing the $j$ with non-zero $(\bar{a}X \oplus \bar{b}Y \oplus \bar{z})_j, j \in J$.
Processing:
if $w_J = 0$ then
    $I \leftarrow \emptyset$
else if $w_J = |J|$ then
    $I \leftarrow J$
else
    Choose $J_1 \subseteq J$ such that $|J_1| = \lceil |J|/2 \rceil$.
    Set $\nu'$ the $m$-vector with $\nu'_j = 1$ iff $j \in J_1$
    Call Algorithm 1 on input $(\bar{a}, \bar{b}, \bar{z} \oplus \nu', n = 4\theta^2 R(\bar{w}))$ to get $w'$.
    Call Algorithm 3 with $(\bar{a}, \bar{b}, \bar{z}, \bar{w}, J_1, w_{J_1} = (\bar{w} + |J_1| - w')/2)$ to get $I_1$
    Call Algorithm 3 with $(\bar{a}, \bar{b}, \bar{z}, \bar{w}, J \setminus J_1, w_J - w_{J_1})$ to get $I_2$
    $I \leftarrow I_1 \cup I_2$
end if
```

This section is dedicated to optimize the attack we presented earlier in this chapter.

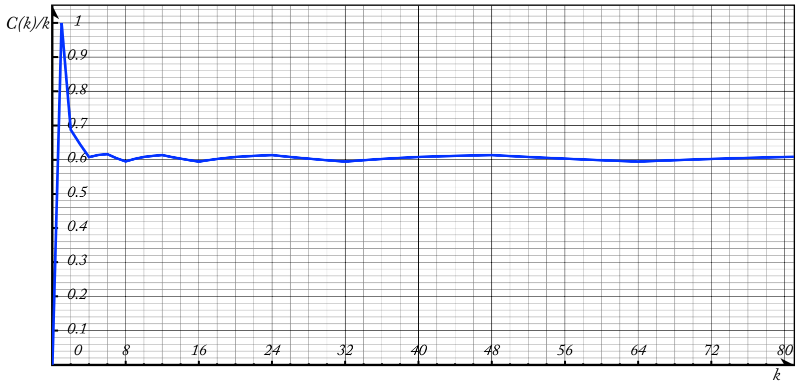

**Figure 4.3:** Plot of the function $C(k)/k$. Note that the function has local minima at values which are powers of 2.

Recall that our attack consists of two phases. At first, we recover the Hamming weight of the error vector and then use that result to compute all its bits. This second step is implemented by Algorithm 2 by solving the following problem: given a $m$-bit vector $\nu$ of Hamming weight $w$ and an oracle measuring whether each bit is 1 or 0 (Algorithm 1), what is the minimal number of measurements to fully recover $\nu$?

Algorithm 2 solves this problem inefficiently by performing $m$ measurements. Instead, Erdős and Rényi showed in [oR63] that the minimal number of measurements required to fully recover $\nu$ is upper-bounded by $(m \log_2 9)/\log_2 m$ and proposed a method to achieve this complexity. For our case, we propose Algorithm 3, which is an adaptation of the method of Erdös and Rényi.

To determine the error positions in a $k$-bit window by measuring the weight, Algorithm 3 uses a divide-and-conquer strategy: it splits the vector into two windows of the same length, recovers the error positions of each of them and then applies this strategy recursively. As it will be shown later, this yields a lower number of measurements comparing to measuring a $k$-bit window bit by bit as Algorithm 2 does.

The number of invocations of Algorithm 1, $C_w(k)$, to fully recover a $k$-bit window with known Hamming weight $w$ by Algorithm 3 is defined by the recursive relation

$$C_w(k) = 1 + \sum_{i=\max\{0,w-\lfloor k/2\rfloor\}}^{\min\{w,\lceil k/2\rceil\}} \frac{\binom{\lfloor k/2\rfloor}{i}\binom{\lceil k/2\rceil}{w-i}}{\binom{k}{w}} \left(C_i(\lfloor k/2\rfloor) + C_{w-i}(\lceil k/2\rceil)\right),$$

with initial values

$$C_0(k) = 0, C_k(k) = 0 \qquad \text{for all } k \in \mathbf{N}^\star$$

We can also compute, $C(k)$, the average number of invocations of Algorithm 1 that are

Table 4.2: Complexity of measuring a 16-bit window applied to the parameter set I and II of HB#.

| | Parameter Set I | | Parameter Set II | |
|---|---|---|---|---|
| $k$ | $\frac{16\,C(k)}{k}$ | Cost measurement | $\frac{16\,C(k)}{k}$ | Cost measurement |
| 2 | 11 | $2^{15.95}$ | 9.75 | $2^{12.43}$ |
| 4 | 9.72 | $2^{15.96}$ | 7.404 | $2^{12.49}$ |
| 8 | 9.52 | $2^{15.99}$ | 6.71 | $2^{12.75}$ |
| 16 | 9.51 | $2^{16.11}$ | 6.69 | $2^{13.90}$ |

needed by Algorithm 3 to recover the erroneous positions in a $k$-bit window $s$:

$$\begin{aligned} C(k) &= 1 + \sum_{w=0}^{k} \binom{k}{w} \Pr[\mathsf{wt}(s) = w] \cdot C_w(k) \\ &= 1 + \sum_{w=0}^{k} \binom{k}{w} \eta^w (1-\eta)^{k-w} C_w(k) \end{aligned}$$

**Splitting the Error Vector.** Algorithm 4 takes benefit from Algorithm 3 and uses it to optimize the number of measurements needed to localize the introduced errors and output $m$ linear equations. Algorithm 4 splits the error vector introduced in a triplet $(\bar{a}, \bar{b}, \bar{z})$ to $m/k$ $k$-bit windows, and each one of these is recovered using Algorithm 3. Moreover, in order to minimize the cost of measurement, i.e, the complexity of Algorithm 1, the weight of the error vector introduced in the target triplet is manipulated to tend towards the optimal value $\bar{w}_{\mathsf{opt}} = \frac{t-m\eta}{1-2\eta}$. That is, once the algorithm learns $k$ positions, if the expected weight of the error vector, $\bar{w}_0$ is smaller than $\bar{w}_{\mathsf{opt}}$, then it flips at most $\bar{w}_{\mathsf{opt}} - \bar{w}_0$ bits of $\bar{z}$ that are at correct positions. In the opposite case, i.e., when $\bar{w}_0$ is greater than $\bar{w}_{\mathsf{opt}}$, the algorithm flips at most $\bar{w}_0 - \bar{w}_{\mathsf{opt}}$ bits of $\bar{z}$ that are at erroneous positions.

The number of calls to Algorithm 3 we need before reaching the optimal case $\bar{w} = \bar{w}_{\mathsf{opt}}$, is then

$$\begin{aligned} i &= \frac{\bar{w}_{\mathsf{opt}} - \bar{w}_0}{k(m - \bar{w}_0)} m && \text{when} \quad \bar{w}_{\mathsf{opt}} \geq \bar{w}_0, \\ i &= \frac{\bar{w}_0 - \bar{w}_{\mathsf{opt}}}{k \cdot \bar{w}_0} m && \text{when} \quad \bar{w}_{\mathsf{opt}} \leq \bar{w}_0. \end{aligned}$$

Hence, the full complexity of Algorithm 4 is

$$N = \theta^2 \left( iR(\bar{w}_0) + \left\lceil \frac{m}{k} - i \right\rceil R_{\mathsf{opt}} \right) C(k).$$

**Algorithm 4** Optimizing Algorithm 2

```
Input: $\bar{a}, \bar{b}, \bar{z}$ and $\bar{w}_0$ the expected value of $\bar{\nu} = \bar{a}X \oplus \bar{b}Y \oplus \bar{z}, k$
Output: A linear equation $\bar{a}X \oplus \bar{b}Y = \bar{c}$
Processing:
Initialize $m$-bit vector $\bar{c} \leftarrow \bar{z}$
Initialize $M \leftarrow \emptyset$
Call Algorithm 1 on input $(\bar{a}, \bar{b}, \bar{z}, n = 4\theta^2 R(\bar{w}_0))$ to get $\bar{w}$
Define a set $\mathcal{S}$ of $J_i = \{ik + 1, \ldots, \min((i + 1)k, m)\}, i = 1 \ldots \lceil \frac{m}{k} \rceil$
repeat
    Choose $J \in S$
    Call Algorithm 1 on input $(\bar{a}, \bar{b}, \bar{z} \oplus J, n = \theta^2 R(\bar{w}))$ to get $\bar{w}' = \mathsf{wt}(\bar{\nu} \wedge J)$
    Call Algorithm 3 with $(\bar{a}, \bar{b}, \bar{z}, \bar{w}, J, w_J = (\bar{w} + |J| - \bar{w}')/2)$ to get $I$
    Set $\bar{c}_i \leftarrow \bar{c}_i \oplus 1$ for all $i \in I$
    $M \leftarrow M \cup I$
    Remove $J$ from $S$
    if $\bar{w} > \bar{w}_{\mathsf{opt}}$ then
        Flip $\min(|I|, \bar{w} - \bar{w}_{\mathsf{opt}})$ bits $\bar{z}_i$ for which $i \in I$
        $\bar{w} \leftarrow \bar{w} - \min(|I|, \bar{w} - \bar{w}_{\mathsf{opt}})$
    else if $\bar{w} < \bar{w}_{\mathsf{opt}}$ then
        Flip $\min(|J \setminus I|, \bar{w}_{\mathsf{opt}} - \bar{w})$ bits $\bar{z}_i$ for which $i \in J \setminus I$
        $\bar{w} \leftarrow \bar{w} + \min(|J \setminus I|, \bar{w}_{\mathsf{opt}} - \bar{w})$
    end if
until $\mathcal{S} \neq \emptyset$
```

### 4.3.6 *Final Algorithm*

The final attack is described in Algorithm 5. The idea is to get a vector with low expected weight using Algorithm 6 and then find all the erroneous positions inserted by the tag to obtain $m$ linear equations and iterate this until we get enough equations to solve and find the secrets $X$ and $Y$. To get the lower complexity, we can flip the last bits of $\bar{z}$ so that we end up with an expected weight of $\bar{w}_{\text{opt}}$. We note that introducing errors in a full segment as defined by Step 4 of Algorithm 4 does not increase the needed number of measurements as $C_w(k) = C_{k-w}(k)$.

**Algorithm 5** Final attack on Random-HB# and HB#

```
Input: $k, w$
Output: $X, Y$ the secrets of the tag
Processing:
Initialize $\mathcal{S} \leftarrow \emptyset$
for $i = 1 \ldots 2 + \lceil \frac{\ell}{m} \rceil$ do
    Call algorithm 6 on input $w$ to get $\bar{a}, \bar{b}, \bar{z}$ with an error vector of expected weight $\bar{w}_0 = (m - w)\eta_w + w(1 - \eta^\circ_w)$
    if $\bar{w}_{\text{opt}} > \bar{w}_0$ then
        Flip the last $(\bar{w}_{\text{opt}} - m\eta)/(1 - 2\eta)$ bits of $\bar{z}$
        Set $\bar{w}_0 \leftarrow \bar{w}_{\text{opt}}$
    end if
    Call Algorithm 4 on input $(\bar{a}, \bar{b}, \bar{z}, \bar{w}_0, k)$ to get $m$ linear equations
    Insert linear equations in $\mathcal{S}$
end for
Solve $\mathcal{S}$
```

The full complexity in terms of intercepted authentications as

$$\left\lceil \frac{\ell}{m} \right\rceil \theta^2 \left( iR(\bar{w}_0) + \left\lceil \frac{m}{k} - i \right\rceil R_{\text{opt}} \right) C(k) \ + \ (2 + \frac{\ell}{m}) \frac{1}{P(w)}.$$

*Application to parameter vector II* With with input $k = 8$ and $w = 300$ we obtain $P(w) = 2^{-7}$, $\bar{w}_0 = 273$ and $\bar{w}_{\text{opt}} = 228$, $i = 24$, $R_{\text{opt}} = 2742.6$, $R(\bar{w}_0) = 7\,026.4$. So the full complexity of the attack is derived from $\theta$ and $\ell$ as shown in Section 4.3.4. This is $2^{25}$ sessions for HB# and $2^{33.8}$ for Random-HB#.

*Application to parameter vector II* In this case, we have $k = 8, w = 0$ and $\bar{w}_0 = 55$. We flip 29 bits to obtain an error vector of expected weight $\bar{w}_{\text{opt}} = 77$, which yields $R_{\text{opt}} = 269.39$ and $i = 0$. The complexity is $2^{19.7}$ sessions for HB# and $2^{28.1}$ for Random-HB#.

## 4.4 Thwarting the Attack: the Case of Small $t$

The case of lower $t$, the false acceptance rate will be very low but the false rejection rate of HB# becomes high (e.g. 0.5 for $t = m\eta$) so that it would require more than one authentication in average for the tag to authenticate itself. The main advantage of this approach is that the complexity of Algorithm 1 becomes exponential. Here, we present a better strategy than calling Algorithm 2 with an triplet $(\bar{a}, \bar{b}, \bar{z})$ obtained by a simple passive attack.

Our goal is to call Algorithm 2 with a $\bar{w}_0$ as low as possible. During the protocol, we can set $(\hat{a}, \hat{b}, \hat{z})$ to $(a, b, z \oplus \bar{\nu})$ with $\bar{\nu}$ of weight $\bar{w}$ until the verifier accepts $\hat{z}$. Then, we launch our attack with $(\bar{a}, \bar{b}, \bar{z}) = (a, b, z)$. A detailed description is showed in Algorithm 6.

**Algorithm 6** Getting $(a, b, z)$ with low Hamming weight

```
Input: $\bar{w}$
Output: $(a, b, z)$ such that $(aX \oplus bY \oplus z)$ has low weight.
Processing:
Pick random vector $\bar{\nu}$ of Hamming weight $\bar{w}$
repeat
    During a protocol with messages $(a, b, z)$, set $\hat{z} = z \oplus \bar{\nu}$
until verifier accepts
```

The probability that the verifier accepts the session with transcript $(a, b, \hat{z})$ is $P(\bar{w})$. This latter can be reformulated as

$$P(\bar{w}) = \sum_{j=0}^{t} \left( \binom{m - \bar{w}}{j} \eta^j (1 - \eta)^{m - \bar{w} - j} \cdot \sum_{i=0}^{t-j} \binom{\bar{w}}{i} \eta^{\bar{w} - i} (1 - \eta)^i \right)$$

When the verifier accepts, the $m - \bar{w}$ positions not in the support of $\bar{\nu}$ are erroneous with probability

$$\eta_{\bar{w}} = \frac{\sum_{j=0}^{t} \left( j \binom{m - \bar{w}}{j} \eta^j (1 - \eta)^{m - \bar{w} - j} \cdot \sum_{i=0}^{t-j} \binom{\bar{w}}{i} \eta^{\bar{w} - i} (1 - \eta)^i \right)}{(m - \bar{w}) P(\bar{w})}.$$

On the other hand, the other positions of $\hat{z}$ in the support of $\bar{\nu}$ are non-zero with probability

$$\eta^{\circ}_{\bar{w}} = \frac{\sum_{j=0}^{t} \left( \binom{m - \bar{w}}{j} \eta^j (1 - \eta)^{m - \bar{w} - j} \cdot \sum_{i=0}^{t-j} i \binom{\bar{w}}{i} \eta^{\bar{w} - i} (1 - \eta)^i \right)}{\bar{w} P(\bar{w})}.$$

Thus, because of the high false rejection rate, if the session with transcript $(a, b, \hat{z})$ gets accepted, then we can expect that the error vector $\nu$, introduced in the original triplet $(a, b, z)$ from the protocol, has weight $\bar{w}_0 = (m - \bar{w})\eta_{\bar{w}} + \bar{w}(1 - \eta^{\circ}_{\bar{w}})$.

**Table 4.3:** Attack cost for the initial bit of the shared key for HB# applied to $t = \lceil \eta m \rfloor$

| Parameter set | Algorithm 1 | Algorithms 6 + 1 |
|---|---|---|
| I | $2^{78}$ | $2^{58.5}$ |
| II | $2^{30}$ | $2^{21}$ |

*Application to Parameter Set II with* $t = 55$ Assume that for the parameter set II we set $t = m\eta \approx 55$. Then, an accepted vector obtained by a passive attack will most likely have weight

$$\begin{aligned}\bar{w}_0 &= (m - \bar{w})\eta_0 + \bar{w}(1 - \eta_0^\circ) \\ &\approx 50\end{aligned}$$

so $4\theta^2 R(\bar{w}_0) = 2^{30}$ operations are sufficient to determine its correct weight.

Calling Algorithm 6 with $\bar{w} = 41$, we get a triplet $(a, b, z)$ with error vector $\nu$ of Hamming weight $\bar{w}_0 = (m - \bar{w})\eta_{41} + \bar{w}(1 - \eta_{41}^\circ) \approx 33$ in $\frac{1}{P(\bar{w})} = 2^{20}$ authentications and can recover the weight of $\nu$ in another $4\theta^2 R(33) = 2^{20}$ operations with Algorithm 1.

Table 4.3 shows the costs to determine the first bit of the secret key, i.e., calling Algorithm 1 with a random vector obtained by a passive attack in comparison to calling Algorithm 6 first and then Algorithm 1 with its output. Note, that recovering successive bits is always cheaper.

## 4.5 Thwarting the Attack: the Case of Vectors without False Rejections

To thwart the previous attacks without taking parameter sets with huge $m$ or high false rejection rate, we could change the protocol so that the prover generates a vector $\nu$ of constant or bounded Hamming weight. In this section we will show that this leads to different attacks.

Assume that the prover accepts a protocol instance $(a, b, z)$, in which an error vector $\nu$ was introduced by the prover, if and only if $\mathsf{wt}(aX \oplus bY \oplus Z) = t$, then, as $\bigoplus_{i=1}^{m} \nu_i = t \bmod 2$, we can write

$$\begin{aligned}\bigoplus_{i=1}^{m}(aX \oplus bY)_i &= \bigoplus_{i=1}^{m}(z_i \oplus \nu_i) \\ &= \bigoplus_{i=1}^{m} z_i \oplus (t \bmod 2)\end{aligned}$$

Using this equality, the adversary can flip two bits of $z$, i.e., she generates a random vector $\bar{\nu}$ of Hamming weight 2. In other words, $\bar{\nu}$ contains exactly two bits set to one, at position $i$ and $j$, while all the other bits are set to 0. When the verifier accepts, the adversary deduces exactly one of the two bits of $z$ that were flipped is at an erroneous position, i.e, that $\nu_i \oplus \nu_j = 1$. In the other case, either the two bit positions are both erroneous or both correct, i.e., $\nu_i \oplus \nu_j = 0$. Hence, through that manipulation, the adversary obtains

$$(aX \oplus bY)\bar{\nu}^\top = z\bar{\nu}^\top \oplus \begin{cases} 1 & \text{if authentication succeeded} \\ 0 & \text{if authentication failed} \end{cases}$$

The probability that the verifier accepts $\hat{z}$ is equal to the probability that exactly one erroneous position was flipped by the effect of $\bar{(nu)}$, which is

$$\frac{\binom{m-w}{1}\binom{w}{1}}{\binom{m}{2}} = \frac{2w(m-w)}{m(m-1)}.$$

*Generalization* The above approach may be generalized to the case where the Hamming weight of $\nu$ is bounded in the original protocol, i.e. when the verifier accepts if $\mathsf{wt}(\nu) \leq t$ and the prover does not generate error vectors with Hamming weight greater than $t$. This was suggested by Gilbert et al. for parameter set III.

Again, the attacker can replace the message $z$ by $\hat{z} = z \oplus \bar{\nu}$ where $\bar{\nu}$ is an $m$-bit vector of Hamming weight 2. In such scenario, authentication fails only when $\mathsf{wt}(\nu) \in \{t-1, t\}$ and the attacker flipped two non-erroneous positions. When this event occurs, the attacker learns two error positions corresponding to the bits set to 1 in $\nu$, i.e.,

$$(aX \oplus bY)_i = z_i, \quad \bar{\nu}_i \neq 0.$$

The success probability of this attack is related to the probability of getting a triplet with an error vector of Hamming weight equal to $t-1$ or $t$, and that the adversary picks a $\bar{\nu}$ that has the effect of flipping two correct positions. It can be computed as

$$q = \frac{\sum_{i=0}^{t} \binom{m}{t-i} \eta^{t-i} (1-\eta)^{m-t+i} \frac{\binom{m-t+i}{2}}{\binom{m}{2}}}{\sum_{i=0}^{t} \binom{m}{i} \eta^i (1-\eta)^{m-i}}$$

*Application to parameter vector III* For the parameter vector III, the attacker learns two bits about the secret key every $1/q = 2^{9.02} \approx 512$ iterations. This is 16 times faster than an attack by Algorithm 1 and needs only $\ell \cdot 2/q = 2^{26}$ to recover a Random-HB# secret key ($2^{19}$ for HB#).

## 4.6 Secure Parameters for Random-HB# and HB#

In this section, we investigate the lower bounds on the parameter sets for which our attack is not effective. Before that, we need to clarify the notion of bit security in HB#.

**Definition 4.6** ($s$**-bit Secure Parameter Set for HB#**)
*For HB#, a parameter set* $(k_x, k_y, m, \eta, t)$ *is said to achieve* $s$*-bit security if recovering one bit of information about the secret matrices* $X_x$ *and* $Y_y$ *or making the verifier accept a session without matching conversation requires an attack with complexity (in terms of protocol sessions) within an order of magnitude of at least* $2^s$ *and comparable time complexity.*

Let us assume that Algorithm 3 succeeds with a total error weight of $t$ when the added error vector has weight $w_0$. We can thus expect that

$$t \approx (m - \bar{w})\frac{w_0}{m} + \bar{w}\left(1 - \frac{w_0}{m}\right) \quad \Longrightarrow \quad w_0 \approx m\frac{t - \bar{w}}{m - 2\bar{w}},$$

since $t < {}^{m}\!/2$ and $w_0$ is a decreasing function in terms of $\bar{w}$.

By using Algorithm 3 with input $\bar{w}$ we can get $(a, b, z)$ such that $\nu = aX \oplus bY \oplus z$ has expected weight $w_0$ in complexity ${}^{1}\!/_{P(\bar{w})}$. Based on this triplet we can run Algorithm 1 twice, once with $r = {}^{1}\!/2$ and then with $r = 1$, with complexity $3\theta^2 R(w_0)$ and recover one bit of information about the matrices with error probability bounded by $\frac{3}{2}\mathsf{erfc}(\theta)$. For $\theta = 1$, this probability is less than ${}^{1}\!/4$. Following our algorithms, the time complexity is "reasonably comparable" (and even negligible) to the complexity in terms of protocol sessions. So we conclude by saying that parameters $m, \eta, t$ leading to the existence of $\bar{w}$ such that

$$\begin{cases} C_1 = \frac{1}{P(\bar{w})} + 3R(w_0) \leq 2^s \\ w_0 = m\frac{t - \bar{w}}{m - 2\bar{w}} \end{cases}$$

are insecure.

Computing the maximal $\bar{w}$ for which $\frac{1}{P(\bar{w})}$ is high, we have $P(\bar{w}) \approx \Phi(u)$ so we expect to have a very small negative $u$ and we can use the approximation $P(\bar{w}) \approx -\varphi(u)/u$. Applying this reasoning to a very low $P(w_0)$, we obtain

$$\begin{aligned} R(w_0) &\approx 4R_{\mathsf{opt}}P(w_0)e^{u_0^2} \\ &\approx -4R_{\mathsf{opt}}\frac{\varphi(u_0)}{u_0}e^{u_0^2}, \end{aligned}$$

and we derive the complexity

$$C_1 \approx -u\sqrt{2\pi}e^{\frac{u^2}{2}} - \frac{4R_{\mathsf{opt}}}{u_0\sqrt{2\pi}}e^{\frac{u_0^2}{2}}.$$

The exponential terms only match when $u = u_0$, leading to $\bar{w} = w_0$ which translates into

$$\bar{w} = \frac{t}{2} \times \frac{m}{m - \bar{w}} \approx \frac{t}{2} \Longrightarrow u_0 = u \approx \frac{t(\frac{1}{2} + \eta) - m\eta}{\sqrt{m\eta(1 - \eta)}}$$

As the equality $e^{\frac{u^2}{2}} = e^{\frac{u_0^2}{2}} = 2^s$ only holds when $u^2 = 2s \ln 2$, we can replace the value of $u$ from the previous expression to obtain

$$2s \ln 2 = \frac{(2m\eta - t(1+2\eta))^2}{m(1-(1-2\eta)^2)} \tag{4.1}$$

Thus, any parameter set involving $(m, t, \eta)$ that satisfies this equation is insecure.

In the following, we use this equation to derive bounds on secure parameters for HB#, depending on the case:

- The worst case complexity in our attack is obtained with minimal $t$, namely for $t = m\eta$. Note that with such a threshold, the honest prover gets rejected with probability $1/2$. In this case, Equation 4.1 simplifies to

  $$m = \frac{2 \ln 2}{\eta^2} \left( \frac{1}{(1-2\eta)^2} - 1 \right) \times s$$

  Hence, any $(m, \eta)$ satisfying this equation is insecure. Using this result, we can derive bounds for secure parameters for HB#

  - For $\eta = 1/8$ and $s = 80$ we obtain $m = 5\,521$ and $t = 690$. We conclude that any parameter with $\eta = 1/8$, $m \leq 5\,521$, and $t = \lfloor m\eta \rceil$ is insecure in the sense that there exists an attack with complexity within the order of magnitude of $2^{80}$.
  - Similarly, with $\eta = 1/4$ and $s = 80$ we find that $m = 5\,323$ and so $t = 1\,331$. Hence, when $\eta = 1/4$ any parameter set with $m \leq 5\,323$ is insecure.

- General applications require the false rejection rate to be very low, i.e., in the order of $2^{-s/2}$. This constraint induces a new equation for the parameters

  $$\sum_{i=t+1}^{m} \binom{m}{i} \eta^i (1-\eta)^{m-i} \leq 2^{s/2}.$$

  Note that we can rewrite Equation (4.1) as

  $$t = \frac{1}{1+2\eta} \left( 2m\eta - (1-2\eta)\sqrt{2ms \ln 2} \right).$$

  On their own, these two equations are not sufficient. Indeed, to satisfy both of tem, we could set a threshold $t$ small enough such that false rejects are negligible and sufficiently far from the expected value $m\eta$ such that the attack does not apply. However, we have also to take care of the most rudimental attack an adversary can perform: sending random vectors $z$. For this reason, we require the false acceptance rate, $P_{\mathsf{FA}}$ to be smaller than $2^{-s}$, i.e.,

  $$\sum_{i=0}^{t} \binom{m}{i} \leq 2^{m-s}$$

**Table 4.4:** Summary of the complexity of our attacks.

| Parameter Set | $k_X$ | $k_Y$ | $m$ | $\eta$ | $t$ | Random-HB# | HB# |
|---|---|---|---|---|---|---|---|
| I | 80 | 512 | 1164 | 0.25 | 405 | $2^{34}$ | $2^{25}$ |
| II | 80 | 512 | 441 | 0.125 | 113 | $2^{28}$ | $2^{20}$ |
| III($w$ bounded) | 80 | 512 | 256 | 0.125 | 48 | $2^{26}$ | $2^{19}$ |

Combining the last three equations yield a bound on the size of the parameters that induce a secure HB# with practical false acceptance and rejection rates. Unfortunately, satisfying parameters are too large. That is, no $m$ smaller than $15\,000$, for both values of $\eta$, satisfies the equations.

## 4.7 Perspectives

As it is depicted in Table 4.4, the attack we presented in this chapter is devastating for all the proposed parameter sets of HB#. On the other side, we could not propose an easy fix against it.

Recently, new proposals for protocols whose security reduces to the LPN problem has been published in a paper by Kiltz et al. [KPC+11]. Their work essentially consists of two contributions. The first one is about illustrating a two-round authentication protocol secure against active adversaries. Yet, the protocol can be shown to be insecure in the MIM model. While the proposed protocol has the advantage over HB$^+$ of having fewer messages to exchange and a tighter reduction gap, it requires the prover, i.e., the RFID tag, to perform additional computations by checking the Hamming weight of the challenge. Still, the authors note that it is possible to eliminate this step at the cost of adding to the secret key two $n$-bit vectors.

The second contribution of the paper is to design a MAC whose existential unforgeability reduces to the LPN problem and use that MAC in a secure challenge-response protocol.

# 5

# CHALLENGING SQUASH'S SECURITY ARGUMENTS

Contents

The simplest authentication protocols are challenge-response protocols in which the verifier starts by sending a challenge to which the answer answers by computing a tag for the challenge using a function, usually a message authentication code, as it is shown in Figure 5.1, or a digital signature algorithm, of the received challenge and a secret key shared between the two parties. Upon reception of the response, the verifier recomputes the expected answer and acknowledge the verifier if both values match.

It is commonly assumed that when dealing with constrained devices such as RFID tags, it is preferable to use symmetric-key cryptographic primitives, i.e., MAC, over public-key primitives such as digital signatures. When based on a MAC, such a challenge-response protocol requires the underlying MAC to be unforgeable under chosen message attacks to be secure. In short, this model considers adversaries that have access to an oracle to which they can submit adaptively chosen messages. The MAC is considered to be secure in a classical sense if no adversary is able to produce a tag for a chosen value that was not submitted to the oracle, except with a very small probability.

In this chapter, we study a proposal for a MAC destined to constrained environments made by Shamir [Sha07] called SQUASH. The results of this analysis are the subject to a paper published at EuroCrypt 2009 [OV09].

## 5.1 SQUASH

### 5.1.1 *Description*

The central idea of SQUASH is borrowed from one of the oldest and most studied public-key cryptosystems: Rabin's cryptosystem. Constructed around a public hard to factor modulus $N$, the Rabin encryption scheme is based on the trapdoor one-way function $f(x) = x^2 \bmod N$. Besides its nice efficiency properties, this function exhibits a strong connection to the factoring problem: it can be easily proven that inverting it leads to the immediate factorization of the modulus $N$. For this reason, and the extensive efforts made on factorization algorithms, Rabin's function is widely considered as a candidate one-way function.

Unfortunately, the implementation of such a function on an RFID tag with appropriate sizes for the modulus is beyond the reach of low-cost RFID tags. Still, Adi Shamir presented at the RFID Security Workshop 2007, a MAC built around the Rabin function which surprisingly enjoys a very compact implementation. Moreover, the presented scheme provided some kind of provable security inherited from the Rabin function.

The first design of SQUASH was a randomized version of the Rabin cryptosystem similar to an earlier proposal for smart-cards [Sha95]. The idea of this randomized version is to get rid of the modular reduction, which is the responsible of the expensiveness of implementing exponentiation in $\mathbf{Z}_N^\star$, and replace it by a regular squaring over the $\mathbf{Z}$. However, computing square roots over the plain integers can be done efficiently. Hence, in order to prevent the

| **Prover** | **Shared Secret:** $K$ | **Verifier** |
|---|---|---|
| | $\xleftarrow{\quad a \quad}$ | Choose $a \in_R \{0,1\}^\alpha$ |
| Compute $t \leftarrow \mathsf{MAC}_K(a)$ | $\xrightarrow{\quad t \quad}$ | **Output:** $\mathsf{Check}_K(a,t)$ |

**Figure 5.1:** Basic challenge-response authentication protocol based on a MAC.

inversion of the function, the output was randomized by adding a multiple of the modulus $N$. In other words, the randomized Rabin function is to compute

$$f(x) = x^2 + r \cdot N,$$

for a randomly chosen $r$ of bit-length larger than $N$'s by $\delta$ bits.

Note that a simple modular reduction removes the effect of $r$. Hence, the recipient can manipulate the output of the randomized Rabin function as he would do with the classical Rabin function. Moreover, the following theorem establishes the security of the scheme.

**Theorem 5.1**
*Let $k \in \mathbf{N}$ denote a security parameter and $\ell(\cdot)$ and $\delta(\cdot)$ two polynomial functions that define the bit-size of $N$ and $r$, respectively. If $\epsilon(\ell(k))$ upper-bounds the probability that an adversary inverts the Rabin function, then no polynomial-time adversary inverts the randomized Rabin function with probability greater than $\epsilon(\ell(k)) + 2^{-\delta(k)}$.*

Even if the randomized Rabin function was designed for smart-cards which, at the time of its publication, were suffering from the same computational restrictions RFID tags are encountering, implementing this function on RFID tags leads to another drawback. The fact that, contrarily to smart-cards which communicated through a physical channel, RFID tags communicates over wireless channels, restricts them in the amount of data they can transmit. Considering this, the large amount of bits a tag has to transmit in the randomized Rabin function is undesirable.

To counter that issue, the SQUASH proposal was to release only a small window in the middle of the $\ell$-bit result of the original Rabin squaring. As it is depicted in Figure 5.2, SQUASH is composed of three steps. At first, it applies a mixing function that takes as input a message and a secret key. Given a public hard-to-factor modulus $N$, it converts the output of the mixing function to an element of $\mathbf{Z}_N^\star$ and then squares the output of the mixing function modulo $N$. In the end, it outputs a window of consecutive bits, consisting of bits that in between positions $a$ and $b$, with $b > a$. Mathematically, SQUASH is described by the following formula.

$$T = \left(2^b \left(F^2(K,m) \mod N\right)\right) \mod 2^a$$

We note that verification is done in the same way: the verifier recomputes the MAC value from the given message and the key. He then checks that it corresponds to the one supplied by the sender.

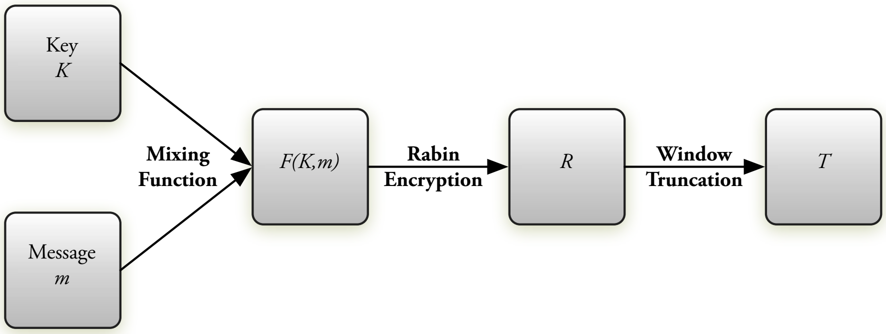

**Figure 5.2:** The Three steps of SQUASH. First it mixes the received challenge with the internal secret key, then computes the Rabin encryption of the result. At last, it outputs specific bits from the encryption.

### 5.1.2 *Implementation Trick and Shamir's Challenge*

The first observation is that SQUASH, as a MAC in which both participants share a secret key and the verifier only needs to "recompute" the operations performed by the sender, does not need the invertibility feature of the Rabin function. Therefore, universal moduli with unknown factorization can be used. In particular, Shamir suggested the use of integers of the form $N = 2^\ell - 1$, known as Mersenne numbers. Again, the modular reduction with such moduli can be well approximated by a simple bit shift. We note that numbers of the form $N = a^\ell - 1$, known as Cunningham numbers, can as well be used. Shamir even mentioned the possible use of more general numbers of the form $N = a \cdot b^c \pm d$ for small $a, b, c, d$.

Besides having a simple modular reduction, the trick behind reducing the cost implementation of SQUASH resolves around a central observation: Since the MAC is not revealing the whole output from the Rabin function, there is no need in computing it entirely. Instead, Shamir proposed to compute an approximation of the final window by starting the convolution in "the middle". Concretely, instead of computing the convolution from the least significant bit and keep the most significant bits that are to be released, it starts at a bit of higher position. However, the eventual carry that propagate from least significant bits can lead to wrong computations of the output. To address this issue, Shamir proposed to start the approximation a few bits before the first one to be outputted. Although the first bits of the convolution may be wrongly computed, having some "guard bits" to "absorb" the carry helps in decreasing the probability of a wrong approximation as it is sufficient that a single bit position in the guard bits does not produce a carry for the window to be correctly computed.

In parallel to the theoretical SQUASH, Shamir proposed a practical implementation with a more aggressive optimization that he called SQUASH-128. Although the initial proposal for SQUASH was to use a universal modulus such as the Mersenne numbers C365 $= 2^{1237} - 1$ and C385 $= 2^{1277} - 1$, whose factorizations are still unknown, it was noted that knowing the full factorization of the modulus seems to not help the adversary in reconstructing the full output of the mixing function. Therefore, the suggestion was to use a very small modulus, $N = 2^{128} - 1$, which factorization can be easily computed as

$$2^{128} - 1 = 3 \times 5 \times 17 \times 257 \times 641 \times 65\,537 \times 274\,177 \times 6\,700417 \times 67\,280\,421\,310\,721.$$

For the mixing function, his proposal was to use the non-linear feedback shift register of the stream cipher GRAIN-128 [HJMM08]. Concretely, Shamir proposed to initialize the 128-bit register with the 64-bit key in its least significant half and with the key XORed with the 64-bit message in its most significant one. The register is then to be clocked 512 times, twice more than in GRAIN-128, and subsequently squared modulo $2^{128} - 1$. Finally, a 32-bit window, consisting of bits between positions 48 and 79, inclusive in both ends, is to be released as the output. In order to correctly perform the window approximation, 8 bits, the ones ranging from positions 40 to 47, are to be computed as guard bits.

## 5.2 SQUASH-0 and SQUASH-1

For the rest of this chapter, we denote by $K$, $M$, $R$, and $T$ the key, message, MAC, and truncation function of SQUASH respectively. The function SQUASH will simply consist of the following:

$$R = T\left(\left(\sum_{i=0}^{\ell-1} 2^i \times f_i(K,M)\right)^2 \bmod N\right),$$

where the $f_i$'s are Boolean functions and the truncation function $T$ is defined by

$$T(x) = \left\lfloor \frac{x \bmod 2^b}{2^a} \right\rfloor. \tag{5.1}$$

By expanding the square, we obtain:

$$R = T\left(\left(\sum_{i=0}^{\ell-1}\sum_{i'=0}^{\ell-1} 2^{i+i'} \times f_i(K,M) f_{i'}(K,M)\right) \bmod N\right) \tag{5.2}$$

The version of SQUASH presented in 2007, which we call SQUASH-0, uses a mixing function $f$ expanding (using a linear feedback shift register) the XOR of the key and the challenge. Due to a private comment by Vaudenay about an attack against SQUASH without truncation, it was updated, in the proceedings version [Sha08], to use a non-linear function.

Since the mixing function outputs $\ell$-bit integers to be fed to the Rabin function, we can represent every bit position $i$, for $i \in [\![0, \ell]\!]$, by linear functions $f_i(K, M) = g_i(K) \oplus L_i(M)$.

In order to algebraically manipulate SQUASH, we need to map the bitwise XOR operation to operations over the integers. For this, we have two possibilities. The first option is to use the relation

$$a \oplus b = a + b - 2ab,$$

which maps the XOR to addition and multiplication. The second option would be to define the notation $\hat{a} = (-1)^a$, $\hat{b} = (-1)^b$ and the map the XOR operation using the relation

$$a \oplus b = \frac{1 - \hat{a}\hat{b}}{2}.$$

This last representation presents the nice property that only one operation between the operands is actually performed, leading to a simpler overall representation. For this reason, we will use it in the rest of this chapter. Hence, by setting $k_i = (-1)^{g_i(K)}$ and $m_i = (-1)^{L_i(M)}$, we apply the mapping to Equation (5.2) and we obtain the following equation which is the starting point of our analysis

$$R = T\left(\frac{1}{4}\sum_{i,i'} 2^{i+i'} m_i m_{i'} k_i k_{i'} - \frac{2^\ell - 1}{2}\sum_i 2^i m_i k_i + \frac{(2^\ell - 1)^2}{4} \bmod N\right) \quad (5.3)$$

Interestingly, when $N$ is a Mersenne number, i.e., $N = 2^\ell - 1$, this last equation simplifies to

$$R = T\left(\frac{1}{4}\sum_{i,i'} 2^{i+i'} m_i m_{i'} k_i k_{i'} \mod (2^\ell - 1)\right) \quad (5.4)$$

In the sequel, we first present the attack by Vaudenay, i.e., without truncation, and apply it to any mixing function of form $g(K) \oplus L(C)$. We then improve on the attack by letting the adversary choose the messages in its attack. At last, we show how to apply this last attack to the case in which a window of consecutive bits is returned, i.e., against the original SQUASH-0. In order to give evidence about the efficiency of each of the attacks we present, we provide, in each case, a numerical application with moduli of size 1024 bits in the case of no truncation and 128 bits in the case of SQUASH-0.

We note that our analysis translates into an attack against Rabin-SAEP case with "known random coins" in which the adversary can request many encryptions of the same plaintext with different randomness and learn, along with the ciphertexts, the random bits used by the algorithm. We stress that this type of attacks are irrelevant for public-key encryption schemes, i.e., the security of Rabin-SAEP is not concerned by our analysis.

## 5.3 Known Message Attack on SQUASH-0 without Window Truncation

In a first step, we omit the truncation function, i.e., we set the function $T$ to be the identity function. A first attack consists of collecting enough equations of the form of Equation (5.3) and solving them using standard algebraic techniques such as linearization [KS99].

Simple linearization consists in expressing every quadratic term $k_i k_{i'}$ as a new unknown to obtain a system of linear equations. Although we get a system with $1/2\ell(\ell+1)$ unknowns, we only need $1/2r(r+1)$ equations. Combining these latter with the equations induced by the $g_i$'s, we can recover the $r$ secret bits. Using standard Gaussian elimination techniques, solving the system requires to perform $O(r^6)$ multiplications, each one of them would be computed with complexity $O(\ell^2)$. Hence, the overall time complexity is $O(\ell^2 r^6)$ while the data complexity to store all the equations would be $O(\ell r^2)$ bits.

We get unknowns and a solving algorithm of complexity $O(\ell^2 r^6)$ (as for $O(r^6)$ multiplications with complexity $O(\ell^2)$) after collection of $O(\ell r^2)$ bits (as for $O(r^2)$ samples of $O(\ell)$ bits). Since $k_i k_{i'} = \pm 1$ which is unexpectedly small, we can also consider algorithms based on lattice reduction using $O(\ell)$ samples only. The attack works even if the $L_i$'s are not linear.

Interestingly, we note that when $N$ is a Mersenne number then $N = 2^\ell - 1$ so Equation (5.3) simplifies by getting rid of $r$ unknowns. Therefore, we have $\frac{r(r-1)}{2}$ unknowns instead of $\frac{r(r+1)}{2}$ and the number of equations needed for the attack decrease accordingly. However, for $\ell$ resp. $r$ in the order of magnitude of $2^{10}$ resp. $2^6$, complexities are still very high.

## 5.4 Chosen Message Attack on SQUASH-0 without Window Truncation

When forced to random messages, we can only solve Equation (5.3) by collecting enough equations and deriving from them equations with only one unknown term through linearization. On the other hand, when the adversary is allowed to choose arbitrary values for the messages, he could try to produce equations with only one unknown term. However, there are two restrictions in choosing the messages. First, the $m_i$'s can only take values in $\{-1, 1\}$ and then one can only obtain sparse equations but through combinations of several equations. Second, the $m_i$'s can not be chosen independently because they result from some expansion. Obviously, when r is very small against $\ell$, there is no way to ensure that a preimage exists of an arbitrary $\ell$-bit expanded value exists. Therefore, one can only select expanded messages with properties accessible from random pickings in the $r$-bit message space. However, in the particular case when the expansion is linear, we can construct vectors of expanded values with the structure of a linear subspace.

From now on, we consider an adversary who submits $2^d$, for a fixed value of $d$, messages to the MAC oracle (Note that if $d$ is logarithmic in the size of the modulus, then the attack still runs in polynomial time). Furthermore, let $\{M_1, \ldots, M_d\}$ denote $d$ random distinct messages and $U$ be the $\ell \times d$ matrix whose $i$-th row corresponds to $L(M_i)$. Finally, we denote by $U_i$ the $i$-th column of $U$.

We make the adversary compute all linear combinaison of the $M_i$'s and submit them to the MAC oracle. In other words, for every $d$-bit vectors $x$, the adversary submits messages of the form

$$M(x) = \bigoplus_j x_j M_j$$

to the MAC oracle and denote $R(x)$, the response obtained for the latter. Once all those responses are obtained, they are combined using an Hadamard-Walsh transform.

**Definition 5.1 (Hadamard-Walsh Transform)**
*Let $d$ be a positive integer and $\varphi : \mathbf{Z}_d \to \mathbf{R}$ be a function over the real numbers. Given a $d$-bit vector $V$, we define the function $\hat{\varphi}$, the Hadamard-Walsh (or multidimensional discrete Fourier) transform of $\varphi$, with respect to $V$ as*

$$\hat{\varphi}(V) = \sum_{x \in \mathbf{Z}_d} (-1)^{x \cdot V} \varphi(x).$$

In a particular case, we remark that $\hat{\varphi}(0) = \sum_{x \in \mathbf{Z}_d} \varphi(x)$.

Note that due to the linearity of the $L_i$'s, we can single out every bit of the linear combinaison of the $M(x)$'s in the following way

$$M_i(x) = (-1)^{L_i(M(x))} = (-1)^{\bigoplus_j x_j L_i(M_j)} = (-1)^{x \cdot U_i}$$

Table 5.1: Basic properties for computing $\hat{R}(V)$.

| $\varphi(x)$ | $\hat{\varphi}(V)$ |
|---|---|
| 1 | $2^d \times 1_{V=0}$ |
| $(-1)^{x_i U_i}$ | $2^d \times 1_{U_i=V}$ |
| $(-1)^{x_i (U_i \oplus U_j)}$ | $2^d \times 1_{U_i \oplus U_j = V}$ |

The linearity of the $L_i$'s can be further exploited to derive the properties listed in Table 5.1. From those properties, we can compute, from Equation (5.3), the Walsh-Hadamard transform of the function $R$

$$\begin{aligned}\hat{R}(V) =& \frac{1}{4} \sum_{i,i'} 2^{i+i'} \left( \sum_x (-1)^{x \cdot (U_i \oplus U_{i'} \oplus V)} \right) k_i k_{i'} \\ &- \frac{2^\ell - 1}{2} \sum_i 2^i \left( \sum_x (-1)^{x \cdot (U_i \oplus V)} \right) k_i \\ &+ \frac{(2^\ell - 1)^2}{4} \sum_x (-1)^{x \cdot V} \pmod{N}.\end{aligned} \tag{5.5}$$

When $N$ is a Mersenne number, i.e., $N = 2^\ell - 1$, then the last two terms of Equation 5.5 vanish so it simplifies to

$$\hat{R}(V) = \frac{1}{4}\sum_{i,i'} 2^{i+i'} \left(\sum_x (-1)^{x\cdot(U_i \oplus U_{i'} \oplus V)}\right) k_i k_{i'} \pmod{2^\ell - 1} \tag{5.6}$$

At this point, the best attack strategy would be to saturate the polynomial $\hat{R}$ so that the coefficients of all the monomials but one equal $0$. So, we have two options for the monomial to keep, either one of degree one, i.e., with unknown $k_i$, or a quadratic one, i.e., with unknown $k_ik_j$. Note that, due to Equation (5.6), this approach does not hold when $N$ is a Mersenne number. Hence, we need to take a specific treatment for that case.

### 5.4.1 *The Non-Mersenne Case*

In the general case, we have two options: The first is to manipulate Equation (5.5) such that the coefficients of all the quadratic monomials and all the monomials of degree one but one are 0. The other option would be to eliminate all the monomials expect one quadratic monomial. Once we simplify Equation (5.5) to any of the two forms, we obtain one linear equation in the key. Later, we present a tradeoff between the two approaches.

**Eliminating the $k_ik_j$ Monomials.** The first strategy is to select messages that eliminate all the quadratic monomials $k_ik_j$ and keep only one monomial of degree one. In light of the results listed in Table 5.1, this can be done by taking pairwise different $U_i$'s and set one of them to be the vector containing only 0's. With respect to the messages $M_1, \ldots, M_d$, this is equivalent to the conjonction of the two hypotheses

- $\forall j \in [\![1, d]\!], \exists I \in [\![0, \ell - 1]\!] : L_I(M_j) = 0$
- $\forall i, i' \in [\![0, \ell - 1]\!], \forall j \in [\![1, d]\!] : L_i(M_j) = L_{i'}(M_j) \implies i = i'$

Clearly, we can find these vectors by using an incremental algorithm to select $C_j$'s in the hyperplane defined by $L_I(C) = 0$. If we generate $d$ random vectors in the hyperplane, under heuristic assumptions, the probability that the condition is fulfilled is roughly $e^{-\ell^2 2^{-d-1}}$ which is constant for $d = 2\lceil \log_2 \ell \rceil$ and equal to $e^{-1/2}$.

We can use Equation (5.5), thanks to the hypotheses we obtain

$$\hat{R}(0) = 2^{d-2}\sum_i 2^{2i}k_i^2 - 2^{d+I-1}(2^\ell - 1)k_I + \frac{2^d(2^\ell - 1)^2}{4} \pmod{N}$$

but since $k_i^2 = 1$ for all $i$ we obtain

$$\hat{R}(0) = 2^{d-1}(2^\ell - 1)\left(\frac{2}{3}2^\ell - \frac{1}{3} - 2^I k_I\right) \pmod{N} \tag{5.7}$$

We can thus deduce $k_I$ when $N$ is not a Mersenne number. This means that recovering the key requires $O(r\ell^2)$ chosen challenges and complexity $O(r\ell^3)$. Clearly, we can trade data complexity against time complexity.

**Eliminating the Monomials of Degree One.** Another approach is to choose the challenges in a way that all the coefficients, except one of degree two, become 0. For this, we take values for $V$ that are different from the zero vector and all the $U_i$'s. Furthermore, if we construct the challenges in a way such that for every $U_i$, there exists a unique $U_j$ such that $U_j = U_i \oplus V$. In this scenario, Equation (5.5) simplifies to

$$\hat{R}(V) = 2^{I+J-1+d} k_I k_J \pmod{N} \tag{5.8}$$

so we can deduce the value of $k_I k_J$.

The advantage of this method over the first one is that from the same set of challenges we can derive many equations of the form $k_I k_J = b$ (which are indeed linear equations) for all $I$ and $J$ such that $V = U_I \oplus U_J$ satisfies the above conditions. With random $M_i$'s, the expected number of such equations is roughly $\frac{1}{2}\ell^2 e^{-\ell^2 2^{-d-1}}$ so for $d \approx 2\log_2 \ell$ we obtain enough equations to recover all bits of $K$ using $O(\ell^2)$ chosen challenges and complexity $O(\ell^3 \log \ell)$.

**Generalization** We can further generalize this attack by taking all values $V$ which are either 0 or equal to some $U_I$ or to some $U_I \oplus U_J$ but without requiring unicity of $I$ or $\{I, J\}$. In general, we obtain an equation which may involve several $k_I$ or $k_I k_J$ as Equation (5.5) simplifies to

$$\begin{aligned}\hat{R}(V) = & \sum_{\{I,J\}:U_I \oplus U_J = V} 2^{I+J-1+d} k_I k_J \\ & - \sum_{I:U_I = V} (2^\ell - 1) 2^{I+d-1} k_I \\ & + (2^\ell - 1)^2 2^{d-2} 1_{V=0} \pmod{N}.\end{aligned} \tag{5.9}$$

Provided that the number of monomials is not too large, the only correct $\pm 1$ assignment of the monomials leading to an expression matching the $\hat{R}(V)$ value can be isolated.

Using $d = \log_2 \frac{r(r+1)}{2}$ we obtain only one unknown per equation on average so we can recover all key bits with complexity $O(\ell r^2 \log r)$ using $O(r^2)$ chosen challenges. We can still slightly improve those asymptotic figures.

Let $\ell m$ be the complexity of getting the matching $\pm 1$ assignments in one equation (i.e. $m$ is the complexity in terms of modulo $N$ additions). The complexity of the Fourier transform is $O(\ell d 2^d)$, so the complexity of the algorithm becomes $O(\ell(d+m)2^d)$. The average number of unknowns per equation is $r^2 2^{-d-1}$. By using an exhaustive search strategy to solve the equation we obtain $\log_2 m \approx r^2 2^{-d-1}$. With $d = 2\log_2 r - \log_2 \log_2 \log r - 1$ we have

$m = \log r$ and we finally obtain a complexity of $O(\ell r^2 \log r / \log\log r)$ with $O(r^2/\log\log r)$ chosen challenges.

We could view the equation as a knapsack problem and use solving algorithms better than exhaustive search. For instance, we can split the equation in two halves and use a claw search algorithm. The effect of this strategy leads us to $\log_2 m \approx \frac{1}{2} r^2 2^{-d-1}$ and we reach the same asymptotic complexity.

### 5.4.2 *The Mersenne Case*

When $N$ is a Mersenne number, the expression of $R$ is only composed of quadratic terms. Following the same reasoning as in the general case, our strategy is to nullify all the coefficients of the monomials but one. Specifically, this translates into choosing the set $\{M_1, \ldots, M_d\}$ in a way that, with respect to the $U_i$'s, every value appears exactly twice. In other words, we require that

$$\forall I \in [\![1, \ell-1]\!], \exists! J \in [\![1, \ell-1]\!] : I \neq J \wedge U_I = U_J.$$

Under this assumption, we can derive from Equation (5.6)

$$\hat{R}(0) = 2^{I+J+d-1} k_I k_J \bmod N. \tag{5.10}$$

Hence, the same analysis developed for the case of saturating all the monomials of degree one applies.

**Generalization.** The main drawback of the latter method is that for the $2^d$ messages, the adversary can get only one equation for the key. In the following, we follow on the general strategy presented in Section  and integrate the results we obtained with the simplifications that using Mersenne numbers imply. Concretely, if we combine Equations (5.6) and (5.9), we obtain

$$\hat{R}(V) = \sum_{n=0}^{\ell-1} 2^n \sum_{\substack{\{I,J\}: U_I \oplus U_J = V, \\ I+J-1+d = n \bmod \ell}} k_I k_J \pmod N. \tag{5.11}$$

So, if the set of $\{I, J\}$'s sparsely spreads on the $(U_I \oplus U_J, (I+J-1+d) \bmod \ell)$ pairs, the knapsack is nearly super-increasing, i.e., that is, each element of the set is greater than the sum of all the numbers before it. So, we can directly *read* all $k_I k_J$ bits in the table of all $\hat{R}(V)$'s. With $d = 2\log_2 r - \log_2 \ell - 1$ we roughly have $\ell$ unknowns per equation and we can expect this phenomenon. So, we obtain a complexity of $O(r^2 \log r)$ with $O(r^2/\ell)$ chosen challenges. For instance, with $N = 2^{1\,277} - 1$ and $r = 128$ we can take $d = 3$ so that 8 chosen challenges are enough to recover all bits. With $r = \ell$ we can take $d = 10$ so that 1 024 chosen challenges are enough.

### 5.4.3 *Numerical Application*

SQUASH with no truncation is trivially broken if we can factor the modulus $N$ so it should be at least of $1\,024$ bits. As an example, for $r = \ell = 1\,024$ we can take $d = 14$ so roughly $2^d \approx 16\,000$ chosen challenges. We obtain at most $\frac{\ell(\ell+1)}{2}2^{-d} \approx 32$ unknowns per equation on average. We can then use a claw search algorithm that works with $2^{16}$ numbers in memory and $2^{16}$ iterations to recover $32$ bits of the key for each equation.

## 5.5 Handling Window Truncation

In what follows, we let $S$ denote the output from the Rabin function. We further recall Equation (5.1) that describes window truncation in SQUASH

$$T(x) = \left\lfloor \frac{x \bmod 2^b}{2^a} \right\rfloor .$$

It is clear that when $S$ is available to adversary, then the analysis from previous sections can be applied. Hence, we assume that the adversary only sees the final output SQUASH, i.e., he can query a MAC oracle for getting the MAC of chosen message.

Releasing a small part from the output of the Rabin function makes its inversion seemingly harder: it is not clear how, even by knowing the factorization, an adversary can reconstruct the missing bits. Consequently, this version of SQUASH was proposed with a very small modulus whose factorization could be easily computed (Recall that the concrete proposal of SQUASH was to use $N = 2^{128} - 1$).

### 5.5.1 *Handling the Truncation of the Combinaison of Many Integers*

Compared with the previous situation, we can no more combine the MAC values of the $M(x)$ but only their extractions $T(M(x))$. Unfortunately, the extraction of a combination of such integers does not coincide with the combination of the extractions because carries may propagate. However, when we sum a relatively small number of integers the overlap remains limited so that we can list all possible values. Indeed, for any $e_1, \ldots, e_q \in \mathsf{Z}_N$ we have

$$e_i \mod 2^b = 2^a T(e_i) + \alpha_i,$$

for an integer $\alpha_i \in [\![0, 2^a - 1]\!]$. Summing over all the $e_i$'s yields

$$\begin{aligned}\left(\sum_{i=1}^{q} e_i \mod N\right) \bmod 2^b &= \left(\sum_{i=1}^{q} e_i - \beta N\right) \bmod 2^b \\ &= \left(2^a \sum_{i=1}^{q} T(e_i) + \sum_{i=1}^{q} \alpha_i - \beta N\right) \bmod 2^{b-a},\end{aligned}$$

for an integer $\beta \in [\![0, q-1]\!]$. If we let $\alpha = T\left(\sum_{i=1}^{q} \alpha_i\right) \in [\![0, q-1]\!]$, we finally obtain

$$T\left(\sum_{i=1}^{q} e_i \mod N\right) = \sum_{i=1}^{q} T(e_i) + T(-\beta N) + \alpha \pmod{2^{b-a}} \qquad (5.12)$$

Although we do not know the value of $\alpha$ and $\beta$ when the complete $e_i$'s values are not revealed, it is still possible from Equation (5.12) to recover these values. In fact, since there are only $q^2$ possible pairs while the right-hand side, like the first term of the left-hand side, can take $2^{b-a}$ different values. By construction, we are ensured that the correct pair $(\alpha, \beta)$ is unique. The other ones can be considered to be random. So, as long as $2 \times 2^{b-a} \geq q^2$, we can build a table of all possible values of $T(-\beta N) + \alpha$ to single out the correct assignment for $\alpha$ and $\beta$ with probability $1/2$.

We note that the result above holds when we consider the alternate sum of the $e_i$'s. In other words, we can show that, for $v_1, \ldots, v_q \in \{-1, 1\}$ such that $\sum_{i=0}^{q} v_i = 0$, we have

$$T\left(\sum_{i=1}^{q} v_i e_i \mod N\right) = \sum_{i=1}^{q} (v_i T(e_i)) + T(-\beta N) + \alpha \pmod{2^{b-a}} \qquad (5.13)$$

where $\alpha \in [\![-1 - q/2, q/2]\!]$ and $\beta \in [\![-1 - q/2, 1 + q/2]\!]$.

**The Mersenne Case.** We further notice that when $N$ is a Mersenne number, we readily have $N \equiv 1 \pmod{2^b}$. Hence, we have the simplification

$$\left(\sum_{i=1}^{q} e_i \mod 2^\ell - 1\right) \mod 2^b = \left(\sum_{i=1}^{q} e_i - \beta\right) \mod 2^b$$

for an integer $\beta \in [\![0, q-1]\!]$. Then, we can write

$$T\left(\sum_{i=1}^{q} e_i \mod 2^\ell - 1\right) = \sum_{i=1}^{q} T(e_i) + T(\beta) + \alpha \pmod{2^{b-a}} \qquad (5.14)$$

The nice property of this expression is that if $q \leq 2^a$ then we always have $T(\beta) = 0$. In this case, the right-hand side of the equation can only take $q$ values. In the other case, $T(\beta)$ is an integer of $q - 2^a$ bits. It can be integrated in the $\alpha$ in $T(-\beta N) + \alpha$ in other cases: all $T(-\beta N) + \alpha$ values are numbers in the $[\![0, 2^d + \left\lfloor \frac{2^d - 1}{2^a} \right\rfloor]\!]$ range. Consequently, assuming that $q \leq 2^a$, Equation (5.14) further simplifies to

$$T\left(\sum_{i=1}^{q} e_i \mod 2^\ell - 1\right) = \sum_{i=1}^{q} T(e_i) + \alpha \pmod{2^{b-a}} \qquad (5.15)$$

This result can also be generalized to the case of a combinaison of addition and subtraction of truncated values. Starting from Equation (5.13), we note that if $q < 2^a$ then the expression $T(-\beta N)$ simplifies to either 0, when $\beta$ is positive or equal to 0, or $2^{b-a} - 1$, when $\beta$ is negative. Hence, the sum $T(-\beta N) + \alpha$ ranges in the interval $\alpha \in [\![-q/2, q/2]\!]$ instead of $[\![-q, q]\!]$ for the general case.

### 5.5.2 *Adapting the Attack on SQUASH-0*

Equations (5.12) and (5.13) provide us with a mean to link the bits from the combinaison of some integers and their truncation. Hence, we can almost readily adapt the analysis of Section 5.4 with $q = 2^d$.

Let us first consider the first attack of Section 5.4.1, namely the one in which the adversary sums over all the $R(x)$'s. We now apply the previous attack (first method) with $n = 2^d$ and the list of all $d$-bit vectors $x$ and set $e_i = S(x)$ corresponding to the challenge $M(x)$. Recall that, under the appropriate assumptions on the messages submitted to the MAC oracle, Equation (5.7) describes the relation between the $I$-th key bit and the sum of the outputs from the Rabin function. Adapting the notation, this equation rewrites as

$$\hat{S}_I(0) = 2^{d-1}(2^\ell - 1)\left(\frac{2}{3}2^\ell - \frac{1}{3} - 2^I k_I\right) \pmod N$$

On the other hand, we have

$$\begin{aligned} T\left(\hat{S}_I(0) \bmod N\right) &= T\left(\sum_{x\in\mathbf{Z}_d} R(x) \bmod N\right) \\ &= \left(\sum_{x\in\mathbf{Z}_d} T(R(x)) + T(-\beta N) + \alpha\right) \bmod 2^{b-a} \\ &= \left(\hat{R}(0) + T(-\beta N) + \alpha\right) \bmod 2^{b-a}. \end{aligned}$$

Hence,

$$T\left(2^{d-1}(2^\ell - 1)\left(\frac{2}{3}2^\ell - \frac{1}{3} - 2^I k_I\right) \bmod N\right) = \left(\hat{R}(0) + T(-\beta N) + \alpha\right) \bmod 2^{b-a}.$$

Here, the pair $(\alpha, \beta)$ can take up $2^{2d}$ values which can be filtered by the $r$-bit value of the left-hand side (recall that $r = b - a$). Hence, the probability that there exists $\alpha$ and $\beta$ such that $T(\hat{S}_I(0))$ matches the right-hand side of the equation is at most $2^{2d-r}$, so for $2d + 1 < r$ it is likely that we can deduce $k_I$. The complexity of the attack in terms of queries remains unchanged whereas the computational complexity is augmented by the cost of building a table of values for $T(-\beta N) + \alpha$.

The second method of Section 5.4.1 in which the adversary saturates the monomials of degree one can also be adapted as follows. Keeping the same assumptions on the $M_i$'s and the vector $V$, Equation (5.8) rewrites as

$$\hat{S}(V) = 2^{I+J-1+d} k_I k_J \pmod N.$$

Again, we can use Equation (5.12) to yield

$$T(2^{I+J-1+d} k_I k_J) = \left(\hat{R}(V) + T(-\beta N) + \alpha\right) \quad \bmod 2^{b-a}$$

Thus, as long as $2d + 1 < r$, we can deduce the value of $k_I k_J$. Again, the complexity of the attack is the same as before in number of queries and slightly overheaded in time for computing the table of values of $T(-\beta N) + \alpha$.

**The Mersenne case.** In the case where $N$ is a Mersenne number, we need to make a specific treatment. Updating the notations of Equation (5.10) under the same assumptions yields

$$\hat{S}_{I,J}(0) = 2^{I+J+d-1} k_I k_J \bmod 2^\ell - 1.$$

Combining this last expression with Equation (5.14), we obtain

$$T(\hat{S}_{I,J}(0)) = \left(\hat{R}(0) + \alpha\right) \bmod 2^{b-a}$$

for some $\alpha$ in the $[\![0, 2^d - 1]\!]$ range. Let $\hat{S}^+_{I,J}(0)$ and $\hat{S}^-_{I,J}(0)$ denote the value of $\hat{S}_{I,J}(0)$ when $k_I k_J$ is equal to $+1$ and $-1$ respectively. Note that $T(\hat{S}^-_{I,J}(0)) + T(\hat{S}^+_{I,J}(0)) = 2^{b-a} - 1$, in other words $T(\hat{S}^-_{I,J}(0))$ and $T(\hat{S}^+_{I,J}(0))$ have all their bits inverted. Furthermore,

$$\begin{aligned} T(\hat{S}^+_{I,J}(0)) &= T\left(2^{(I+J+d-1) \bmod \ell}\right) \\ &= \begin{cases} 2^{((I+J+d-1) \bmod \ell)-a} & \text{if} \quad (I+J+d) \bmod \ell \in [\![a+1, b]\!] \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

This is enough to deduce $k_I k_J$ for $(I, J)$ pairs such that there is no $\alpha$ for which $T(\hat{S}^+_{I,J}(0))$ matches the right-hand side. Thus we can recover $k_I k_J$.

### 5.5.3 *Generalization*

As we proceeded in Section 5.4, we can generalize the attack to take any combinaison of the MAC responses. In general, for all $V$ there exists $\alpha$ and $\beta$ such that

$$\begin{aligned} & T\left(\left(\sum_{\{I,J\}:U_I \oplus U_J = V} 2^{I+J-1+d} k_I k_J \right.\right. \\ & \qquad \left.\left. - \sum_{I:U_I=V} (2^\ell - 1) 2^{I+d-1} k_I + (2^\ell - 1)^2 2^{d-2} 1_{V=0}\right) \bmod N\right) \\ = & \left(\hat{R}(V) + T(-\beta N) + \alpha\right) \bmod 2^{b-a} \end{aligned}$$

with

$$\begin{cases} \alpha \in [\![0, 2^d]\!], \quad \beta \in [\![0, 2^d - 1]\!] & \text{if} \quad V = 0 \\ \alpha \in [\![-1 - 2^{d-1}, 2^{d-1}]\!], \quad \beta \in [\![-2^{d-1} + 1, 2^{d-1} - 1]\!] & \text{if} \quad V \neq 0 \end{cases}$$

Our attack strategy can now be summarized as follows.

1. Take a value for $d$. Make a table of all $T(-\beta N) + \alpha$ values. This table has less than $2^{2d}$ terms, and exactly $2^d + 1$ terms in the Mersenne case, and can be compressed by dropping the $d$ least significant bits corresponding to the $\alpha$ part. In the Mersenne case, it can be compressed to nothing as numbers of form $T(-\beta N)+\alpha$ are all in the interval $[\![-2^{d-1}, 2^{d-1}]\!]$ modulo $2^{b-a}$.
2. Pick $d$ challenges at random and query all the $2^d$ combinations $C(x)$. Get the responses $R(x)$.
3. Compute the discrete Fourier transform $\hat{R}$ in $O(\ell d 2^d)$.
4. For each $V$, try all $\pm 1$ assignments of occurring unknowns in $\hat{S}(V)$ and keep all those such that $T(\hat{S}(V) \bmod N) - \hat{R}(V)$ matches an entry in the table of $T(-\beta N) + \alpha$.

Again, this attack uses $O(2^d)$ chosen challenges and a complexity of $O(\ell(d + 2^{s2^{-d}})2^d)$ where $s$ is the number of unknowns, i.e. $s = \frac{r(r+1)}{2}$ resp. $s = \frac{r(r-1)}{2}$ in the Mersenne case. The remaining question is whether all wrong assignments are discarded.

For a given equation, each of the $2^{s2^{-d}}$ wrong assignments is discarded with probability $2^{2d-(b-a)}$ resp. $2^{d-(b-a)}$. Thus, if $b - a > 2d + s2^{-d}$ resp. $b - a > d + s2^{-d}$ they can all be filtered out. The minimum of the right-hand side is $2\log_2 s + 2\log_2 \frac{e \ln 2}{2}$ resp. $\log_2 s + \log_2(e \ln 2)$ and reached by $d = \log_2 s + \log_2 \frac{\ln 2}{2}$ resp. $d = \log_2 s + \log_2 \ln 2$. By taking this respective value for $d$ we have $O(r^2)$ chosen challenges and a complexity of $O(\ell r^2 \log r)$, and the condition becomes $b-a > 4\log_2 r + 2\log_2 \frac{e \ln 2}{2} - 2$ resp. $b-a > 2\log_2 r + \log_2(2e \ln 2)$. If $b - a > 4\log_2 r - 2$ resp. $b - a > 2\log_2 r$ this condition is always satisfied.

**The Mersenne case.** Finally, the Mersenne case can simplify further using Equation (5.11). We take $d = 2\log_2 r - \log_2 \ell - 1$ and run the attack with $O(r^2/\ell)$ chosen challenges and complexity $O(r^2 \log r)$. Assuming that all unknowns $k_I k_J$ sparsely spread on $(U_I \oplus U_J, (I + J - 1 + d) \bmod \ell)$ pairs then $T(\hat{R}(V) \bmod N)$ yields $b - a - d$ useful bits with roughly one $k_I k_J$ per bit and ends with $d$ garbage bits coming from $T(-\beta N) + \alpha$. So, we can directly *read* the bits through the window and it works assuming that $b - a > d$, which reads $b - a > 2\log_2 r - \log_2 \ell - 1$.

**Application to SQUASH-128 with Linear Mapping** We now use the recommended parameters by Shamir: $\ell = 128$, $N = 2^{128} - 1$, $a = 48$, $b = 80$ and plug them into SQUASH-0. Although Shamir suggested to use a 64-bit secret key with non-linear mixing, we assume here that the mixing is of the form $f = g \oplus L$ with linear $L$ but that $g$ expands to $r = 128$ secret bits (possibly non-linearly). We have $s = 8\,128$ unknowns of form $k_i k_{i'}$. With $d = 10$ we obtain $1\,024$ vectors $V$ so we can expect to find 8 unknowns in each equation. Equations are of form

$$T(\hat{S}(V) \bmod N) = \left(\hat{R}(V) + T(-\beta N) + \alpha\right) \bmod 2^{b-a}$$

where $(T(-\beta N)+\alpha) \bmod 2^{b-a}$ is in the range $[-2^9, 2^9]$ which gives a set of at most $2^{10}+1$. Filtering the $2^8 - 1$ wrong assignments on the 8 unknowns we can expect $2^{-13}$ false acceptances in addition to the right one. Simple consistency checks can discard wrong assignments, if any, and recover all $k_i$'s. Clearly, all computations are pretty simple and we only used $2^{10}$ chosen challenges.

Using the final trick in the Mersenne case we use $d = 6$ and thus $64$ chosen challenges to get $64$ equations which yield $26$ bits each.

With $N = 2^{1\,277} - 1$ and the worst case $\ell = r$, i.e., the mixing function is not expanding the key, the attack works for $b - a \geq 21$ and we can take $d = 19$. We request for $2^{19}$ chosen challenges. We obtain $2^{19}$ equations with roughly $1.6$ unknowns per equation.

By using the final trick we take $d = 10$. The $T(-\beta N) + \alpha$ part wastes $10$ bits from the window and we can expect to have a single unknown per remaining bit so that we can simply read it through the window. Provided that the window has at least $32$ bits we expect to read $22$ bits in each of the $1\,024$ equations so we can recover all bits.

## 5.6 Extending to Non-linear Mappings

In case the mapping $L$ is a (non-linear) *permutation*, we can adapt our attack strategy by choosing the challenges as follow

- pick $d$ challenges $C_1, \ldots, C_d$.
- compute the chosen challenges by $C^\star(x) = L^{-1}\left(\bigoplus_j x_j L(C_j)\right)$.

By using,

$$c_i^\star(x) = (-1)^{L_i(C^\star(x))} = (-1)^{\bigoplus_j x_j L_i(C_j)} = (-1)^{x \cdot U_i}$$

Equation (5.5) remains unchanged so that we can still apply all the attacks described through Sections 5.4 and 5.5. More generally, we can extend these attacks to *any* mixing function of form $f(K, C) = g(K) \oplus L(C)$ as long as we can find vector spaces of dimension $d$ in the range of $L$.

## 5.7 Conclusion

One argument for motivating the SQUASH algorithm consisted of playing the "blame game": if anyone could break SQUASH, then the Rabin cryptosystem is the one which should be blamed instead of the SQUASH design. Clearly, our attack demonstrates that this argument is not correct. There are instances of the SQUASH algorithm which can be broken although we still have no clue how to factor integers. Indeed, our method translates into a "known random coins attack" against Rabin-SAEP which leads to a plaintext recovery. Known random coins attacks are not relevant for public-key cryptosystems although they are in the way SQUASH is using it.

So, although the “blame game” argument is not valid, the security of SQUASH is still an open problem.

# Part II

## PRIVACY IN RFID PROTOCOLS

# 6

# PRIVACY FAILURES IN RFID PROTOCOLS

Contents

This chapter mainly serves as a motivation for the upcoming ones: We exhibit privacy attacks on several RFID protocols. Namely, we show that ProbIP [CS07], MARP [KYK06], Auth2 [TSL07], YA-TRAP+ [LBdM06], O-TRAP [LBdM06], and RIPP-FS [CPMS07] all fail to protect the privacy of the tag's holders.

The adversaries we consider in this chapter are assumed to not be able to tamper with tags. In other words, we consider adversaries who are able to interact with readers and tags and have control over the communication link. Since RFID tags do communicate through an unprotected wireless channel, it is appropriate to assume such abilities for an attacker.

Before that, we introduce a simple ad-hoc privacy model that will be employed to show the privacy shortcomings of the protocols mentioned above.

This chapter includes results that were earlier published in the proceedings of two conferences, the first one in ISPEC 2008 [OP08a] and the second one in ACNS 2008 [OP08b].

## 6.1 An ad-hoc Privacy Model

We will later deal with the problem of building a global model capturing the notion of privacy. For now, we will consider a simple model, in some ways equivalent to the one of Juels and Weis [JW07], with some differences essentially lying in the constraints put on the adversary.

Although it is not its goal, we will use that model to capture the basic notions of untraceability and anonymity of RFID tags.

Similarly to Juels and Weis, we capture the notion of privacy as the inability for any adversary to infer the identity of a tag chosen from a pair she has chosen. Concretely, after interacting with the RFID system, the adversary is asked to select two RFID tags and receives one of them. Her goal is then to discover the identity of the received tag. For that, she is still allowed to interact with the system and the target tag. In the end, we consider that the adversary has defeated the privacy of the scheme if her guess for the correct identity of the tag is true with a probability significantly greater than the one of when she outputs a random guess.

We stress again that we neither aim to propose this definition as a privacy model nor claim novelty of our definition. Instead, we will use this model exclusively in this chapter and the following one for the analysis of the privacy and security issues of recent RFID protocols. In fact, the model defined herein can be seen as an alternative definition of the model of Juels-Weis [JW07] with some differences, e. g., in the constraints put on the adversary (see the discussion in Section 6.5.1) in a style that is more in line with the model of Bellare, Pointcheval, and Rogaway [BPR00] for password-based authenticated key exchange (AKE) protocols. The reason for borrowing the formalism of AKE protocols is mainly due to the close relationship these latter enjoy with RFID protocols. Indeed, the goal of both primitives is for a party to authenticate himself to another one with whom it shares some partially secret bits. Examples of such shared data include a public key, a password, or an encryption key. Moreover, AKE protocols often run in an asymmetric scenario: the verifier of the protocol may be a resourceful server while the prover is a client with limited capabilities. In such a case, it is often assumed that the prover, i. e., the tag in RFID systems, is corruptible while the most powerful entity is resilient to corruption. In other words, an adversary may be able to obtain the secret held by the weaker party by tampering. We further follow on a common assumption and limit the model to one RFID reader that has an inner up to date copy of the database.

We define an RFID scheme as a polynomial-time two party authentication protocol between a tag $T_{\mathsf{ID}}$ and a reader $R$. While we assume each tag $T_{\mathsf{ID}}$ to hold a secret $K_{\mathsf{ID}}$, the database that the reader accesses contains all the tags' secrets, i.e., it is a table of the form $(\mathsf{ID}_j, K_j)$. After running a protocol instance, the reader outputs either $\mathsf{Accept}(\mathsf{ID}_j)$ if it authenticates a tag whose $\mathsf{ID}$ is listed in the database , or $\mathsf{Reject}$ otherwise. Conversely, in the case of mutual authentication, i. e., when the reader is also required to authenticate itself, the partner tag outputs $\mathsf{Accept}(\mathcal{R}_j)$ in case of success and $\mathsf{Reject}$ otherwise. Hereafter, we formally define our notions of partnership and session completion.

**Definition 6.1 (Partnership & Session Completion)**
*We say that a reader instance $\mathcal{R}_j$ and a tag instance $\mathcal{T}_i$ are partners if, and only if, both have output $\mathsf{Accept}(\mathcal{T}_i)$ and $\mathsf{Accept}(\mathcal{R}_j)$ respectively, signifying the completion of the protocol session.*

An adversary $\mathcal{A}$ is a malicious entity, modeled as a probabilistic polynomial-time algorithm, who controls all the communications between readers and tags and interacts with them as defined by the protocol. Concretely, the adversary interface with the RFID system through the following oracles.

- $\mathsf{Execute}(\mathcal{R}, \mathcal{T}, i)$ **query.** This oracle models *passive* attacks, i.e., the ability for an adversary to eavesdrop on a protocol instance. As such, it triggers a full protocol instance with identifier $i$ between the reader $\mathcal{R}$ and the tag $\mathcal{T}$ and returns its transcript to the adversary.
- $\mathsf{Send}(U_1, U_2, i, m)$ **query.** This query models *active* attacks by allowing the adversary $\mathcal{A}$ to impersonate a reader $U_1 \in Readers$ (resp. a tag $U_1 \in Tags$) in some protocol session $i$ and send a message $m$ of its choice to an instance of a tag $U_2 \in Tags$ (resp. a reader $U_2 \in Readers$). This query subsumes the TagInit and ReaderInit queries as well as challenge and response messages in the Juels-Weis model.
- $\mathsf{Corrupt}(\mathcal{T}, K)$ **query.** This query allows the adversary $\mathcal{A}$ to learn the stored secret $K'$ of the tag $\mathcal{T} \in Tags$, and which further sets the stored secret to $K$. It captures the notion of *forward security* or *forward privacy* and the extent of the damage caused by the compromise of the tag's stored secret. This is the analog of the SetKey query of the Juels-Weis model.
- $\mathsf{Test}_{\mathsf{UPriv}}(U, i)$ **query.** This query is the only query that does not correspond to any of $\mathcal{A}$'s abilities or any real-world event. This query allows to define the indistinguishability-based notion of *untraceable privacy* ($\mathsf{UPriv}$). If the party has accepted and is being asked a $\mathsf{Test}$ query, then depending on a randomly chosen bit $b \in \{0, 1\}$, $\mathcal{A}$ is given $\mathcal{T}_b$ from

the set $\{\mathcal{T}_0, \mathcal{T}_1\}$. Informally, $\mathcal{A}$ succeeds if it can guess the bit $b$. In order for the notion to be meaningful, we restrict the adversary to perform Test queries on sessions that terminated correctly without any party being corrupted. Such a session is said to be fresh and its formal definition is given hereafter.

**Definition 6.2 (Freshness)**
*A party instance is fresh at the end of execution if, and only if,*

1. *it has output* Accept *with or without a partner instance,*
2. *both the instance and its partner instance (if such a partner exists) have not been sent a* Corrupt *query.*

**Definition 6.3 (Untraceable Privacy (UPriv))**
*Untraceable privacy (*UPriv*) is defined using the game $\mathcal{G}$ played between a malicious adversary $\mathcal{A}$ and a collection of reader and tag instances. $\mathcal{A}$ runs the game $\mathcal{G}$ whose setting is as follows.*

1. ***Phase 1 (Learning):*** *$\mathcal{A}$ is able to send any* Execute, Send, *and* Corrupt *queries at will.*
2. ***Phase 2 (Challenge):***
   1. *At some point during $\mathcal{G}$, $\mathcal{A}$ will choose a fresh session on which to be tested and send a* Test *query corresponding to the test session. Note that the test session chosen must be fresh in the sense of Definition 6.2. Depending on a randomly chosen bit $b \in \{0, 1\}$, $\mathcal{A}$ is given a tag $\mathcal{T}_b$ from the set $\{\mathcal{T}_0, \mathcal{T}_1\}$.*
   2. *$\mathcal{A}$ continues making any* Execute, Send, *and* Corrupt *queries at will, subjected to the restrictions that the definition of freshness described in Definition 6.2 is not violated.*

   ***Phase 3 (Guess):*** *Eventually, $\mathcal{A}$ terminates the game simulation and outputs a bit $b'$, which is its guess of the value of $b$.*

*The success of $\mathcal{A}$ in winning $\mathcal{G}$ and thus breaking the notion of* UPriv *is quantified in terms of $\mathcal{A}$'s advantage in distinguishing whether $\mathcal{A}$ received $\mathcal{T}_0$ or $\mathcal{T}_1$, i.e. it correctly guesses $b$. This is denoted by $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{UPriv}}(k)$ where $k$ is the security parameter.*

The LBdM model [LBdM07] similarly allows the corruption of tags. Nevertheless, proofs of security are set in Canetti's universal composability (UC) framework [Can00].

Vaudenay's model [Vau06, Vau07] is stronger than both the Juels-Weis and Le-Burmester-de Medeiros models in terms of the adversary's corruption ability. In more detail, it is stronger than the Juels-Weis model in the sense that it allows corruption even of the two tags used in the challenge phase. It is stronger than the Le-Burmester-de Medeiros model in the sense that it considers all its privacy notions even for corrupted tags, in contrast to the Le-Burmester-de Medeiros model that only considers corruption for its forward privacy notion.

Our choice to describe our tracing attacks in later sections with reference to a defined model is for more uniformity between similar attacks on different RFID protocols, and for better clarity to illustrate how an adversary can circumvent the protocols using precise types of interactions that she exploits, as captured by her oracle queries. This will facilitate the task of a designer when an attempt is made to redesign an attacked protocol.

| **Tag** | | **System** |
|---|---|---|
| **State:** $K_{\mathsf{ID}}$ | | **Database:** $\{\dots,(\mathsf{ID},K_{\mathsf{ID}}),\dots\}$ |
| For | $\xleftarrow{\text{HELLO}}$ | |
| $(a_1,b_1)\dots,(a_P,b_P)\in_R\{0,1\}^{k+\ell}$ s.t. | | |
| $\forall i\in[\![1,P]\!]:\mathsf{Hwt}(K_{\downarrow a_i}\oplus b_i)=\frac{\ell}{2}$ | | |
| | $\xrightarrow{a_1,b_1,\dots,a_\ell,b_\ell}$ | Find $(\mathsf{ID},K_{\mathsf{ID}})$ s.t. |
| | | $K_{\mathsf{ID}}$ satisfies all the equations |

**Figure 6.1:** The ProIP protocol

## 6.2 ProbIP

### 6.2.1 *ProbIP and the SAT Problem*

At RFIDSec '07, Castellucia and Soos [CS07] proposed an RFID protocol (ProbIP) that allows tag identification by legitimate readers. Its security is based on the SAT problem. A SAT instance is defined by a propositional logic formula written in conjonctive normal form, i. e., the AND of several literals, which are, in their turn, written in disjonctive normal form,i. e., as the combinaison of OR and NOT of boolean variables. An example of a SAT instance is given below.

$$(x_1 \vee x_2 \vee \neg x_5) \wedge (\neg x_2 \vee x_3 \vee x_4) \wedge (\neg x_1 \vee \neg x_3 \vee x_4).$$

Now, given a SAT instance, the associated decisional SAT problem is to determine whether there exists an assignment for the boolean variables such that the formula evaluates to True. The converse computational problem is to find this solution, if it exists. A similar problem, the $\ell/2$-in-$\ell$ SAT problem, is to determine whether there exists, from $L$ variables, a truth assignment to those variables so that each clause has exactly $\ell/2$ true literals.

This problem is famous for being the first one to be proven to lie in the class of complexity $\mathcal{NP}$-Complete in the seminal paper of Cook [Coo71]. However, $\mathcal{NP}$-hardness treats the complexity of solving *any* instance of a decisional problem. In other words, it only considers the worst-case instances of a problem. Thus, when constructing a cryptographic primitive it is crucial to ensure that the instances of the $\mathcal{NP}$-Complete problem that are generated are indeed "hard" to solve. Several cryptosystems based on $\mathcal{NP}$-Complete problems were broken just because the generated instances were in fact "easy" to solve. For concrete examples, we refer the interested reader to [Sha84] and [Vau98].

As it is depicted in Figure 6.1, the core idea of ProbIP is to make the tag generate instances of the $\ell/2$-in-$\ell$ SAT problem. For that, each tag is given a $k$-bit secret key $K$ and the reader is given access to the list of all secrets. The protocol starts by a Hello message from the reader that initiates a protocol instance. To compute its answer, the tag generates a pair of vectors $(a, b)$ such that $a$ is a $k$-bit vector whose Hamming weight is equal to $\ell$ and $b$ is an

$\ell$-bit vector. Besides this, we let $K_{\downarrow a}$ denote the $\ell$-bit vector which contains the bits of $K_{\mathsf{ID}}$ in positions corresponding to the positions of all the elements of $a$ equal to 1. We further restrict the Hamming weight of the $\ell$-bit vector $K_{\downarrow a} \oplus b$ to be equal to $\ell/2$, i.e., it has exactly $\ell/2$ bits equal to 1. For a complete authentication round, the tag repeats this operation $P$ times. In other words, it generates $P$ pairs, $(a_1, b_1), \ldots, (a_P, b_P)$ that satisfy the above conditions. Hence, the output of one authentication session for the tag is an (under-defined) linear system of equations of the form.

$$\begin{cases} \sum_{i=1}^{L}(K_{a_i^1} \oplus b_i^1) = \frac{L}{2} \\ \sum_{i=1}^{L}(K_{a_i^2} \oplus b_i^2) = \frac{L}{2} \\ \cdots\cdots \\ \sum_{i=1}^{L}(K_{a_i^P} \oplus b_i^P) = \frac{L}{2} \end{cases}$$

To recover the identity of the tag, the reader goes through its list of secrets and tests which one of them satisfy all the equations. In the end, the tag whose secret solves all equation is accepted as the partner tag. We note that this operation is more efficient if instead of testing all equations at once for every key, each equation could act as a filter: the reader first keeps all keys that satisfy the first equation, then tests them on the second one and so on. Indeed, the whole complexity decreases from $Pn/2$ to $s$.

Depending on the parameter set, it may be that a key different from the one held by a tag satisfies all the equations and be recognized as the partner tag. This event is commonly known to as a false positive. To compute the probability of false positives occurring, one has to look at the number of equations for which a random but fixed key can be a solution versus the total number of equations. When the RFID system consists of $n$ tags, Castellucia and Soos showed that this probability is given by

$$P_{\mathsf{FA}} = n \left( \frac{\binom{k}{\ell/2}\binom{k-\ell/2}{\ell/2}}{\binom{2k}{\ell}} \right)^P$$

From this probability, we can derive the number of equations $P$ that a tag has to provide the reader to authenticate itself. However, for a security point of view, there is still an upper-bound for $P$ above which the $\ell/2$-in-$\ell$ SAT problem becomes easier to solve. Nevertheless, having a to small $P$ may induce a high false acceptance rate, which harms the correctness of the whole scheme. Hence, it is crucial to find a balance between security and efficiency. In order to measure the increasing difficulty of the problem when $P$ changes and determine parameter sets, the authors of ProbIP proposed to use a SAT solver, called Minisat, to tentatively solve a $\ell/2$-in-$\ell$ SAT problem with $P$ equations. Unfortunately, no concrete parameter set was suggested.

The security of the scheme was analyzed under the Juels-Weis model. As the adversary selects two tags and is given one of them, chosen randomly, she has to guess the real identity of the latter with a non-negligible probability, i.e., significantly larger than $1$ (see Chapter 7 for a complete description of the Juels-Weis model). For that, the adversary needs to interact with the target tag and will ultimately need to decide from which secret was an $\ell/2$-in-$\ell$ SAT instance generated. Since this problem reduces to the decisional $\ell/2$-in-$\ell$ SAT problem, any successful attack on ProbIP leads to an efficient solver of the $\ell/2$-in-$\ell$ SAT problem.

### 6.2.2 *Violation of Anonymous Privacy*

Before submitting the two tags to the challenger, the Juels-Weis model allows the adversary to interact with all the tags. Namely, the adversary can query the two target ones as many times as she wishes. This is even more easy to carry out when the tag does not authenticate its partner as it is the case in ProbIP. In the following, we show that these interactions lead to the recovery of the tag's secret, thus violating both its security and privacy.

In short, an adversary could just query the tag until she ends up with enough equations. At this point, it becomes useless to hand the system to a SAT solver since a Gaussian elimination type algorithm would be able to recover the key in polynomial time. More formally, the attack runs as follow. We consider an RFID system with two RFID tags, $T_0$ and $T_1$. We make the adversary send HELLO messages to each of the two tags via Send queries to the tag until she gets $\ell$ equations. Since each request generates $P$ equations, an adversary would need to query the tag $n/P$ times. After that, she obtains the following system in which $v_i^j$ denotes a boolean variable that is set to 1 if the $i$-th bit of $K$ is present in the $j$-th equation

$$\begin{cases} \sum_{i=1}^{L} v_i^1(K[i] \oplus b_i^1) & = \frac{L}{2} \\ \sum_{i=1}^{L} v_i^2(K[i] \oplus b_i^2) & = \frac{L}{2} \\ \dots \\ \sum_{i=1}^{L} v_i^\ell(K[i] \oplus b_i^\ell) & = \frac{L}{2} \end{cases} \tag{6.1}$$

As for any boolean $v$ we can write $v + \bar{v} = 1$, we replace any $\overline{K[i]}$ by the value $1 - K[i]$. There are as many as $3^n$ possible equations as the coefficients of each variable $K[i]$ take three values: $0, 1, -1$.

This way, the adversary gets a linear system of $n$ equations and $n$ variables that can be solved using standard methods such as the Gaussian elimination method. In the case where the $n$ equations are not linearly independant, the adversary can still obtain more equations from the tag by sending HELLO messages until she gets enough equations.

### 6.2.3 *Future Development*

The weakness of this authentication protocol comes from the fact that at each round the adversary gets some information from the same key. So a quick way to counter the attack would be to include a key-updating mechanism similar to OSK [OSK05] at the end of the protocol using a one-way function.

Another approach, recently taken by Kiltz et al. [KPC⁺11] was to randomize the tag's response by having some of the equations erroneous with some probability $\eta$. A discussion of this scheme can be read in Chapter 4.

## 6.3 MARP

### 6.3.1 *Description*

Starting from the observation that RFID tags do not support expensive computations that are quasi-mandatory to achieve security and privacy, Kim et al. [KYK06] considered the use of a third party acting between the reader and the tags, a mobile agent for RFID privacy abbreviated MARP hereafter. In practice, the role of the MARP can be played by a PDA or a mobile phone. The idea of Kim et al. was to bind a tag to a MARP so that it is the latter who authenticates to the reader on behalf of the tag. For that, the scheme they proposed is composed of three sub-protocols. At first, each tag is given a PIN that can be used to unlock it. A copy of that PIN is also stored in the database. The first sub-protocol, called the initial setup phase, is used to transfer that PIN authentication capabilities of a tag to a designated MARP: at the end of the protocol, the MARP learns a secret, associated to the tag's PIN, that allows it to acts on behalf of the latter. Concretely, this operation is supposed to represent a transfer of ownership. This operation typically happens when an item is bought in a store and the client's MARP registers the PIN of the tag attached to the product.

Once the secret information of the tag is stored in the MARP, the tag is put into sleep mode. This is called the privacy preserving phase as it allows the MARP to act on behalf of the tag. It is also the most typical mode of the proposed scheme as data communication occurs only between a MARP and the reader. Another mode, called authentication mode, is also proposed for when the reader wants to ensure that a MARP is effectively paired with a tag as it claims. As MARPs only learn the hash of the tags' keys, the protocol consists of the reader sending an encrypted challenge to the MARP. The latter decrypts it and forwards it to the tag who hashes its XOR with the key. Finally, that hash value is sent back to MARP who encrypts it and forwards it to the reader. A mathematical description of the scheme is depicted in Figure 6.2. To avoid confusion with the next protocol, we shall name this protocol MARP-1.

Another protocol, which we refer to as MARP-2, and does not feature all those different modes was also proposed by the authors of MARP-1. Instead, it allows to have a double authentication of MARP and a tag at once. The first is authenticated using its key pair and the information it has received from the tag during the initialization phase while the second uses its PIN. The detailled steps of this protocol are depicted in Figure 6.3.

It is worth mentioning that in both protocols MARP-1 and MARP-2 all communication

| **Reader** ($\mathsf{ID}_g$) | | **MARP** | | **Tag** |
|---|---|---|---|---|
| **Key Pair:** $(sk_g, pk_g)$ | | **Key pair:** $(sk_m, pk_m)$ | | **Secrets:** $\mathsf{ID}, \mathsf{PIN}_{\mathsf{ID}}, K_{\mathsf{ID}}$ |
| | | **Initialization Phase** | | |
| | $\xrightarrow{\mathsf{PIN}_{\mathsf{ID}}}$ | Store $\mathsf{PIN}_{\mathsf{ID}}$ | | |
| | | $h_{\mathsf{PIN}} \leftarrow h(\mathsf{PIN}_{\mathsf{ID}})$ | $\xrightarrow{h_{\mathsf{PIN}}}$ | $x_{\mathsf{ID}} = \mathsf{PIN}_{\mathsf{ID}} \oplus \mathsf{ID}$ |
| | | Store $\mathsf{ID}$, $h(K_{\mathsf{ID}})$ | $\xleftarrow{x_{\mathsf{ID}}, x_K}$ | $x_K = \mathsf{PIN}_{\mathsf{ID}} \oplus h(K_{\mathsf{ID}})$ |
| | | **Privacy Preserving Phase** | | |
| Pick $R_r$ | | | | |
| $\sigma \leftarrow \mathsf{Sign}_{sk_g}(\mathsf{ID}_g \| R_r)$ | $\xrightarrow{\mathsf{ID}_g, R_r, \sigma}$ | Check Signature | | |
| | | Pick $R_m$ | | |
| | | $c_1 \leftarrow \mathsf{Enc}_{pk_g}(R_r \| R_m)$ | | |
| | $\xleftarrow{a_1, c_1}$ | $a_1 \leftarrow \mathsf{Sign}_{sk_m}(c_1)$ | | |
| Check Signature | | | | |
| Recover $R_m$ | | | | |
| $\sigma_r \leftarrow \mathsf{Sign}_{sk_g}(R_m)$ | | | | |
| $c_r \leftarrow \mathsf{Enc}_{pk_m}(\sigma_r)$ | $\xrightarrow{c_r}$ | Check Signature | | |
| | | $e \leftarrow E_{h(K_{\mathsf{ID}})}(\mathsf{ID})$ | | |
| | | $c_2 \leftarrow E_{sk_m}(\mathsf{ID} \| e)$ | | |
| Check signature | $\xleftarrow{a_2, c_2}$ | $a_2 \leftarrow \mathsf{Sign}_{sk_m}(c_2)$ | | |
| Recover $\mathsf{ID}$ | | | | |
| | | **Authentication Phase** | | |
| Pick $R$ | | | | |
| $e \leftarrow \mathsf{Enc}_{pk_m}(R)$ | $\xrightarrow{e}$ | Decrypt $e$ | $\xrightarrow{R}$ | $a_t = h(R \oplus K_{\mathsf{ID}})$ |
| Recover $\mathsf{PIN}_{\mathsf{ID}}$ | $\xleftarrow{e_2}$ | $e_2 \leftarrow \mathsf{Enc}_{pk_g}(a_t)$ | $\xleftarrow{a_t}$ | |

**Figure 6.2:** The MARP-1 protocol, comprising 3 phases: setup, privacy protection, and authentication.

channels, except the one between the reader and the server, are assumed to be insecure. That is, any malicious entity can access all those channels during all phases and manipulate the data transmitted over them.

### 6.3.2 *Cryptanalysis of MARP-1*

**Tracing.** Note that $a_2$ is fixed per tag, being a function of a particular tag $T_t$'s unique identifier $\mathsf{ID}_t$ and its secret key $K_{\mathsf{ID}}$. As the channel between the reader and the MARP is not confidential, an adversary via Execute queries (i.e. eavesdropping) can easily track the movement of $T_t$ by checking for matches of $a_2$ with previously captured values, as the encryption scheme is deterministic. Alternatively, the adversary can replay an old $R$ from MARP to the tag via Send queries, and check if the response $a_t$ matches the old value of $a_t$ corresponding to the

| **Reader** ($\mathsf{ID}_g$)<br>**Key Pair:** $(sk_g, pk_g)$ |  | **MARP**<br>**Key pair:** $(sk_m, pk_m)$ |  | **Tag**<br>**Secrets:** $\mathsf{PIN}_{\mathsf{ID}}, K_{\mathsf{ID}}$ |
|---|---|---|---|---|
|  |  | **MARP Authentication** |  |  |
| Pick $R_r$ |  |  |  |  |
| $\sigma \leftarrow \mathsf{Sign}_{sk_g}(\mathsf{ID}_g\|\|R_r)$ | $\xrightarrow{\mathsf{ID}_g,R_r,\sigma}$ | Check Signature |  |  |
|  |  | Pick $R_m$ |  |  |
|  |  | $c_1 \leftarrow \mathsf{Enc}_{pk_g}(R_r\|\|R_m)$ |  |  |
|  | $\xleftarrow{a_1,c_1}$ | $a_1 \leftarrow \mathsf{Sign}_{sk_m}(c_1)$ |  |  |
| Check Signature |  |  |  |  |
| Recover $R_m$ |  |  |  |  |
| $\sigma_r \leftarrow \mathsf{Sign}_{sk_g}(R_m)$ |  |  |  |  |
| $c_r \leftarrow \mathsf{Enc}_{pk_m}(\sigma_r)$ | $\xrightarrow{c_r}$ | Check Signature |  |  |
|  |  | $e \leftarrow E_{h(K_{\mathsf{ID}})}(\mathsf{ID})$ |  |  |
|  |  | $c_2 \leftarrow E_{sk_m}(\mathsf{ID}\|\|e)$ |  |  |
| Check signature | $\xleftarrow{a_2,c_2}$ | $a_2 \leftarrow \mathsf{Sign}_{sk_m}(c_2)$ |  |  |
| Recover ID |  |  |  |  |
|  |  | **Tag Authentication** |  |  |
| Pick $R_s$ |  |  |  |  |
| $h_r = h(K_{\mathsf{ID}}) \oplus R_s$ | $\xrightarrow{h_r}$ | Pick $R_d$ |  |  |
|  |  | $h_d = h(R_d \oplus h(\mathsf{PIN}_{\mathsf{ID}}))$ |  |  |
|  |  | $h_P = h(\mathsf{PIN}_{\mathsf{ID}}) \oplus R_s$ | $\xrightarrow{R_d,h_d,h_P}$ | Recover $R_s$ |
|  | $\xleftarrow{a_3}$ |  | $\xleftarrow{a_3)}$ | $a_3 = h(K_{\mathsf{ID}} \oplus R_s)$ |
| Check that $a_3$ matches |  |  |  |  |

**Figure 6.3:** The MARP-2 protocol, comprising 2 phases: MARP authentication and tag authentication.

replayed $R$.

We remark that these attacks have less requirements than the ones performed by Juels and Weis [JW07] on some other older RFID protocols that require Corrupt queries.

**Violating the anonymous privacy.** Note that the initial setup messages allow to compute

$$\begin{aligned} z &= [\mathsf{PIN}_{\mathsf{ID}} \oplus \mathsf{ID}_t] \oplus [\mathsf{PIN}_{\mathsf{ID}} \oplus h(K_{\mathsf{ID}})] \\ &= \mathsf{ID}_t \oplus h(K_{\mathsf{ID}}). \end{aligned}$$

Then the adversary simply issues Execute queries to be able to compute $z$, and then issues a Send query to replace the message $R$ from MARP to the tag with $R' = 0$, and so the tag responds with $a_t = h(K_{\mathsf{ID}})$. This allows to compute

$$\begin{aligned} z \oplus a_t &= [\mathsf{ID}_t \oplus h(K_{\mathsf{ID}})] \oplus h(K_{\mathsf{ID}}) \\ &= \mathsf{ID}_t, \end{aligned}$$

and so reveals a potential unique identifier of the tag, which can be cross-checked against the possible list of identifiers for a match.

### 6.3.3 *Tracing MARP-2*

MARP-2 also allows tracing. By eavesdropping both messages via Execute queries between the reader and MARP and between the MARP and the tag, an adversary gets $h(K_{\mathsf{ID}}) \oplus R_s$ and $h(\mathsf{PIN}_{\mathsf{ID}}) \oplus R_s$. By XOR-ing these two values, the adversary gets $h(\mathsf{PIN}_{\mathsf{ID}}) \oplus h(K_{\mathsf{ID}})$ which does not depend on the session parameters and can be used to trace a tag.

This scheme is also vulnerable to replay attacks since the response of the tag only depends on the parameters sent by MARP. So if an adversary sends twice the same message $a_s$ via Send queries, she will get the same response $a_3$ which can also be used for tracing.

## 6.4 Auth2

### 6.4.1 *Description*

Tan et al. [TSL07] addressed the problem of relying on a permanent link to a database server that keep all the tags' secrets for authentication. As in practice there are a dozen of reasons for this connection to be interrupted, they motivated the need of readers that can act without that permanent link. The naïve solution which consists of simply uploading the whole database into the readers is not a reasonable approach, not only for the amount of time and data communication that it induces, but also because it is unsafe to have all the tags' secret put

| **Reader** $\mathcal{R}_i$<br>**Secret:** $L = \{\dots, \ell_{\mathsf{ID}} = f(r_i \| t_{\mathsf{ID}}), \dots\}$ | | **Tag** $\mathcal{T}_{\mathsf{ID}}$<br>**Secret:** $t_{\mathsf{ID}}$ |
|---|---|---|
| | $\xleftarrow{n_t}$ | Pick $n_t$ |
| Pick $n_i$ | $\xrightarrow{n_i, r_i}$ | $h_1 = \mathsf{Trunc}_m(f(r_i \| t_{\mathsf{ID}}))$ |
| Check $\exists \ell_{\mathsf{ID}} \in L: \quad \mathsf{Trunc}_m(\ell_{\mathsf{ID}}) = h_1$<br>Compute ID $= h_2 \oplus h(\ell_{\mathsf{ID}} \| n_i \| n_t)$. | $\xleftarrow{h_1, h_2}$ | $h_2 = h(f(r_j \| t_i) \| n_i \| n_t) \oplus \mathrm{ID}_i$. |

**Figure 6.4:** The Auth2 protocol.

into a device that an adversary may control. Instead, Tan et al. proposed to give each reader the output of a function of each tag's secret and a secret unique to that reader. Concretely, given each tag's secret $t_{\mathsf{ID}}$, that is also listed in the database, and every reader identifier $r_i$, the manager computes a list $L$ including all $f(r_i, t_{\mathsf{ID}})$. Given that the function $f$ is one-way, no malicious entity is able to recover any of the tags' secrets from $L$. On the other hand, correctness of the scheme would require that no collision on the output of $f$ occurs. Hence, the function $f$ needs at least to be collision-resistant.

We now concentrate on the first variant of the second protocol proposed by Tan et al., which we call Auth2. As Figure 6.4 shows, the tag first starts by sending a random nonce $n_t$ to which the reader replies with a nonce $n_j$ and its unique identifier $r_i$. Upon receiving this answer, the tag computes $h_1 = \mathsf{Trunc}_m(h(f(r_i, t_{\mathsf{ID}})))$ and $h_2 = h(f(r_i \| t_{\mathsf{ID}}) \| n_t \| n_j)$. Here $\mathsf{Trunc}_m$ denotes the function that truncates its input to its $m$ least significant bits while $f$ and $h$ are two collision-resistant hash function.

### 6.4.2 *Cryptanalysis of Auth2*

In their security analysis, the authors of Auth2 considered two notions of tracing: definite and indefinite. Definite tracing occurs when an adversary is able to keep track of one precise tag while indefinite tracing is the ability of tracing a members of a group without distinction between them. That is, the adversary is not able to tell more than the fact that the tag under her watch belongs to a certain group she has encountered before. The authors did not claim that Auth2 was secure against indefinite tracing attacks. Instead, they argued for its security against definite tracing attacks as even if the value $h_1$ is fixed per tag, truncating it to $m$ bits leads to many collisions for different tags. Since the output from each tag is not unique, an attacker should not be able to distinguish which tag is outputting this value.

Nevertheless, we show that the Auth2 protocol allows to trace a single tag using the information obtained from two different readers. The attack runs as follow.

1. **Learning**: The attacker eavesdrops several protocol sessions involving a tag $\mathcal{T}_0$ and $\alpha$ readers $\mathcal{R}_1, \dots, \mathcal{R}_\alpha$ via $\mathsf{Execute}$ queries. At the end of this phase, the attacker obtains $\alpha$ pairs of the form $(r_i, h(f(r_i \| t_0))_m)$.

2. **Challenge**: Some time later, when the adversary wishes to track the tag $\mathcal{T}_0$, she starts a session with the challenge tag $\mathcal{T}_b \in \{\mathcal{T}_0, \mathcal{T}_1\}$ replaying $r_1$ by issuing a Send query and comparing the response from the tag for a match on the first part of the message with $h(f(r_1||t_0))_m$. Then, she starts another session replaying $r_2$ via a Send query and checks the response from the tag for a match on the first message component with $h(f(r_2||t_0))_m$. She continues to do that operation for all $\alpha$ pairs she learned in the first phase.

   Now, if two tags have a probability, induced by the birthday paradox, of having the same $h_1$ equal to $2^{-m/2}$ then the probability that $\mathcal{T}_1$ has its responses to the $\alpha$ challenges sent equal to those of $\mathcal{T}_0$ is upper-bounded by $2^{-\alpha m/2}$. Hence, one can choose $\alpha$ so that the probability is negligible. In this case, it becomes highly likely that this is the same tag whose session she had initially eavesdropped, i.e. $\mathcal{T}_b = \mathcal{T}_0$. So, the adversary outputs $\hat{b} = 0$. In the other case, the adversary deduces that $\mathcal{T}_b = \mathcal{T}_1$ and outputs $\hat{b} = 1$.

Note that the attacker always wins when $\mathcal{T}_b = \mathcal{T}_0$ but fails when $\mathcal{T}_b = \mathcal{T}_1$ and all queries match. From this, we can derive the advantage of the adversary

$$\begin{aligned}
\mathsf{Adv} &= \Pr[\hat{b} = b] - 1/2 \\
&= 1/2 \Pr[\hat{b} = b | b = 1] + 1/2 \Pr[\hat{b} = b | b = 0] - 1/2 \\
&= 1/2 \Pr[\hat{b} = b | b = 1] \\
&= 1/2 (1 - \Pr[\hat{b} = \neg b | b = 1]) \\
&\leq 1/2 (1 - 2^{-\alpha m/2}).
\end{aligned}$$

Since it has a non-negligible advantage, the adversary we described earlier is significant.

## 6.5 YA-TRAP, YA-TRAP+ and O-TRAP

In this section, we study a series of optimistic authentication protocols that were proposed for RFID tags: Ya-TRAP, designed by Tsudik [Tsu06], and their follow-up YA-TRAP+ and O-TRAP, due to Burmester et al. [LBdM06]. The term optimistic refer to two different behaviors readers follow depending on whether the system is under attack or not. When no malicious entity interferes with the system, a very optimized procedure, compared to standard protocols from the literature, is performed. However, when the protocol deviates from the ideal case, a special procedure is launched from the reader. Although this last procedure usually has a higher cost than typical verification algorithms for RFIDs, one hopes that a system does not get under attack most of the time in practice. If this is the case, then deploying two approaches, one of them being optimistic, can be beneficial for the scalability of the system.

| **Reader** | | **Tag** |
|---|---|---|
| **Secret hash table:** | | **Public:** $t_{\mathsf{max}}$ |
| $\{\dots,(t_j,\mathrm{HMAC}_{K_{\mathsf{ID}}}(t_j)),\dots\}$ | | **Secret:** $K_{\mathsf{ID}}, t_{\mathsf{ID}}$ |
| | $\xrightarrow{t_j}$ | if $t_j \in [\![t_i+1, t_{\mathsf{max}}]\!]$ |
| | | $h_j \leftarrow \mathsf{HMAC}_{K_{\mathsf{ID}}}(t_j)$ |
| | | $t_{\mathsf{ID}} \leftarrow t_j$ |
| | | else |
| check $\exists\mathsf{ID}$ s.t. $(t_{\mathsf{ID}}, h_j) \in L$. | $\xleftarrow{h_j}$ | $h_j \leftarrow \mathsf{PRNG}_i(t_j)$. |

**Figure 6.5:** The YA-TRAP protocol.

### 6.5.1 *YA-TRAP*

The steps of YA-TRAP are given in Figure 6.5, where HMAC refers to the HMAC construction of a MAC from a has function [BCK96] and PRNG is a pseudo-random number generator. Each tag is initialized with an initial timestamp $t_0$ and a max value for it, denoted $t_{max}$, as well as a unique secret value $K_{\mathsf{ID}}$. Regarding their computational capabilities, tags are assumed to have PRNG implemented, and we denote by $\mathsf{PRNG}^j_{\mathsf{ID}}$ the $j$-th element outputted from the sequence of $\mathcal{T}_{\mathsf{ID}}$'s PRNG.

YA-TRAP is a simple challenge-response protocol in which the reader starts by issuing a timestamp $t_j$ for the challenge. The tag's response consists of computing the function HMAC with its secret key $K_{\mathsf{ID}}$ and the received timestamp $t_j$ if this latter is in the interval limited by the current timestamp $t_c$ and its maximum value $t_{\mathsf{max}}$. However, when the last condition is not fulfilled, the tag instead answers with $\mathsf{PRNG}_{\mathsf{ID}}(t_j)$. For verification, the reader is given a hash table, computed by the database server which holds all the tags' keys, consisting of the outputs of the HMAC of each tag's secret and their corresponding timestamp. In order to recover the tag's identity, the reader searches that list for a pair matching $(t_j, h_j)$ and returns the corresponding tag.

We note that this approach is optimistic in the sense that the reader is able to recover the identity of the partner tag when the tag did not update its timestamp value. In other words, this procedure only works when the system is *not* under attack. We consider two scenarios for desynchronization attacks. In the first one, the tag gets desynchronized to a value $t_{\mathsf{ID}}$ that is smaller than or equal $t_{\mathsf{max}}$. To recover the identity of that tag, the reader can try increasing values for the challenge timestamp and forward the tag's answers to the database. Having knowledge of all secrets, the database is able to recover the tag's identity providing that the reader supplies it with the correct $t_{\mathsf{ID}}$. Clearly, if the reader sends all possible values for $t_{\mathsf{ID}}$ then the database will be able to recognize the partner tag. However, when $t_{\mathsf{ID}}$ is equal to $t_{\mathsf{max}}$, the tag always outputs a random answer independent from its internal key and is thus permanently unable to authenticate itself. In other words, an adversary could mount a denial of service attack by sending the tag in the future, i.e., sending $t_{\mathsf{max}}$. Tsudik acknowledged that securing against this type of attacks requires to put more computation on the tag and was not a primary goal of YA-TRAP. Finally, note that for an RFID system composed of $n$ tags, the complexity in the optimistic case for the reader and database side is $O(n)$ to construct the hash table and then $O(1)$ to recover the partner tag whereas in the other case the complexity for recovering one tag is $O(t_{\mathsf{max}} n)$.

Two operating modes were proposed for YA-TRAP, *real-time* and *batch*. The difference between the two being that while in the former mode, the reader instantaneously authenticates a tag, in batch mode the reader only collects responses for multiple sessions and later communicates with the database server for identification. For applications that do not require an immediate response, such as inventory control, batch mode presents the advantage of being easier to deploy since the readers are not required to maintain a persistent link to the database. However, when immediate feedback is required, such as library check-outs, retail outlets, or contactless credit cards, real-time mode should be used.

The main goal of YA-TRAP's design was to achieve untraceable privacy (UPriv) with adversaries assumed to be able to corrupt tags. Albeit Tsudik explicitly stated that resistance to denial of service attacks was not among the features of YA-TRAP, we show that such an attack still allows an adversary to track any chosen tag.

**Tracing tags in real time.** In the YA-TRAP specification, it was suggested that the top value $t_{\mathsf{max}}$ of a tag's timestamp does not need to be unique but could instead be shared by a batch of tags.

Consider a scenario where tags have different $t_{\mathsf{max}}$, operating in real-time mode. Indeed, acknowledging the fact that tags are produced by different manufacturers for diverse applications, it seems inevitable that some tags will have a different $t_{\mathsf{max}}$. This leads us to an adversary who can trace a tag, hence breaking the UPriv notion of privacy, as follows. For simplicity, assume two tags $\mathcal{T}_0$ and $\mathcal{T}_1$ with respective $t_{\mathsf{max}_0}$ and $t_{\mathsf{max}_1}$, where $t_{\mathsf{max}_0} < t_{\mathsf{max}_1}$.

1. **Learning**: Issue a Send query with $t_j = t_{\mathsf{max}_0}$ to a tag $\mathcal{T} \in \{\mathcal{T}_0, \mathcal{T}_1\}$. Since $t_{\mathsf{max}_0}$ is much into the future than current $t_i$ value, a response $h_j = \mathsf{HMAC}_{K_{\mathsf{ID}}}(t_j)$ is expected, irrespective of which tag it is. Furthermore, the tag will update its local time counter as $t_i = t_{\mathsf{max}_0}$. This action serves to send the tag into the future by marking it for future tracing.
2. **Challenge**: Some time later, when it is desired to trace the tag, issue a Send query with $t_j$ for $t_{\mathsf{max}_0} < t_j < t_{\mathsf{max}_1}$. If $\mathcal{T} = \mathcal{T}_0$, it will respond $h_j = \mathsf{PRNG}_i^j$ and will not successfully pass the validation check by the reader. If $\mathcal{T} = \mathcal{T}_1$, it will respond $h_j = \mathsf{HMAC}_{K_{\mathsf{ID}}}(t_j)$ and will successfully pass the validation check. Thus by observing the reader-tag interaction via Execute queries, an adversary can distinguish between $\mathcal{T}_0$ and $\mathcal{T}_1$ and win the privacy game.

Juels and Weis [JW07] gave two tracing attacks on YA-TRAP that are valid in their privacy model, thus showing YA-TRAP does not meet their definition of strong privacy. Their model arguably assume that the adversary is able to interact with each tag on its own before having to recover the identity of one of them. This feature clearly reflects real-world capabilities of attackers in the context of RFID tags. Moreover, it turns out to be mandatory for the attack of Juels and Weis to work. In contrary, we do not make this assumption and simply assume that the adversary interacts with all tags similarly. That is, our attack applies to a more constrained setting for the adversary by forcing a common $t_{\mathsf{max}}$ for all tags.

YA-TRAP was designed to specifically output a random response even if the tag does not want to be validated by the reader, such that an adversary is unable to distinguish between that random response and a proper response. Yet, by observing the output of the reader-tag interaction, i.e. seeing if the tag passes the validation or not, still allows the distinguishing. In this sense, using the YA-TRAP approach of generating random responses by itself is not sufficient to prevent tracing.

To reiterate, our attack can be prevented if the adversary is unable to observe the output of the reader-tag interaction, i.e. it does not know if the tag successfully passes the reader's validation check. This inability in fact corresponds to the *narrow* adversary model defined in Vaudenay's privacy model [Vau07]. One example setting that fits this narrow model is the batch mode suggested for YA-TRAP. Nevertheless, the batch mode is not relevant for applications where immediate feedback is required and is only meaningful when tags are assumed to be honest since they are not authenticated on the spot but later. Clearly, this last assumption is hard to justify.

**Cloning.** First note that due to computational restrictions, it must be that $t_{\mathsf{max}} - t_0$ is a polynomial function in the security parameter of the scheme. Hence, we can have an adversary enumerating all those timestamps and querying a particular tag with all of them. In the end, the adversary obtains a list of pairs of the form $(t_j, h_j)$ that she can use to produce a clone to the earlier tag. The forged tag only needs to have that list and to answer to the $h_j$ that corresponds to the $t_j$ it receives. Clearly, this tag gets authenticated so the YA-TRAP protocol does not protect against cloning attacks.

### 6.5.2 *YA-TRAP+*

To address availability of all RFID tags, which is the main conceptual limit of YA-TRAP, Burmester et al. [LBdM06] proposed an extension to the latter protocol called YA-TRAP+. The difference between the two version essentially lies in the absence of the max timestamp $t_{\mathsf{max}}$. Moreover, the reader is assumed to have access to the tags' secrets $K_{\mathsf{ID}}$ (instead of the outputs of a function of these secrets). For authentication, the reader issues a timestamp $t$ and a random value $r_t$, just like in YA-TRAP. The response from the tag then depends on the comparison of the received timestamp with the one he received during the session before. If the timestamp it receives is greater than the stored one then the tag answers with $h_1 = H_{K_i}(00||t||r_t)$ and updates its internal timestamp $t_{\mathsf{ID}}$ to the received one. In the other case, the tag computes $h_1 = H_{K_{\mathsf{ID}}}(01||r_i||r_t)$, for a randomly chosen $r_i$, but does not update $t_{\mathsf{ID}}$.

| **Reader**<br>**Database:** $\{\dots,(\mathsf{ID},K_{\mathsf{ID}}),\dots\}$ | | **Tag**<br>**Secret:** $K_{\mathsf{ID}}, t_{\mathsf{ID}}$ |
|---|---|---|
| Pick $r_t$ | $\xrightarrow{t_j,r_t}$ | Pick $r_{\mathsf{ID}}$ |
| | | if $(t > t_{\mathsf{ID}})$ |
| | | $h_1 = H_{K_i}(00\vert\vert t\vert\vert r_t)$ |
| | | $t_{\mathsf{ID}} \leftarrow t$ (without optional part) |
| | | else |
| check $\exists(t_j, K_{\mathsf{ID}}) \in L$ s.t. | $\xleftarrow{r_{\mathsf{ID}},h_1}$ | $h_1 = H_{K_i}(01\vert\vert r_{\mathsf{ID}}\vert\vert r_t)$ |
| $h_1 = H_{K_{\mathsf{ID}}}(00\vert\vert t\vert\vert r_t)\vee$ | | |
| $h_1 = H_{K_{\mathsf{ID}}}(01\vert\vert r_{\mathsf{ID}}\vert\vert r_t)$ | | |
| | **Optional Part** | |
| $h_2 = H_{K_{\mathsf{ID}}}(10\vert\vert r_{\mathsf{ID}}\vert\vert t)$ | $\xrightarrow{h_2}$ | if $(t > t_{\mathsf{ID}} \wedge h_2 = H_{K_{\mathsf{ID}}}(10\vert\vert r_{\mathsf{ID}}\vert\vert t))$ |
| | | $t_{\mathsf{ID}} \leftarrow t$ |

**Figure 6.6:** The YA-TRAP+ protocol.

To avoid distinguishing attacks by the number of values that the tag returns, it is set such that it sends $r_i$ and $h_1$ in both cases, although it is useless in the first one. Verification from the reader is straightforward since it has all the tags' secrets.

To fix the vulnerability to DoS attacks which affects YA-TRAP, it was proposed to add an optional phase to YA-TRAP+ which implements reader authentication. In this variant, the reader issues a third message $h_2 = H_{K_{\mathsf{ID}}}(10||r_i||t)$. When a tag receives that message, it decides whether it matches with the answer it expects. In case of a match, the tag updates its $t_{\mathsf{ID}}$ to $t$, providing that $t > t_{\mathsf{ID}}$. The whole protocol is shown in Figure 6.6.

It turns out that the tracing attack against YA-TRAP is simpler when applied to YA-TRAP+ if its optional second pass is implemented. The attack runs as follows.

1. **Learning**: An adversary first issues Send queries to the tag $\mathcal{T}_0$ with some $r_t$ and a value $t$ that is predictably much larger than the tag's $t_{\mathsf{ID}}$.The adversary then obtains the response $r_i, h_1 = H_K(00||t||r_t)$. After that, she sends a random message $h_2$ which, with very high probability will make $\mathcal{T}_0$ not authenticate its partner as the reader. Consequently, $\mathcal{T}_0$ does not update its internal time counter $t_{\mathsf{ID}}$ to $t$.
2. **Challenge**: We let the adversary first issue a Send query to the challenge tag $\mathcal{T}_b$ with the same $r_t$ and $t$. If the challenge tag is $\mathcal{T}_0$, it will return the response $r_i', h_1 = H_K(00||t||r_t)$ for which $h_1$ is the same as the one answered in the first phase. Otherwise, the adversary knows $\mathcal{T}_b = \mathcal{T}_1$. This allows to track the tag and win the privacy game.

Note that YA-TRAP+ was specifically designed to resist the kind of tracing attack that we mounted on its predecessor YA-TRAP. Yet, this result shows that the optional second pass of

| **Reader** | | **Tag** |
|---:|:---:|:---|
| **Database:** $\{\dots,(r_{\text{ID}},K_{\text{ID}}),\dots\}$ | | **Secret:** $K_{\text{ID}}$ |
| Pick $r_t$ | $\xrightarrow{r_t}$ | |
| check $\exists(r_{\text{ID}},K_{\text{ID}})$ in DB s.t. | $\xleftarrow{r_{\text{ID}},h}$ | $h = H_{K_{\text{ID}}}(r_t, r_{\text{ID}})$ |
| $h = H_{K_{\text{ID}}}(r_t, r_{\text{ID}}) \vee h = H_{K_{\text{ID}}}(r_t, r_{\text{ID}})$ | | $r_{\text{ID}} \leftarrow H_{K_{\text{ID}}}(r_{\text{ID}})$ |
| $r_{\text{ID}} \leftarrow H_{K_{\text{ID}}}(r_{\text{ID}})$ | | |

**Figure 6.7:** The O-TRAP protocol.

YA-TRAP+, which was meant to provide additional security to resist denial of service attacks, makes the protocol vulnerable to tracing attacks.

### 6.5.3 *O-TRAP*

Beside proposing YA-TRAP+, Burmester et al. proposed another authentication protocol for RFID tags called O-TRAP. In its spirit, this protocol is similar to OSK [OSK05]. To authenticate itself, the tag computes a keyed function $H_{K_{\text{ID}}}$ of a challenge sent by reader $r_t$ and a self chosen nonce $r_{\text{ID}}$, i.e., it computes $h = H_{K_{\text{ID}}}(r_t, r_{\text{ID}})$, and sends both the output of $H$ and its nonce to the reader. Having knowledge of the keys, the reader goes through all those keys to find the one for which the output of $H$ matches. The steps of O-TRAP are shown in Figure 6.7.

1. **Learning**: An adversary can issue a Send query to the tag $\mathcal{T}_0$ with random values $r_t$ repeatedly, causing the tag to update its $r_{\text{ID}}$ each time such that it is way into the future compared to its synchronization with the reader.

2. **Challenge**: The adversary observes the future interaction between a tag $\mathcal{T}_b \in \{\mathcal{T}_0, \mathcal{T}_1\}$ and a reader via Execute queries to see if the reader accepts the tag as valid. If not, then the adversary knows this was the tag that it marked during the learning phase, i.e. $\mathcal{T}_b = \mathcal{T}_0$. Else, $\mathcal{T}_b = \mathcal{T}_1$.

Note that this kind of attack has been independently applied by Juels and Weis [JW07] to a couple of other older RFID protocols. Yet what is interesting, as has been demonstrated here, is that more recent provably secure protocols like YA-TRAP+ and O-TRAP still allow for tracing. In this particular case, the privacy leakage of the protocols come from the poor formulation of privacy in the LBdM model as it permits to prove that protocols that allow tracing attacks are private.

| **Reader** | | **Tag** |
|---|---|---|
| **Database:**<br>$\{\ldots,(\mathsf{ID},K^j_{\mathsf{ID}},\mathrm{HMAC}_{K^j_{\mathsf{ID}}}(t_j)),\ldots\}$ | | **Shared secret:**<br>$K_i, K_{\mathsf{ID}}, t_{\mathsf{ID}}$ |
| | $\xrightarrow{K'_j,t'_j}$ | |
| | | $\delta t = t'_j - t_j$<br>if $(\delta t > 0 \wedge H^{\delta t}(K'_j) = K_j)$<br>$t_j = t'_j,$<br>$K_j = K'_j,$<br>$K_{\mathcal{T}_i} = H^{\delta t}(K_{\mathcal{T}_i}),$<br>$h_j = \mathsf{HMAC}_{K_{\mathcal{T}_i}}(t'_j).$<br>else<br>$h_j = \mathsf{PRNG}^j_i.$ |
| | $\xleftarrow{h_j}$ | |
| check $\exists \mathcal{T}_i, h_j : \langle \mathcal{T}_i, K^j_{\mathcal{T}_i}, h_j\rangle \in L.$ | | |

**Figure 6.8:** The RIPP-FS protocol.

## 6.6 RIPP-FS

RIPP-FS was proposed by Conti et al. [CPMS07] as an improvement to the YA-TRAP type protocols that features resilience to denial of service attacks and forward privacy. This last notion deals with the privacy of sessions that precedes the leakage of the tag's secrets to an attacker. Albeit tag authentication works in a similar way to YA-TRAP, RIPP-FS includes an additional key that is shared between all the tags and the reader and is used to authenticate the latter. Concretely, that key is derived from a hash chain seeded by a value $w$ and is defined as follow.

$$\begin{cases} K_\ell = w \\ K_i = H(K_{i+1}) = H^{\ell-1}(w), i = 0,\ldots,\ell-1 \end{cases}$$

To perform reader authentication, every tag is given $K_0$. For time period $i$, it is the key $K_i$ that will be sent by the reader as the first message for authentication along with a period counter $t_i$. Having the key of the last period, the tag checks that $K_{i-1} = H(K_i)$ (if one time period separates the current authentication from the last one. In general, the tag checks that $K_{i+t_i-t_{\mathsf{ID}}} = H(K_i)$ ). The rest of the protocol follows YA-TRAP: the tag updates its internal period counter $t_{\mathsf{ID}}$ to $t_i$ if the former value is greater than the later. The tag also updates $K_{i+t_i-t_{\mathsf{ID}}}$ to $K_i$ and returns $\mathsf{HMAC}_{K_{\mathsf{ID}}}(t_i)$ to the reader which is able to recover $K_{\mathsf{ID}}$ and hence deducing the identity of the partner tag. As in the YA-TRAP protocol, when the received timestamp is smaller than the stored one, the tag does not perform any of the updates mentioned before and rather answers with $\mathsf{PRNG}_{\mathsf{ID}}(i)$. The steps of RIPP-FS are given in Figure 6.8.

In a similar way to YA-TRAP, it is possible to trace an RFID tag implementing the RIPP-FS protocol in the following way.

1. **Learning**:
   a) Query $\mathsf{Send}$ to the reader to initiate two protocol sessions, obtaining $(K_j, t_j)$ and $(K_{j+1}, t_{j+1})$, where $t_{j+1} > t_j$, and $K_j = H(K_{j+1})$.
   b) Make a $\mathsf{Send}$ query to a tag $\mathcal{T}_0$ with the value $(K_{j+1}, t_{j+1})$. Since this is a valid message generated from the reader, a response $h_j = \mathsf{HMAC}_{K_{\mathsf{ID}_0}}(t_{i+1})$ is expected. More importantly, the tag will update its time interval counter as $t_{\mathsf{ID}_0} = t_{i+1}$, as well as the other secrets $K_i = K_{i+1}$ and $K_{\mathsf{ID}_0} = H^{t_i - t_{\mathsf{ID}}}(K_{\mathsf{ID}_0})$.
2. **Challenge**: Some time later, when an adversary decides to trace a tag, she issues a $\mathsf{Send}$ query with $(K_i, t_i)$ to the challenge tag $\mathcal{T}_b$, and passes the response to the reader. If $\mathcal{T}_b = \mathcal{T}_0$, then the target tag's response will have been $h_{j+1} = \mathsf{PRNG}_{\mathsf{ID}}(i)$ and will not successfully pass the validation check by the reader. However, when $\mathcal{T}_b = \mathcal{T}_1$, the response $h_{j+1} = \mathsf{HMAC}_{K_{\mathsf{ID}_b}}(t_i)$ will successfully pass the validation check. Thus by passively observing the reader-tag interaction via $\mathsf{Execute}$ queries, an adversary can distinguish between $\mathcal{T}_0$ and $\mathcal{T}_1$ and win the privacy game.

## 6.7 A Backward and Forward Untraceable Protocol

At ICICS '06, Lim and Kwon [LK06] proposed an RFID protocol that offers untraceable privacy ($\mathsf{UPriv}$) both before and after corruption of a tag. This is indeed a major feat, since other RFID schemes in literature are only able to treat backward untraceability, i.e. a corrupted tag cannot be linked to any past completed sessions.

The initialization phase is as follows:

1. The reader chooses a random secret $K_i$ for each tag $\mathcal{T}_i$, and evaluates $m-1$ evolutions of $K_i^0 = K_i$, i.e. $K_i^j = g(K_i^{j-1})$ for $1 \leq j \leq m-1$, where $g$ is a pseudorandom function. It then computes $t_i^j = ext_{l_2}(K_i^j)$ for $0 \leq j \leq m-1$, where $l_2$ is some appropriate bit length, $ext_l(x)$ is an extraction function returning $l$ bits of $x$.
2. The reader also chooses a random $u_i$ for each tag $\mathcal{T}_i$ and computes a key chain $\{w_i^j\}_{j=0}^{n-1}$ of length $n$, such that $w_i^n = u_i$ and $w_i^j = h(w_i^{j+1})$ for $0 \leq j \leq n-1$, where $h$ is a pseudorandom function.
3. The tag stores $\langle w_{i,T}, K_i \rangle$ where $w_{i,T} = w_i^0$ and initializes a failure counter $c_i = 0$.
4. The reader creates two tables $L_1, L_2$ for $\mathcal{T}_i$ in its database, where $L_2$ is empty and $L_1$ has entries of the form $\langle s_i, \{t_i^j\}_{j=0}^{m-1}, u_i, n_i, w_{i,T}, w_{i,S} \rangle$ where $n_i = n$ and $w_{i,S} = w_i^1$ thus $w_{i,T} = h(w_{i,S})$.

After initialization, a normal protocol session is illustrated as in Figure 6.9, where $f$ is a pseudorandom function. For further discussions on this protocol, the interested reader is referred to [LK06].

| **Reader $\mathcal{R}$** | | **Tag $\mathcal{T}_i$** |
|---:|:---:|:---|
| **Database:** $\{\dots,(K_i,\mathsf{tables}L_1,L_2),\dots\}$ | | **Secret:** $w_{i,T},c_i,K_i$ |
| pick $r_1$ | $\xrightarrow{r_1}$ | |
| | | $t_i \leftarrow \mathsf{ext}_{l_2}(K_i)$ |
| | | pick $r_2$ |
| | $\xleftarrow{t_i,r_2,\sigma_1}$ | $\sigma_1 \leftarrow \mathsf{ext}_{l_1}(f(K_i,r_1\Vert r_2))$. |
| check $\exists t_i^j : (t_i^j = t_i) \wedge (t_i^j \in \{t_i^k\}_{k=0}^{m-1})\wedge$ | | |
| $\langle K_i,\{t_i^k\}_{k=0}^{m-1},u_i,n_i,w_{i,T},w_{i,S}\rangle \in (L_1 \cup L_2)$ | | |
| calculate $K_i' = g(K_i)^j, \sigma_1' = ext_{l_2}(f(K_i',r_1\Vert r_2))$ | | |
| and check that $\sigma_1' = \sigma_1$ | | |
| calculate $\sigma_2 = f(K_i',r_2\Vert r_1)\oplus w_{i,S}$ | $\xrightarrow{\sigma_2}$ | $w_{i,S} = f(K_i,r_2\Vert r_1)\oplus\sigma_2$. |
| for $k = 0\dots m-j-1$ calculate: $\hat{t}_i^k = t_i^{j+k+1}$; | | check $h(w_{i,S}) = w_{i,T}$. |
| for $k = m-j\dots m-1$ calculate: | | If yes: |
| $\hat{K}_i = g(K_i'), \hat{t}_i^k = ext_{l_2}(g(\hat{K}_i)^{k-m+j})$; | | $c_i = 0; w_{i,T} = w_{i,S}$; |
| update $\hat{K}_i, \{t_i^k\}_{k=0}^{m-1}$ in $L_2$ | | $K_i = g(K_i \oplus (w_{i,T}\Vert r_1\Vert r_2))$. |
| calculate $K_i = g(K_i\oplus(w_{i,S}\Vert r_1\Vert r_2))$; | | else |
| $t_i^j = ext_{l_2}(g(K_i)^j)$ for $j = 0\dots m-1$; | | $c_i = c_i + 1$; |
| $n_i = n_i - 1, w_{i,T} = w_{i,S}, w_{i,S} = h(u_i)^{n_i}$ | | if $c_i < m$ |
| update $\langle K_i,\{t_i^k\}_{k=0}^{m-1},n_i,w_{i,T},w_{i,S}\rangle$ in $L_1$ | | update $K_i = g(K_i)$. |

**Figure 6.9:** The backward and forward untraceable RFID protocol.

**Tracing the Tag.** For the purpose of understanding our attack, it suffices to review the gist of the Lim-Kwon protocol. The tag updates its stored secret $K_i$ in two possible ways. If the reader is successfully authenticated, it would update as $K_i = g(K_i \oplus (w_{i,T}||r_1||r_2))$. Else, the tag would update as $K_i = g(K_i)$, up to $m$ times of unsuccessful authentications, after which the tag stops updating its $K_i$. This eventual non-updating allows the reader to catch up.

Our attack nevertheless works using the basic pattern of desynchronization that we applied in Section 6.4. Recall that the idea of the attack is to intentionally desynchronize the tag from the reader by sending the tag into the future.

1. **Learning:** An adversary sends $m$ number of queries $r_1^j$ for $1 \leq j \leq m$ to the tag $\mathcal{T}_0$, and records the tag's response $t_j$ for $1 \leq j \leq m$. Since the adversary is impersonating the reader, thus each time it will not pass the check by the tag, and so each time the tag would update its stored secret as $K_i = g(K_i)$, from which $t_i$ will be derived in the next session.
2. **Challenge:** Query $r_1^m$ to the tag $\mathcal{T}_b \in \{\mathcal{T}_0,\mathcal{T}_1\}$, and obtain its response $t^*$.
3. **Guess:** Check if $t^* = t_m$. If so, then the adversary knows this was the tag it queried during the learning phase i.e. $\mathcal{T}_b = \mathcal{T}_0$. Else, it knows that $\mathcal{T}_b = \mathcal{T}_1$.

Lim and Kwon remarked that once a tag is successfully authenticated by a reader, then the tag's stored secret $K_i$ would be freshly randomized so that tracing of any kind is prevented. Yet, our adversary can repeat the above step of the Learning phase by sending $m$ arbitrary queries $r_1^j$ for $1 \leq j \leq m$ to the tag again to desynchronize it and the same tracing attack applies.

In order to solve the DoS problem, the authors included a feature into the design of the protocol that unfortunately allowed our attack causing the tag to be traceable even without corruption, although the goal for their protocol was much stronger i.e. backward and forward untraceability even with corruption.

**Violating the Forward Untraceability.** Another goal of the protocol is to achieve forward untraceability, i.e. even if a tag is corrupted thus leaking its stored secret $K_i$, it should be impossible for the adversary to trace the tag in future sessions. Nevertheless, we describe an attack in the context of the example application provided by Lim and Kwon of a tag embedded in a purchased item. Initially, the seller's reader $\mathcal{R}_1$ has legitimate access to the tag. At the point of purchase, ownership of this access should transfer to the buyer's reader $\mathcal{R}_2$. The attack can be mounted either by the seller's reader or by an outsider adversary having access to Corrupt queries.

1. An outsider adversary issues a Corrupt query to the tag $\mathcal{T}_b$, obtaining its stored secret $K_i$. Alternatively, the seller's reader $\mathcal{R}_1$ knows the stored secret $K_i$ and $w_{i,T}$.
2. At the point of purchase, the buyer's reader $\mathcal{R}_2$ interacts with the tag in a protocol session, thus updating $K_i$. During this time, the adversary eavesdrops the values $r_1, r_2$ communicated in the session.
3. Right after the interaction between the tag and the buyer's reader $\mathcal{R}_2$, the adversary initiates a protocol session with the tag. Since she knows the previous $K_i$, and also the latest values of $r_1, r_2$, the adversary can recompute the latest $K_i = g(K_i \oplus (w_{i,T} || r_1 || r_2))$ and thus produces a message the passes the tag's verification procedure. This way, the adversary can trace the tag in all future sessions and block other readers, including the buyer's,from authenticating the tag.

This result contradicts the protocol's claim that its ownership transfer is perfect. While Lim and Kwon argued that the protocol achieves forward untraceability under the assumption that the adversary cannot eavesdrop on all future legitimate interactions involving the tag and the reader; the above attack works without violating that assumption. Lim and Kwon also gave a provable security model for forward untraceability in its Appendix. However, their protocol was not formally proven in that model, and only a sketch of the proof was provided in [LK06].

## 6.8 O-FRAP and O-FRAKE

At AsiaCCS '07, Le et al. [LBdM06] presented a universally composable (UC) [Can00] privacy model for RFID protocols, and proposed O-FRAP and O-FRAKE. These two protocols

| **Reader** $\mathcal{R}_j$<br>**Database:** $\{\ldots,(r_i,K_i),\ldots\}$ | | **Tag** $\mathcal{T}_i$<br>**Secret:** $r_i, K_i$ |
|---:|:---:|:---|
| pick $r$ | $\xrightarrow{r}$ | |
| | $\xleftarrow{r_i,v_2}$ | $v_1 \Vert v_2 \Vert v_3 \Vert v_4 \Vert v_5 = F(K_i, r \Vert r_i)$ |
| check $\exists(r_i, , K_i)$ in DB | | Set $r_i \leftarrow v_1$. |
| calculate $v'_1 \Vert v'_2 \Vert v'_3 \Vert v'_4 = F(K_i, r \Vert r_i)$ | | |
| check $v'_2 = v_2$ | | |
| output Accept($\mathcal{T}_i$) | | |
| update $(r_i, K_i) = (v'_1, v'_4)$ in DB | $\xrightarrow{v'_3}$ | If $(v_3 = v'_3)$ |
| | | Output Accept($\mathcal{R}_j$). |
| | | Set $K_i \leftarrow v_4$ |

**Figure 6.10:** The O-FRAP protocol.

| **Reader**<br>**Database:** $\{\ldots,(r_i,K_i,SK_i),\ldots\}$ | | **Tag**<br>**Secret:** $r_i, K_i, SK_i$ |
|---:|:---:|:---|
| pick $r$ | $\xrightarrow{r}$ | |
| | $\xleftarrow{r_i,v_2}$ | $v_1 \Vert v_2 \Vert v_3 \Vert v_4 \Vert v_5 = F(K_i, r \Vert r_i)$ |
| check $\exists(r_i, , K_i, SK_i)$ in DB | | Set $r_i \leftarrow v_1$. |
| calculate $v'_1 \Vert v'_2 \Vert v'_3 \Vert v'_4 \Vert v'_5 = F(K_i, r \Vert r_i)$ | | |
| check $v'_2 = v_2$ | | |
| output Accept($\mathcal{T}_i, SK_i$) | | |
| update $(r_i, K_i, SK_i) = (v'_1, v'_4, v'_5)$ in DB | $\xrightarrow{v'_3}$ | If $(v_3 = v'_3)$ |
| | | Output Accept($\mathcal{R}_j, SK_i$). |
| | | Set $\langle K_i, SK_i \rangle \leftarrow \langle v_4, v_5 \rangle$. |

**Figure 6.11:** The O-FRAKE protocol.

are shown in Figures 6.10 and 6.11, respectively, in which $F$ denotes a pseudorandom function.

### 6.8.1 *Tracing O-FRAP*

O-FRAP is formally proven to be a secure untraceable RFID protocol in the LBdM model where corruption of tags is allowed, in the sense that the only information revealed to an adversary is if a party is a tag or a reader. Yet we show here how its untraceable privacy can be violated by presenting a tracing attack that is valid even in a weaker privacy model where corruption possibility is not granted to the adversary.

The attack works as follows:

1. **Learning:** The adversary sends an arbitrary $r$ value to the tag $\mathcal{T}_0$, but does not complete

the protocol. This causes the tag to update its $r_i$, while its $K_i$ remains unchanged, thus marking the tag for future tracing.

2. **Challenge:** To trace the tag in future, the adversary observes the interaction between the reader and the tag $\mathcal{T}_b$.

3. **Guess:** If the reader does not output Accept, then the adversary knows that this tag was indeed the tag that it marked in step (1), i.e. $\mathcal{T}_b = \mathcal{T}_0$. Otherwises, he deduces that $\mathcal{T}_b = \mathcal{T}_1$.

### 6.8.2 *Violating the Forward Privacy of O-FRAP*

In the Le-Burmester-de Medeiros model, corruption is not allowed before a protocol session is initiated, and it is assumed that upon corruption of a party, either a tag or a reader, then the corrupted party's current incomplete session offers no privacy. It is claimed that privacy is maintained for all previously completed sessions involving the corrupted party.

To motivate our case, we consider the definition of subsession completion in the LBdM model. A subsession is a party's view of its current protocol session, e.g. during an O-FRAP protocol session, both the reader and the tag have their own separate views of that session, so-called their subsession. To quote from [LBdM06], "Upon successful completion of a subsession, each party accepts its corresponding partner as authenticated." Thus, at the point where a party outputs Accept, its subsession is already considered completed.

Referring to the O-FRAP description in Figure 6.10, the reader's subsession is completed at the point when it outputs Accept, i.e. before it updates its entry in $L$ and before it sends $v_3'$ to the tag. Meanwhile, the tag's subsession is completed at the point that it outputs Accept, i.e. before it updates its $K_i$. In the context of the Le-Burmester-deMedeiros model, corruption of a party at this point should not violate the privacy of the party corresponding to its completed subsession. This is the problem with the O-FRAP proof that we are exploiting. Indeed, we show how this can be circumvented.

1. The adversary first eavesdrops on an O-FRAP session and records $\langle r, r_i, v_2 \rangle$.

2. Then, it corrupts a tag $\mathcal{T}_i'$ at the point after the tag outputs Accept. It thus obtains $K_i'$ corresponding to a previoulsy completed subsession, and not the updated $K_i' = v_4$.

3. The adversary calculates $v_1^*||v_2^*||v_3^*||v_4^* = F(K_i', r||r_i)$. It can then check the computed $v_2^*$ with its recorded $v_2$ for a match, thereby associating the tag $\mathcal{T}_i'$ to the particular completed subsession corresponding to its recorded $\langle r, r_i, v_2 \rangle$.

Our attack here requires a stronger adversary than the other attacks we have presented in earlier sections of this chapter. Yet, assuming corruption capabilities is taken into account by the Le-Burmester-deMedeiros model in which O-FRAP's privacy was proven, and shows that O-FRAP does not achieve its goal of forward untraceable privacy.

### 6.8.3 *Breaking the Forward Secrecy of O-FRAKE*

The above attack can be extended to break the forward secrecy of the O-FRAKE protocol, which is an extension of O-FRAP that furthermore establishes a shared secret session key between the tag and reader.

1. The adversary first eavesdrops an O-FRAKE session and records $\langle r, r_i, v_2 \rangle$.
2. It then corrupts a tag $\mathcal{T}_i'$ at the point after the tag outputs Accept. It thus obtains a pair $\langle K_i', SK_i' \rangle$ corresponding to a previously completed subsession, and not the updated $\langle K_i', SK_i' \rangle = \langle v_4, v_5 \rangle$.
3. The adversary calculates $v_1^*||v_2^*||v_3^*||v_4^*||v_5^* = F(K_i', r||r_i)$. It can then check the computed $v_2^*$ with its recorded $v_2$ for a match, thereby associating the tag $\mathcal{T}_i'$ to the particular completed subsession corresponding to its recorded $\langle r, r_i, v_2 \rangle$; and further it also knows that the established session key for that associated session is $SK_i'$.

## 6.9 Conclusion

Although we have used a very limited privacy model, we have been able to show that several RFID protocols that allegedly addressed privacy were vulnerable to rather simple attacks. We identify the main cause behind these failure to be the lack of formal analysis. Indeed, most presented protocols were only supported by informal arguments that cannot take into account all the possible attacks an adversary can perform. Therefore, we stress the need of studying the extend of privacy an RFID protocol offers by providing a formal proof of security.

Moreover, we have shown that the choice of the model is crucial as it can be that a protocol is proven private according to a model with a correct reduction and Still be vulnerable to privacy attacks not covered by the model. As it was demonstrated with the O-FRAP and O-FRAKE protocols, this applies to the LBdM model.

# 7

# PRIVACY MODELS FOR RFID

Contents

Before moving on to Vaudenay' privacy model, we review other privacy models that were proposed. We proceed chronologically and present Avoine et al.'s model [ADO06] (ADO), the Juel-Weis model[JW07, JW07] and its extension due to Damgård and Ostergaard [DP08] (We refer to the Juels-Weis model and its extension as the eJW model.) and finally the zero-knowledge based model of Deng et al. [DLYZ10]. These models will be formally compared to our model in the next chapter.

As a contribution, we show that the privacy experiment of Juels and Weis simplifies when one takes correctness and soundness into consideration.

## 7.1 The ADO Model

To the best of our knowledge, the first formal treatment for studying the privacy of RFID systems is due to Avoine, Dysli, and Oechslin [ADO06] who used an ad-hoc model to analyze Molnar and Wagner's scheme [MW04]. It was subsequently improved by Avoine in his PhD thesis [Avo05]. We refer to this model by ADO.

In short, this model is based on the notion of indistinguishability: A scheme is supposed to preserve privacy if an adversary choosing a target tag and getting either that tag or another one, with both events happening with probability $1/2$, cannot tell which tag she received with a better chance than guessing, i.e., deducing the tag's identity with probability $1/2$. To perform the attack, the adversary is given the secret state of another RFID tag and is allowed to interact with the target tag before it is submitted to the challenger.

## 7.2 The Extended-Juels-Weis Model

The ADO model was generalized by Juels and Weis [JW07, JW09] who elaborated on ADO's privacy game to attain a notion that is closer to classical indistinguishability games for encryption schemes. That is, contrarily to the ADO model, the adversary gets to choose both target tags and receives one of them in return. The adversary is also able to corrupt any tag except the two targets and has control over the communication channel. Note that the corruption model of Juels and Weis allows the adversary to set a new key for the corrupted tag.

Juels and Weis model RFID systems as a set of tags interacting with a single reader is set up by an algorithm denoted Gen that outputs $n$ secrets, each one for a tag, and gives the reader the $n$ secrets. The reader can, contrarily to tags, maintain multiple sessions in parallel. For this, the reader binds every running protocol session to a unique session identifier sid that is put in a table containing all the messages belonging to the session.

Attackers are assumed to have complete control over all communications between parties. They interact with the RFID system through several interfaces.

- The ReaderInit interface allows to trigger protocol sessions and make the reader output the session identifier sid and the first message for the session. (The Juels-Weis model assumes that it is always the reader that initiates the communication.)

- The TagInit interface serves to bind a tag to a session. As such, it needs to receive an sid. Once sid has been set for the reader or a tag, the adversary may send messages of the form $(\mathsf{sid}, m)$ to which the party answers with a message computed using the previous messages related to sid, sid, its secret, i.e., the tag's key or the reader's list of keys, and its internal randomness. Note that when a tag receives a TagInit, it aborts the current session and deletes all internal data, except for the key $K$, even if this happens while the tag is in a middle of another protocol session. At some point, the reader performs a verification step by computing a function over its entire internal state, including all running sessions and any internal key material to output an "accept" or a "reject" for the session sid and close it. That is, the adversary is always assumed to be able to determine whether a protocol instance is succeeded or failed.

- Tag corruption and initialization are done through an interface denoted SetKey. Upon reception of a SetKey message with parameter $K'$, a tag answers with $K$, its current secret, and replaces it by $K'$. The tag does not update its secret if the parameter is not specified. A tag that has received a SetKey message is said to be corrupted.

**Definition 7.1 (Privacy in the Juels-Weis Model)**
*Let $k$ be a security parameter and $\mathcal{A}$ be a polynomial time algorithm that takes as input four parameters $n$, $s$, $r$, $t$ and follows the following privacy experiment.*

```
System Setup
  for $i = 1$ to $n$ do
      $K_i \leftarrow \mathsf{Gen}(1^k)$
  end for
  Init reader with $(K_1, \ldots, K_n)$
  Set each tag's $T_i$ secret by SetKey$(K_i)$
Phase 1: Learning
  Interact with the system without exceeding $r$ ReadInit calls, $t$ TagInit calls, and $s$ computation steps. Leave at least two tags uncorrupted.
Phase 2: Challenge
  Select two uncorrupted tags and denote them $T_0^\star$, $T_1^\star$.
  Pick $b \in_R \{0, 1\}$ and give $\mathcal{A}$ access to $T_b^\star$. $T_{\neg b}^\star$ becomes unreachable.
  Interact with the system without exceeding $r$ ReadInit calls, $t$ TagInit calls, and $s$ computation steps and sending SetKey to $T_b^\star$.
  Output $b'$
Winning condition: $b = b'$
```

*A scheme is said to be $(n, r, s, t)$-private if every polynomial-time adversary $\mathcal{A}$ playing the privacy* *experiment is such that*

$$\left|\Pr[\mathcal{A}(1^k)\ \textit{wins}] - \frac{1}{2}\right| = \mathsf{negl}(k)$$

This definition was also extended to cover the case of forward-privacy, i.e., when the adversary learns about the key of the target tag. This eventuality was included in the privacy experiment by adding a step before the eighth one in which the attacker issues a SetKey message on $T_b^\star$.

However, two key concepts were not discussed by Juels and Weis: correctness and soundness. That is, an RFID scheme is useless if a tag having an undisturbed session with the reader does not get authenticated or if an attacker can come up with a way to impersonate a tag to the reader. (In fact, designing a private RFID protocol without both requirements is easy: It suffices to make the tag output random bits.) This issue was adressed by Damgård and Pedersen [DP08] who introduced two notions of correctness and soundness. In short, correctness ensures that whenever a tag and a reader share a protocol session, the tag gets authenticated whereas soundness requires that no polynomial-time adversary who can corrupt all the system's tags but one can make the reader accept a tag $T$ for a protocol session in which $T$ was not involved.

Damgård and Pedersen also gave further clarifications on the structure of the protocols that are considered in the model. As the Juels-Weis model does not seem to support asymmetric keys (that choice was probably made because public-key cryptography is believed to be too expensive for simple devices as RFID tags), Damgård and Pedersen refined the definition of symmetric-key based protocol by giving to each tag, instead of a key $K_i$, an access to a random oracle $\Psi_{K_i}$. On its side, the reader is given the list of all these keys and hence has access to all the oracles $\Psi_{K_i}$. Moreover, Damgård and Pedersen showed that security for an RFID system mandates that the reader accesses $\Psi_{K_i}$ to authenticate a tag $T_i$. Notice that proceeding in this way implies a linear complexity for the reader in the number of the system's RFID tag. Consequently, corrupting a tag $T_i$ is modeled by granting access to the oracle $\Psi_{K_i}$. We further note that privacy in the sense of Juels-Weis can only be obtained for systems in which the keys are *independent*, i.e., when the $\mathsf{Gen}$ algorithm is stateless and does not receive any hidden parameter.

Albeit the description given above can be used to describe most protocols based on conventional cryptography, it may be that there are more efficient ways for the reader to verify a protocol session. This is especially the case for correlated tags' secrets where the complexity can be reduced to a logarithmic factor [MW04] at the cost of a weaker model of privacy in which the adversary may not be allowed to corrupt every tag from the system. Despite its limitations, the weaker model has its benefits in practical settings. So, it was taken into account by Damgård and Pedersen who introduced a fifth parameter for the adversary in the privacy experiment that defines the maximal numbers of tags the adversary can corrupt in order to retain privacy.

**Simple Privacy for the eJW Model** Juels and Weis argued that the attacker's ability to corrupt any tag except the two targets induces the requirements that the scheme should not use strongly correlated secrets for the tags. In the following, we incorporate Damgård and Pedersen's requirements of correctness and soundness to yield a simpler but equivalent privacy experiment for the eJW model.

**Definition 7.2 (Simple Privacy for the eJW model)**
*Let the simple privacy experiment denote the privacy experiment of the Juels-Weis model in which the adversary is never allowed to send* SETKEY *messages. This experiment writes as*

```
System Setup
  for $i = 1$ to $n$ do
      $K_i \leftarrow \mathsf{Gen}(1^k)$
  end for
  Init reader with $(K_1, \dots, K_n)$
  Set each tag's $T_i$ secret by SETKEY$(K_i)$
Phase 1: Learning
  Interact with the system without exceeding $r$ READINIT calls, $t$ TAGINIT calls, $s$ computation
      steps, and without issuing SETKEY messages.
Phase 2: Challenge
  Select two tags and denote them $T_0^\star$, $T_1^\star$.
  Pick $b \in_R \{0, 1\}$ and give $\mathcal{A}$ access to $T_b^\star$. $T_{\neg b}^\star$ becomes unreachable.
  Interact with the system without exceeding $r$ READINIT calls, $t$ TAGINIT calls, $s$ computation
      steps, and without issuing SETKEY messages.
  Output $b'$
Winning condition: $b = b'$
```

*We say that a scheme is $(n, r, s, t)$-simple private if every polynomial-time adversary $\mathcal{A}$ playing the simple privacy experiment is such that*

$$\left| \Pr[\mathcal{A}(1^k) \text{ wins}] - \frac{1}{2} \right| = \mathsf{negl}(k).$$

**Theorem 7.1**
*If an RFID scheme is correct, simple private, and uses independent keys, then the scheme is private.*

**Proof.** We assume, without loss of generality, that once the adversary corrupts a tag, she does not query it anymore. Instead, she uses her access to the oracle $\Psi_\cdot$ to simulate her interactions with corrupted tags. To simulate the final outcome of a protocol session, i.e., the last message from the reader, the adversary checks with the list of oracles she obtained from her SETKEY queries. Correctness ensures that if the reader authenticates a corrupted tag, the adversary's simulation does the same.

If after trying with all these queries, the adversary is still unable to find the session partner, she forwards the request to the reader. Clearly, the latter will not identify a corrupted tag so simulation is perfect.

At last, using the fact that keys are independent, we can assert that the protocol messages produced by the uncorrupted tags are unrelated to the states obtained by corruption. Therefore, the success probability of the adversary is not affected by the simulation. The resulted game corresponds to the simple privacy experiment. □

## 7.3 Zero-Knowledge Privacy

The formulation of zero-knowledge privacy [DLYZ10], abbreviated zk-privacy, is derived from the literature on zero-knowledge [GMR85, GMR89] with the idea of linking privacy and zero-knowledge. The link between these two notions is done by noticing the fact that privacy requires an adversary interacting with a random tag not to learn anything else than what she could have deduced herself is reminiscent of the fact that a verifier should not learn anything from interacting with the prover in zero-knowledge interactive proofs (with the difference that in RFID protocols the malicious attacker is not one of the scheme's entities). Consequently, the privacy definition follows the simulation paradigm: An adversary is considered not to have learnt anything from interacting with a tag if every outputs it makes can be produced by another polynomial-time algorithm, called simulator, that does not have access to that tag.

Adversaries in zk-privacy are placed in the center of the RFID system and have complete control over communication channels, except for the one between the reader and its database server. To interact with the system, the adversary has access to a set of four interfaces denoted InitReader, to create a protocol instance, SendT, to send a message to a tag, SendR, to send a message to a reader, and Corrupt, for corruption (which reveals the content of the tag's permanent and volatile memory to the adversary). ZK-Privacy assumes that protocol sessions end with a message from the reader telling whether the session succeeded or failed. It is therefore unnecessary to define an interface for the outcome of a protocol.

The privacy experiment is composed of two phases. After the creation of the reader and a number of tags by a procedure denoted Setup, the adversary $\mathcal{A}$ interacts with the system through the oracles mentionned above. At the end of this phase, she outputs her state and a set $\mathcal{C}$ composed of so-called clean tags. Clean tags are uncorrupted tags that are currently at the status of waiting for the first-round message from the reader to start a new session. A target tag, denoted $T_g$, is then randomly chosen from the set $\mathcal{C}$. In a second stage, the adversary can still interact with all the tags that are not in $\mathcal{C}$ and additionally has blind access to $T_g$, i.e., it cannot corrupt it.

Simulators are also composed of two phases. The first phase of the attack for the simulator is identical to the attack of the adversary described above. However, when it gets the randomly chosen target tag, and after interacting with the tags not in $\mathcal{C}$, the simulator outputs a simulated view, denoted sview. Note that the simulator does not have access to $T_g$. ZK-privacy requires sview to include all oracle answers to the queries made by the simulator. A protocol is said to be zk-private if for every adversary there exists a simulator such that the view of the former is computationally indistinguishable from the sview computed by the latter. Extensions to forward-zk-privacy and backward-zk-privacy were also defined. Hereafter, we give a definition of zk-privacy but we refer the reader to the original papers for a more complete definition, especially for the other properties of completeness and soundness.

**Definition 7.3 (ZK-Privacy)**
*An RFID protocol satisfies zk-privacy, if for any polynomial-time adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ there exists a polynomial-time simulator $\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2)$ such that for all sufficiently large $k$, the outputs from the following games are indistinguishable. ($\mathcal{C}$ is a set of clean RFID tags.)*

*Real game:*

```
1: Run $\mathsf{Setup}(1^k) \to (\mathsf{param}, \mathcal{R}, \mathcal{T})$.
2: Execute $\mathcal{A}_1(\mathsf{param}, \mathcal{R}, \mathcal{T}) \to (\mathcal{C}, st)$
3: pick $\mathcal{T}_g \in_R \mathcal{C}$
4: Run $\mathcal{A}_2(\mathcal{R}, \mathcal{T} - \mathcal{C}, \mathcal{T}_g, st)$
5: Output $(g, \mathsf{view}_{\mathcal{A}})$
```

*Simulated game:*

```
1: Run $\mathsf{Setup}(1^k) \to (\mathsf{param}, \mathcal{R}, \mathcal{T})$.
2: Execute $\mathcal{S}_1(\mathsf{param}, \mathcal{R}, \mathcal{T}) \to (\mathcal{C}, st)$
3: pick $\mathcal{T}_g \in_R \mathcal{C}$
4: Run $\mathcal{S}_2(\mathcal{R}, \mathcal{T} - \mathcal{C}, st) \to \mathsf{sview}$
5: Output $(g, \mathsf{sview})$
```

*Note that* sview *must contain all oracle answers to the queries made by $\mathcal{S}$.*

It is worth mentioning that zk-privacy's notion of privacy is stronger than the one of the eJW model: Any zk-private protocol is private in the sense of [JW07].

In the next chapter, we study the relationship between zk-privacy and our privacy models. Loosely speaking, we exploit the fact that zk-privacy's simulators only need to deal with the removal of one target tag and ensure that privacy is still preserved. However, cases may be that an adversary needs two tags to defeat privacy. We show that in these settings zk-privacy fails to detect any information leakage. At the same time, we show that our model successfully deals with those situations.

# 8

# VAUDENAY'S PRIVACY MODEL

Contents

This Chapter is dedicated to present and compare the model proposed by Vaudenay at AsiaCrypt 2007 [Vau07]. The model we describe in here differs in some details from the one published in [Vau07] in that we do clarify some vague notions used by Vaudenay such as distributions and how RFID tags are selected.

After that, we show implications and separation results with the eJW [JW07, DP08] and ZK-privacy [DLYZ10] models mentioned in the previous chapter. Finally, we present Hermans et al.'s adaptation of the model [HPVP11], discuss its semantics and its point of divergence with Vaudenay's model.

## 8.1 Description

Contrarily to the models we presented in Chapter 7, Vaudenay's model does not fix any particular experiment to capture the notion of privacy. Instead, privacy in Vaudenay's sense is achieved if the adversary is unable to learn any information, written as a boolean statement, from interacting with the RFID system. Of course, an attacker could still learn unrelated facts such as the number of RFID tags that compose the system or how many protocol instances were triggered. However, it seems that nothing can prevent a curious entity who has control over all communication channels from extracting such information. For this reason, Vaudenay's model treats privacy loss in RFID systems as the leakage of information that comes from the wireless link, i.e., the protocol messages. This means that tag tampering is not considered by itself a privacy loss in Vaudenay's model: The fact that an attacker succeeds in extracting the secret contained in an RFID tag is not a privacy leakage. Nevertheless, that piece of information can still lead to be a privacy loss if the secret proves to be useful for deducing an information about past sessions (In this context, we commonly name the notion Forward privacy) or future ones (a protocol preventing such attacks is said to be backward private, or Strong private in Vaudenay's model).

The model also takes into account the possibility of getting the result of protocol instances by the adversary that was introduced by the Juels-Weis model. It also considers the possibility of an adversary inserting illegitimate tags into the system. An illegitimate tag behaves exactly like a legitimate one except that it does not have a corresponding entry in the database server. In other words, an illegitimate tag never gets authenticated. From a practical point of view, one may see these illegitimate tags as RFID tags that follow the same protocol specs but belong to a different RFID system. It is not hard to see that the use of such tags always compromises privacy: If a malicious entity gets holds of a tag, she can easily tell whether it belongs to a system or not by looking at the outcome of a protocol session on the system's reader. Clearly, this privacy leakage is independent from the cryptography used in the tag. It is therefore not considered as a privacy loss in Vaudenay's model.

To capture its notion of privacy, Vaudenay proposed the notion of blinder. Basically, if the adversary is unable to use the protocol messages to compute its statement, then those protocol messages could be changed to ones that are produced by an entity who, like the adversary, does not know the tags' secrets and that change would go unnoticed to the adversary. Producing those messages is delegated to an entity called the blinder. Still, the blinder has to be aware of the actions of the adversary. In particular, it has knowledge of all the tags' identifiers and the internal state of the corrupted ones (this only applies when considering strong or destructive privacy). As a stateful algorithm, the blinder is allowed to gradually take into account the history of the adversary's actions to compute its protocol messages. In the end, an RFID system preserves privacy if for every adversary against privacy, there exists a blinder that runs in polynomial time and for which replacing the protocol messages by the one computed by this blinder does not affect the adversary's final statement. Alternatively, we can consider a stronger, more restrictive, notion of privacy by requiring the blinder to be universal, independent from the adversary. As we shall see later, all the blinders that were constructed by Vaudenay [Vau07] were in fact universal.

## 8.2 Definition of the Model

### 8.2.1 *RFID System*

Like most models for RFID, Vaudenay considers an RFID system to be composed of a reader permanently connected to a database server and a number of RFID tags that communicate with the reader through a wireless link. As the link between the reader and the database is assumed to be secure, Vaudenay takes the simplification of merging the reader and the database server into one entity.

In Vaudenay's model, every tag is bound to a unique publicly known identifier $\mathsf{ID}$ and is given a secret state $S_{\mathsf{ID}}$. The reader has a key pair $(pk, sk)$ (this key pair will be useful for protocols that use public-key cryptography) and the database consists of entries of the form $(\mathsf{ID}, K_{\mathsf{ID}})$, where $\mathsf{ID}$ refers to a tag identifier and $K_{\mathsf{ID}}$ is the key corresponding to its state.

**Definition 8.1 (RFID System)**
*An RFID system is composed of the following three algorithms*

- $\mathsf{SetupReader}(1^k) \to (sk, pk)$. *This first probabilistic polynomial-time algorithm is used to initialize the reader. As such, it takes as input a security parameter $k$ and outputs a pair of secret/public key $(sk, pk)$ for the reader (if no public-key cryptography is used in the RFID scheme then $pk$ is set to $\perp$).*
- $\mathsf{SetupTag}_{pk}(\mathsf{ID}) \to (K_{\mathsf{ID}}, S_{\mathsf{ID}})$. *This probabilistic polynomial-time algorithm creates a tag with unique identifier $\mathsf{ID}$. The state $S_{\mathsf{ID}}$ is stored inside the tag while an entry $(\mathsf{ID}, K_{\mathsf{ID}})$ is inserted in the server's database $\mathsf{DB}$ when the created tag is legitimate.*
- *A two-party game run between a reader and a tag $\mathsf{ID}$, in which each one of them follows an interactive polynomial-time probabilistic algorithm, comes to complete the definition.*

*Apart from a random tape, the algorithm for the tag takes as input the state $S_{\mathsf{ID}}$ while the algorithm for the reader takes as input the database* DB*, and the reader's secret key $sk$. In the end, the reader ends up with a tape* Output*, set to $\bot$ if the instance failed from its perspective. A protocol execution with* ID *is called succeeded if it has a matching conversation with the reader with output* ID.

**Simple RFID Protocols** The definition of an RFID system given above is very general and does not take into account the particularity of RFID tags. Due to their constraints, most of the RFID schemes proposed in the literature are elementary challenge-response protocols: the reader sends a challenge to which the tag replies with the output of some (often randomized) function of its state and the received challenge. Some of these protocols may also include a challenge from the tag. This is particularly the case when reader authentication is performed. In order to identify its partner, the reader sends a query to the database. The database server processes the query by applying a predicate $\Psi$, that depends on the secret key and the protocol transcript, on every entry and outputs the only pair $(\mathsf{ID}, K_{\mathsf{ID}})$ that satisfies the predicate. The eventual future messages in the session from the reader may depend on this database entry. In the event that more than one entry satisfies the predicate, the database acts as if she could not identify the correct tag (Note that in the case of correct RFID schemes, this happens with negligible probability). Such schemes are named *simple RFID schemes*.

**Definition 8.2 (Simple RFID Scheme)**
*An RFID scheme is said to be simple if the following conditions are fulfilled:*

- *Protocol messages do not depend on $sk$. They may depend on some entry $(\mathsf{ID}, K_{\mathsf{ID}})$ if the latter has already been identified as the partner tag for the session.*
- *The reader sends a query to the database with its secret key $sk$ and the (possibly partial) transcript $\tau_p$ obtained from a protocol session $\pi$.*
- *There exists a predicate, i.e., a deterministic polynomial-time algorithm that outputs a single bit, $\Psi$ that takes as input $sk$, $\tau_p$, and a database entry $(\mathsf{ID}, K_{\mathsf{ID}})$ such that the response from the database is computed by returning the set of database entries, denoted $E_\pi$, that satisfies the predicate (this implies that the predicate is tested on every database entry). The reader then uniformly chooses one entry from $E_\pi$ and returns it. If no such entry is found, then it returns $\bot$.*
- *Once a tag* ID *has been identified in the database, its corresponding secret in the database, $K_{\mathsf{ID}}$, may be updated to a new value. When it takes place, this procedure is carried out by an algorithm* Update *that is given as input $sk$,* ID*, $K_{\mathsf{ID}}$, and the full transcript of the protocol instance $\tau$. This algorithm outputs a new $K_{\mathsf{ID}}$ and the database entry $(\mathsf{ID}, K_{\mathsf{ID}})$ is updated.*

Note that this definition slightly differs from the one given by Vaudenay in that the original definition assumed the existence of an arbitrary efficient sampling algorithm to choose the final tag ID from $E_\pi$ whereas we fix the distribution to be uniform.

### 8.2.2 *Adversarial Capabilities*

We follow on Vaudenay's definitions and consider powerful adversaries who have complete control over the communication channel: The adversary sits at the center of the RFID system and has all messages transiting through her. Therefore, she has the ability to insert, delete, and modify messages. She can also provoke authentication sessions and relay, insert, and delete messages. She additionally is given the ability to order the creation of legitimate RFID tags and can request the creation of illegitimate ones. The difference between the two type of tags being that only legitimate tags have a corresponding entry in the system's database.

Furthermore, adversaries have the ability to have one or more RFID tags in their vicinity to watch over them. RFID tags that are under control of the adversary are called drawn and the others are said to be free. For a tag that has been drawn, all its interactions become controlled by the adversary: This latter has access to all its communications, can trigger protocol instances with the reader and send messages to the tag. Not only can an adversary choose to draw specific tags, but she can select them randomly following a specified probability distribution. For instance, the adversary may be able to draw one tag over two, one with probability $^1/_3$ and the other with probability $^2/_3$ without knowing a priori which tag she has obtained. At any time, an adversary may decide to release a drawn tag and the latter becomes free. Apart from the drawn tags, the DrawTag oracle returns a bit for each one of them telling whether each tag is legitimate or not. The reason for introducing these bits is to prevent the kind of attacks that were described in Section 5.1.1 concerning legitimate and illegitimate tags.

Drawn tags can also be tampered with: an adversary can "open" a tag and retrieve its internal state. While the leakage of the internal state through tag corruption is not under question, whether the adversary can also extract the contents of its volatile memory, i.e., the random variables it was using right before it was opened is debatable. Even if the contents of the temporary memory fade away when not connected to a power source, recent attacks have shown that it is still possible to "freeze" the volatile memory and extract bits from it [HSH$^+$08]. However, this issue is not a threat if we only consider two-message protocols and assume that the parties securely delete the contents of their volatile memory right after the protocol terminates.

More formally, we give the following definition for an adversary, adaptated from Vaudenay's original work.

**Definition 8.3 (Adversary against an RFID System)**
*An adversary against an RFID system is a probabilistic polynomial-time algorithm which takes a public key $pk$ as input and interacts with the system through the following nine interfaces.*

- *CreateTag$^b$*$(\mathsf{ID})$*: create a tag with unique identifier* $\mathsf{ID}$*. Depending on the bit $b$ submitted by the adversary, the tag may be legitimate, when $b = 1$, or illegitimate, when $b = 0$. After calling upon* $\mathsf{SetupTag}_{pk}(\mathsf{ID}) \to (K_{\mathsf{ID}}, S_{\mathsf{ID}})$ *for both type of tags, the pair* $(\mathsf{ID}, K_{\mathsf{ID}})$ *is inserted into the database if the adversary queried for a legitimate tag.*

- *DrawTag*$(\mathsf{Samp}) \to ((\mathsf{vtag}_1, b_1), \ldots, (\mathsf{vtag}_n, b_n))$*: select a set of tags according to a distribution specified by a polynomial-time sampling algorithm* Samp. *During the period in which a tag is drawn, the adversary has complete control over its interactions. Along* vtag, *a bit* $b$*, set to 1 whenever the drawn tag is legitimate and to 0 when it is illegitimate, is returned. When a tag is drawn, it is designated by a unique virtual fresh identifier* vtag. *Drawing a tag that was already drawn makes the oracle output* $\perp$.

  *Additionally, this interface keeps a private table* $\mathcal{T}$ *that keeps track of the real identifier of each drawn tag, i.e., it is such that* $\mathcal{T}(\mathsf{vtag})$ *is the real identifier of the virtual tag* vtag.
- *Free*$(\mathsf{vtag})$*: release the RFID tag with virutal identifier* vtag *and makes it unreachable for the adversary. Yet, the adversray can still choose to later draw it again.*
- *Launch* $\to \pi$*: make the reader launch a new protocol instance* $\pi$*. Without loss of generality, this oracle can be assumed to be deterministic. For easier notations, we denote by* $\mathsf{Output}(\pi)$ *the tape that the reader obtains after the completion of the instance* $\pi$.
- *SendReader*$(m, \pi) \to m'$*: send a message* $m$ *to a protocol instance* $\pi$ *for the reader.*
- *SendTag*$(m, \mathsf{vtag}) \to m'$*: send a message* $m$ *for the drawn tag* vtag *and receives the answer* $m'$.
- *Execute*$(\mathsf{vtag}) \to (\pi, \tau)$*: executes a complete protocol instance between the reader and the drawn tag* vtag. *It returns the transcript of the protocol denoted* $\tau$*, the list of successive protocol messages.*
- *Result*$(\pi) \to x$*: returns the result of the completed protocol instance* $\pi$*. Namely, it yields* $0$ *when* $\mathsf{Output} = \perp$ *and* $1$ *otherwise.*
- *Corrupt*$(\mathsf{vtag}) \to S$*: returns the current state* $S$ *of the tag* $\mathcal{T}(\mathsf{vtag})$*. It does not return the content of the temporary memory of the tag.*

Definition 8.3 differs from Vaudenay's with respecct to the introduction of the algorithm Samp in DrawTag queries. Vaudenay uses a vague term of "distribution" for the input of DrawTag which may use exponential length. For example, this happens when the adversary wants to draw all tags in a random order at the same time. As it will be made more explicit in the next chapter, such a restriction is necessary for the efficiency of the RFID system, especially for the security proofs to hold. That is, as soon as we want to reduce the security of the scheme to a computational assumption, the environment has to be executable in a polynomial number of steps. That can not be guaranteed unless samplings can be performed in polynomial time.

Depending on the type of RFID tags, adversaries may not be able to query all the interfaces defined above. Several classes of adversaries deal with those disparities.

**Definition 8.4 (Adversarial Classes)**
*Depending on the restrictions on accessing the interfaces listed in Definition , we categorize adversaries in several classes.*

- ***Strong.*** *This is the class of adversaries that has absolute access and no restriction.*

- ***Destructive.*** *It refers to the class of adversaries for who tampering with a tag results in its destruction. In a more formal sense, a Destructive adversary is not allowed to issue any query with* vtag *after requesting* CORRUPT(vtag).
- ***Forward.*** *After a Forward adversary corrupts a tag, she is only allowed to corrupt other tags or terminate.*
- ***Weak.*** *This class captures the set of adversaries who cannot corrupt any tag.*

*Orthogonal to this classification, we also consider the case in which adversaries do not have access to the* RESULT *oracle. Such adversaries are referred to as Narrow and for every class listed above, we consider a Narrow counterpart.*

- ***Narrow-Strong.*** *This class denotes the set of Strong adversaries who do not access* RESULT.
- ***Narrow-Destructive.*** *This includes all Destructive adversaries who cannot access* RESULT.
- ***Narrow-Forward.*** *This is equivalent to Narrow ∪ Forward.*
- ***Narrow-Weak.*** *This is the class of the weakest adversaries who can neither corrupt tags nor access* RESULT.

Regarding the relation between those adversarial classes, it is clear that for every non-narrow class $P$ we have Narrow$-P \subset P$. It also holds that Weak $\subset$ Forward $\subset$ Destructive $\subset$ Strong and Narrow-Weak $\subset$ Narrow-Forward $\subset$ Narrow-Destructive $\subset$ Narrow-Strong.

### 8.2.3 *Matching Conversation*

Before defining the necessary properties of an RFID scheme, we formalize the event that a tag and a reader have an undisturbed protocol instance. This notion will prove to be useful to define correctness and security for RFID systems.

**Definition 8.5 (Matching Conversation)**
*We say that a protocol instance $\pi$ had a matching conversation with the tag* ID *if they exchanged well interleaved and faithfully (but maybe with some time delay) messages until $\pi$ is completed.*

### 8.2.4 *Correctness*

Basically, correctness formalizes the fact that whenever the reader and a tag ID participate in an undisturbed protocol session, the reader authenticates the tag, that is, it ends up with Output $=$ ID, except with a small negligible probability. The difference between our definition and vaudenay's definition of correctness is that we take into account all possible actions that may have happened in the past for the system. That is, we require that a legitimate tag remains successful in authenticating itself and an illegitimate one gets rejected regardless of the past events that occurred in the RFID system. The definition we propose is in fact close to the definition of adaptive completeness from the ZK-Privacy model [DLYZ10].

**Definition 8.6 (Correctness of an RFID Scheme)**
*Let $\mathcal{A}$ be a Strong adversary interacting with the RFID system in which she creates $n$ RFID tags and produces no output. We also assume without loss of generality that $\mathcal{A}$ frees all tags before terminating.*

*An RFID scheme is said to be correct if for every such $\mathcal{A}$ and every efficient sampling algorithm* Samp *on the set of the system's tags, we have*

$$\Pr\left[\begin{array}{c|c} b=1 \wedge \mathsf{Output}(\pi)=\mathcal{T}(\mathsf{vtag}) & (pk,sk)\leftarrow \mathsf{SetupReader}(1^k) \\ \vee & \textit{Execute } \mathcal{A}(pk) \\ b=0 \wedge \mathsf{Output}(\pi)=\perp & (\mathsf{vtag},b)\leftarrow \textit{DrawTag}(\mathsf{Samp}) \\ & (\pi,\cdot)\leftarrow \textit{Execute}(\mathsf{vtag}) \end{array}\right] = 1-\mathsf{negl}(k)$$

We also propose a weaker notion of correctness in which only tags that have not completed more than $t$ consecutive unsuccessful instances get authenticated by the reader. That is, we propose the following definition.

**Definition 8.7 (Weak Correctness for Simple RFID Schemes)**
*A simple RFID system is said to be weakly-correct if*

- *There exists an efficiently computable predicate $\Psi'$ such that if a tag* ID *and the reader have a matching conversation in a session $\pi$ and the tag* ID *has previously completed $t$ successive sessions without the reader authenticating it, we have*

$$|\Pr[\Psi'(\mathsf{ID},t)\to 1]-\Pr[\mathsf{Output}(\pi)=\mathsf{ID}]| = \mathsf{negl}(k)$$

- *For every Strong adversary $\mathcal{A}$ that produces no output but frees all the tags before terminating, and every efficient sampling algorithm* Samp *on the set of the system's tags, we have*

$$\Pr\left[\begin{array}{c|c} (b=1 \wedge \Psi'(\mathcal{T}(\mathsf{vtag}),t) & (pk,sk)\leftarrow \mathsf{SetupReader}(1^k) \\ \wedge\mathsf{Output}(\pi)=\mathcal{T}(\mathsf{vtag})) & \mathcal{A}(pk) \\ \vee & (\mathsf{vtag},b)\leftarrow \textit{DrawTag}(\mathsf{Samp}) \\ (b=0 \wedge \mathsf{Output}(\pi)=\perp) & (\pi,\cdot)\leftarrow \textit{Execute}(\mathsf{vtag}) \end{array}\right] = 1-\mathsf{negl}(k)$$

For simple schemes, this definition of weak-correctness means that the output of $\Psi'$ is computationally indistinguishable from the output of $\Psi$ with matching sessions and known tag identifiers. Clearly, this definition is less restrictive than Vaudenay's who mandated perfect indistinguishability between the two predicates, i.e., the original definition states thar the two predicates have to be equivalent. Since the definition of correctness leaves a negligible probability that the reader authenticates another tag in place of the one it is running the instance with, reflecting this probability in the $\Psi'$ predicate is reasonable.

### 8.2.5 *Security*

Security is the equivalent of soundness in the eJW model. It formalizes the fact that no adversary should be able to make the reader accept a protocol session in which the adversary has been actively involved in the sense that sge did not only relay messages. In summary, an RFID scheme is said to be secure if no Strong adversary is able to make a reader protocol instance recognize an uncorrupted tag $\mathsf{ID}$ except with negligible probability and that is even if the adversary corrupts all the other tags, unless $\pi$ and the tag have a matching conversation.

**Definition 8.8 (Security of an RFID System)**
*We say that an RFID scheme is secure if for every Strong adversary, the probability that the reader ends with a tape* $\mathsf{Output} = \mathsf{ID}$ *for a session* $\pi$ *that has no matching conversation with the tag* $\mathsf{ID}$ *is negligible in the RFID scheme's security parameter.*

Simple RFID schemes enjoy an interesting property: their security reduces to an adversary playing with a system consisting of a single tag $\mathsf{ID}_t$ and having access to an oracle implementing $\Psi(sk, \cdot, \cdot, \cdot)$ to which she can submit triplets of the form $(\mathsf{ID}, K_{\mathsf{ID}}, \tau)$ with the restriction $\mathsf{ID} \neq \mathsf{ID}_t$. In the following, we give the formal proof that this simplification holds for our class of simple and weakly-correct RFID systems.

**Definition 8.9 (Security of Simple and Weakly-Correct RFID Systems)**
*For simple RFID schemes that are weakly-correct, we consider the following simplified security game for adversaries who are given access to an oracle* $\mathcal{O}_\Psi$ *who checks the predicate* $\Psi(sk, \cdot, \cdot, \cdot)$.

```
1: $(sk, pk) \leftarrow \mathsf{SetupReader}(1^k)$
2: $\textsc{CreateTag}^1(\mathsf{ID})$
3: $\mathsf{vtag} \leftarrow \textsc{DrawTag}(\mathsf{ID})$
4: $\pi \leftarrow \textsc{Launch}$
5: Run $\mathcal{A}^{\mathcal{O}_\Psi}$ interacting with Launch, SendReader, and SendTag. $\mathcal{A}^{\mathcal{O}_\Psi}$ is not allowed
   to specify $\mathsf{ID}$ in its queries to $\mathcal{O}_\Psi$.
6: $b \leftarrow \textsc{Result}(\pi)$
7: Output 1 if $\pi$ has no matching conversation with $\mathsf{ID}$ and $\textsc{Result}(\pi) = 1$.
```

*The scheme is said to be simply secure if the winning probability of any adversary playing the simple security experiment is negligible in the security parameter.*

**Lemma 8.1**
*For simple and weakly-correct RFID schemes, simple security implies security.*

**Proof.** We use the game proof methodology to reduce an adversary against the security of the scheme to an adversary playing the simple security game. We denote by $S_i$ the event that $\mathcal{A}$ wins the experiment described by game $i$.

*Game 0.* This denotes the original security game played by a fixed Strong adversary $\mathcal{A}$. We let $S_0$ be the event that $\mathcal{A}$ succeeds. Recall that $\mathcal{A}$ has access to all interfaces. We assume, w.l.o.g., that $\mathcal{A}$ stops as soon as it wins the security game, i.e., one protocol session $\pi$ identifies a tag $\mathsf{ID}$ without the two having a matching conversation.

*Game 1.* We relax $\mathcal{A}$'s winning condition by declaring that is sufficient that one instance $\pi$ with transcript $\tau$ satisfies $\Psi$ on an input $(\mathsf{ID}, K_{\mathsf{ID}})$ for which $\mathsf{ID}$ had no matching conversation with $\pi$. We further stop the adversary as soon as it wins the game under this condition. Note that the adversary wins the original security game if this tag has been selected from the set $E_\pi$. Therefore, we find that

$$\Pr[S_1] \geq \Pr[S_0]$$

*Game 2.* We add a new condition for $\mathcal{A}$ to win by requiring it to correctly guess the target tag $\mathsf{ID}$ when created and the target instance $\pi$ when launched. If $S_3$ is the event that the adversary wins this game and $n$, $t$ are the number of tags created and sessions launched respectively, we have

$$\Pr[S_2] \geq \frac{1}{nt} \Pr[S_1]$$

*Game 3.* In this game, we simulate all $\mathcal{A}$'s drawings. That is we construct an algorithm $\mathcal{A}_1$ such that, each time a tag is created, $\mathcal{A}_1$ draws it, and subsequently simulates all $\mathcal{A}$'s DrawTag and Free queries. Clearly, the views of $\mathcal{A}$ in both games are perfectly indistinguishable so the winning probability remains unaffected. In other words,

$$\Pr[S_3] - \Pr[S_2] = 0$$

*Game 4.* We now simulate the creation of all tags except the target one. That is, we process all CreateTag queries with a parameter different from $\mathsf{ID}$ in the following way. $\mathcal{A}$ calls $\mathsf{SetupTag}_{pk}(\cdot)$ to generate the tag state and the key for the database. If the query concerns a legitimate tag, $\mathcal{A}$ inserts the entry into a list of legitimate tags $\mathsf{Tags}_1$. Since $\mathcal{A}$ has knowledge of all states of the tags, she can simulate all SendTag queries related to any tag, except $\mathsf{ID}$ that is forwarded to the SendTag interface (Recall that $\mathcal{A}$ draws tags herself so she knows the real $\mathsf{ID}$ of every tag). The simulation is thus perfect, i.e.,

We also need to show that $\mathsf{Output}$, and thus Result, can be simulated. To determine the outcome of a protocol session, $\mathcal{A}$ tests queries $\mathcal{O}_\Psi$ on every entry except $(\mathsf{ID}, K_{\mathsf{ID}})$ to determine which entry satisfies $\Psi$. As for $(\mathsf{ID}, K_{\mathsf{ID}})$, $\mathcal{A}$ assumes that $\Psi$ would answer $0$ if the instance does not have matching conversation with that tag. Otherwise, it assumes it to be $1$. Therefore, when the predicate tested with $(\mathsf{ID}, K_{\mathsf{ID}})$ would have yielded $0$, $\mathcal{A}$ perfectly simulates $\mathsf{Output}$ (the rest of the protocol messages do not depend on $K_{\mathsf{ID}}$ if $\mathsf{ID}$ has not been identified). If the predicate would have answered $1$ with $(\mathsf{ID}, K_{\mathsf{ID}})$ and without matching conversation, it should already have been the target session and this is addressed with another selection in Game 2. So, simulation is perfect and we find that

$$\Pr[S_4] = \Pr[S_3]$$

> Note that the adversary submits its SendReader query if its simulated output is ID so that the database entry can be correctly updated.

Finally, we notice that Game 3 is described by the simple security experiment. We therefore conclude that simple security for simple and weakly-correct RFID schemes implies security. □

### 8.2.6 *Privacy*

The intuition behind the privacy definition in the Vaudenay model is that any significant adversary against privacy should output a statement *deduced from the interactions between the tags and the system*. This was formalized using the classical simulation paradigm: any adversary making effective use of the protocol messages should have its success probability affected if she were to interact with an intermediate between the RFID system and the adversary that simulates the protocol messages. This intermediate system is called a blinder. As privacy only concerns the wireless link, blinders are not required to simulate tag creation and corruption queries.

Therefore, the first step in defining Vaudenay's notion of privacy is to define blinders.

**Definition 8.10 (Blinder)**
*A blinder $B$ for an adversary $\mathcal{A}$ is a stateful polynomial-time algorithm which sees the same messages as $\mathcal{A}$ and simulates the Launch, SendReader, SendTag, and Result oracles to $\mathcal{A}$. The blinder does not have access to the reader's tape and is given neither the reader's secret key $sk$ nor access to the database.*

*A blinded adversary $\mathcal{A}_B$ is an adversary who does not produce any Launch, SendReader, SendTag, Result oracles query but relies on $B$ to obtain answers for those queries.*

The second definition formalizes the privacy game and the fact that privacy means that no adversary deduces any information from the protocol messages.

**Definition 8.11 (Privacy)**
*We consider adversaries who start with an attack phase consisting of oracle queries and some computations then pursuing an analysis phase with no oracle query. In between phases, the adversary receives the hidden table $\mathcal{T}$ of the DrawTag oracle then outputs a bit $b$, or equivalently a boolean statement that evaluates to $1$ or $0$. The adversary wins if the output is $1$.*

*We say that the RFID scheme is $P$ private if for any simulatable adversary $\mathcal{A}$ which belongs to class $P$ there exists a blinder $B$ for which we have*

$$\Pr[\mathcal{A}(pk) \to 1] - \Pr[\mathcal{A}_B(pk) \to 1]| = \mathsf{negl}(k).$$

$$\begin{tikzcd}
\text{Strong} \arrow[r, Rightarrow] \arrow[d, Rightarrow, shift left] & \text{Destructive} \arrow[r, Rightarrow] \arrow[d, Rightarrow, shift left] & \text{Forward} \arrow[r, Rightarrow] \arrow[d, Leftrightarrow] & \text{Weak} \arrow[d, Leftrightarrow] \\
\text{Narrow-Strong} \arrow[r, Rightarrow] \arrow[u, Rightarrow, shift left, "/" marking] & \text{Narrow-Destructive} \arrow[r, Rightarrow] \arrow[u, Rightarrow, shift left, "/" marking] & \text{Narrow-Forward} \arrow[r, Rightarrow] & \text{Narrow-Weak}
\end{tikzcd}$$

**Figure 8.1:** Implications, separations and Equivalences in Vaudenay's Privacy notions. The implications from weaker to stronger privacy notions assume weak-correctness and security for the RFID scheme.

## 8.3 Equivalences and Impossibilities of some Privacy Classes

In this section, we present Vaudenay's results in the equivalence between the different levels of privacy. A special attention is also given to the relation between Forward and key exchange protocols' notion of perfect-forward secrecy as well as to the impossibility result of Strong privacy. Figure 8.1 summarizes these results.

### 8.3.1 *From Narrow Privacy to Privacy*

The first result we present is related to the relation between Narrow-Weak and Weak privacy, and between Narrow-Forward and Forward privacy. Basically, Vaudenay showed that if an RFID authentication protocol that is strongly correct, secure, and achieves Narrow-Weak privacy, resp. Narrow-Forward privacy, is Weak private, resp. Forward private.

**Lemma 8.2**
*Assume a correct and secure RFID protocol. If that protocol is Narrow-Weak private, resp. Narrow-Forward private, then it is Weak private, resp. Forward private.*

The argument holding behind the proof of this theorem is that since the adversary knows whether a drawn tag is legitimate, she must be able to predict the outcome of a protocol session executed with a legitimate tag. On the other hand, security guarantees that no illegitimate tag gets authenticated by the reader, except with negligible probability. (Recall that an illegitimate tag is created by calling SetupTag so an adversary could just obtain a state by running this algorithm and simulating a tag with that state to the reader.) Therefore, the Result interface can be always simulated to the adversary by looking at whether a drawn tag is legitimate, in which case it replies with 1, and with 0 otherwise. Again, correctness ensures that the simulation in case of success is perfect while using an hybrid argument in conjonction with security guarantees that the simulation in case of failure of one protocol instance is computationally indistinguishable from Result's answers.

Interestingly, this result does not apply to Destructive and Strong adversaries. That is because through tag tampering, these type of adversaries get the ability to simulate a potentially legitimate tag that passes authentication. However, from the blinder point of view, no message was forwarded from a tag to a reader. In some sense, the blinder would need to be able to deduce the ID of any identified tag. Unfortunately, this is not realizable if the protocol guarantees Destructive privacy. For instance, the protocol described in Section 8.4.3 and depicted in Figure 8.4 is correct, secure and proven to be Narrow-Strong private while not being Destructive private.

### 8.3.2 *The Impossibility of Strong Privacy*

The most severe drawback of Vaudenay's privacy model is probably its impossibility of achieving Strong privacy. For a better understanding of the solution we propose later, we detail the proof of this result here.

The source of the impossibility result comes from the incompatibility of Narrow-Strong and Destructive privacy. At a first glance, Destructive privacy may seem to be equivalent to Forward privacy if the state of tags are (at least computationally) independent, i.e., the probability of guessing a tag state is unchanged if the algorithm is given another tag state. However, Vaudenay's definitions are made such that a corruption gives a Destructive adversary the ability to compute protocol messages that lead to a successful outcome for the reader (despite the fact that in Destructive privacy a tag does not exist after corruption). For a scheme that meets this level of privacy, there should exist a blinder that successfully predict the outcome of that session. However, it turns out that constructing a blinder that can deduce the hidden actions of the adversary is not an easy task. (Up to date, there is no known instance of a Destructive private protocol.) Nevertheless, the existence of such a blinder for a protocol implies the existence of a significant Narrow-Strong adversary. Therefore, achieving both Destructive and Narrow-Strong privacy at once is impossible.

More explicitly, let us consider a Destructive adversary $\mathcal{A}$ against an RFID scheme who creates one legitimate tag and corrupts it. Then it picks a random state and launches a protocol instance with the reader simulating either the legitimate tag or using the random state. In the end, the adversary outputs true if the reader accepts the instance. The following algorithm, written for the case of a two message protocol, describes $\mathcal{A}$'s behavior.

```
CreateTag$^1$(ID$_0$)
vtag $\leftarrow$ DrawTag(ID$_0$)
$S_0 \leftarrow$ Corrupt(vtag)
$(S_1, \cdot) \leftarrow$ SetupTag$_{pk}$(ID$_1$)
Choose $b \in_R \{0,1\}$
$\pi \leftarrow$ Launch
$a \leftarrow$ SendReader$(\cdot, \pi)$
Compute $c$, the answer of a tag with state $S_b$ receiving challenge $a$ and send it back to the reader.
$b' \leftarrow$ Result$(\pi)$
Output $b \oplus b'$
```

On one hand, because of correctness, the Result interface always answers with 1 when the adversary simulates the legitimate tag. In other words,

$$\Pr[\mathcal{A} \to 1 | b = 0] = 1.$$

On the other hand, when the adversary simulates a tag with random state, then, due to security, the Result interface outputs 1 with a negligible probability. Hence,

$$\Pr[\mathcal{A} \to 1 | b = 1] = 1 - \mathsf{negl}(k).$$

Combining both probabilities, we deduce that

$$\Pr[\mathcal{A} \to 1] = 1 - \mathsf{negl}(k).$$

Since the protocol offers Destructive privacy, there must exist a blinder $B$ for the adversary $\mathcal{A}$ such that $|\Pr[\mathcal{A} \to 1] - \Pr[\mathcal{A}^B \to 1]|$ is negligible. As it sees all the actions of $\mathcal{A}$, this blinder gets a tag ID, $\mathsf{ID}_0$ along with its state $S_0$, that is revealed by $\mathcal{A}$'s corruption query and is asked to simulate two oracle queries, namely one SendReader query and one Result query. Hence, the blinder can be seen as a two-stage algorithm first simulating a message from the reader then waiting for an answer from a tag after which it computes the result of the protocol.

In the following, we are interested in the simulation of the Result query. By correctly predicting the outcome of the protocol, the blinder acts as a distinguisher between a tag with known state and a random one who never uses the secret key of the reader. This means that for a Destructive private scheme, it must be possible to identify tags whose states are known a priori. However, this allows to construct a Narrow-Strong adversary $\mathcal{A}'$ that uses that blinder to identify an anonymous tag communicating with the reader. This adversary works as follows.

```
CreateTag($\mathsf{ID}_0$)
$\mathsf{vtag}_0 \leftarrow$ DrawTag($\mathsf{ID}_0$)
$S_0 \leftarrow$ Corrupt($\mathsf{vtag}_0$)
Free($\mathsf{vtag}_0$)
CreateTag($\mathsf{ID}_1$)
$\mathsf{vtag}_1 \leftarrow$ DrawTag($\mathsf{ID}_1$)
$S_1 \leftarrow$ Corrupt($\mathsf{vtag}_1$)
Free($\mathsf{vtag}_1$)
$\mathsf{vtag} \leftarrow$ DrawTag($\Pr[\mathcal{T}(\mathsf{vtag}) = \mathsf{ID}_0] = \Pr[\mathcal{T}(\mathsf{vtag}) = \mathsf{ID}_1] = \frac{1}{2}$)
Run $B$ on input $pk$, $\mathsf{ID}_0$, $S_0$ and get SendReader message $a$ for $\mathsf{vtag}$.
$c \leftarrow \mathsf{SendTag}(\mathsf{vtag}, a)$
Feed $B$ with $c$ and get its output $b$.
Get $\mathcal{T}$ and output $\mathcal{T}(\mathsf{vtag}) = \mathsf{ID}_{\neg b}$.
```

Clearly, we have that

$$\Pr[\mathcal{A}' \to 1] = \Pr[\mathcal{A}^B \to 1] = 1 - \mathsf{negl}.$$

However, predicting which tag has been drawn is impossible for any blinder who does not have access to the protocol messages. So, for every blinder $B'$, it must be that

$$\Pr[\mathcal{A}'^{B'} \to 1] = \frac{1}{2}.$$

Therefore, no Destructive private scheme is Narrow-Strong private. Since Strong privacy implies both notions together, it is impossible to achieve.

| **Tag** | | **System** |
|---|---|---|
| **State:** $S_{\mathsf{ID}}$ | $S_{\mathsf{ID}} = K_{\mathsf{ID}}$ | **Database:** $\{\dots, (\mathsf{ID}, K_{\mathsf{ID}}), \dots\}$ |
| | $\xleftarrow{\quad a \quad}$ | Choose $a \in_R \{0,1\}^{\alpha/2}$ |
| Choose $b \in_R \{0,1\}^{\alpha/2}$ | | |
| $c = F_{K_{\mathsf{ID}}}(a, b)$ | $\xrightarrow{\quad b,c \quad}$ | Check that $\exists(\mathsf{ID}, K_{\mathsf{ID}}) \in \mathsf{DB} : F_{K_{\mathsf{ID}}}(a, b) = c$ |
| | | **Output:** $\mathsf{ID}$ or $\perp$ if not found |

**Figure 8.2:** A correct, secure, and Weak private RFID authentication protocol based on a pseudo-random function.

## 8.4 Case Studies

This section is dedicated to present Vaudenay's RFID protocols that were included in [Vau07]. Notice that all the proposed protocols consist of two messages, a challenge sent by the reader and an answer computed by the tag.

### 8.4.1 *A Weak-Private Protocol from a PRF*

The first example given by Vaudenay concerns the achievability of Weak privacy by using a PRF. In the sequel, we let $\alpha$ and $\gamma$ be two polynomials functions over **N** and we assume a PRF $F : \{0,1\}^k \times \{0,1\}^{\alpha(k)} \to \{0,1\}^{\gamma(k)}$. The protocol, depicted in Figure 8.2, is described hereafter.

- **Setup.** Since the scheme does not use public-key cryptography, this algorithm is void and is never used.
- **SetupTag.** To create a tag $\mathsf{ID}$, pick a random $k$-bit key $K_{\mathsf{ID}}$ and set $S_{\mathsf{ID}} = K_{\mathsf{ID}}$. When the tag is legitimate, the entry $(\mathsf{ID}, K_{\mathsf{ID}})$ is put in the database.
- **Authentication.** Tag authentication is performed by a challenge-response protocol which works as follow
  - First, the reader picks a $\alpha/2$-bit string $a$ and sends it to the tag.
  - The latter also picks a random $\alpha/2$-bit string $b$ and computes $c = F_{S_{\mathsf{ID}}}(a\|b)$. The tag's answer consists of $b$ and $c$.
  - Finally, the reader looks in its database for a pair $(\mathsf{ID}, K_{\mathsf{ID}})$ that satisfies the equality $c = F_{S_{\mathsf{ID}}}(a\|b)$ and outputs the corresponding $\mathsf{ID}$. If no entry satisfies this equality then the protocol ends in failure and the reader outputs $0$.

We note that this protocol has been proposed by Feldhofer et al. [FDW04] with the PRF being instantiated by a block cipher.

**Theorem 8.1**
*If $F$ is a PRF, the above RFID scheme is correct, secure, and Weak private.*

| **Tag** | | **System** |
|---|---|---|
| **State:** $S_{\mathsf{ID}}$ | $S_{\mathsf{ID}} = K_{\mathsf{ID}}$ | **Database:** $\{\dots, (\mathsf{ID}, K_{\mathsf{ID}}), \dots\}$ |
| | $\xleftarrow{\quad a \quad}$ | Choose $a \in_R \{0,1\}^\alpha$ |
| Compute $c = G(S_{\mathsf{ID}} \Vert a)$ | $\xrightarrow{\quad c \quad}$ | Check that $\exists(\mathsf{ID}, K_{\mathsf{ID}}) \in \mathsf{DB}, i \in \llbracket 1, t \rrbracket :$ |
| Update $S_{\mathsf{ID}} \leftarrow H(S_{\mathsf{ID}})$ | | $G(H^i(K_{\mathsf{ID}}) \Vert a) = c$ |
| | | If entry found, update $K_{\mathsf{ID}} \leftarrow H^i(K_{\mathsf{ID}})$ |
| | | **Output:** ID or $\perp$ if not found |

Figure 8.3: A weakly correct, secure, and Narrow-Destructive private RFID authentication protocol in the random oracle model.

The proof of this theorem can be found in [Vau07].

### 8.4.2 *Narrow-Destructive Privacy from the OSK Protocol*

The next example we present is a modified version of the OSK protocol that includes a randomized challenge sent by the reader. We keep the previous notation and let $\alpha$ and $\gamma$ be two polynomials over $\mathbf{N}$. The scheme further assumes the existence of one random function $G : \{0,1\}^k \times \{0,1\}^{\alpha(k)} \to \{0,1\}^{\gamma(k)}$ and one random permutation $H : \{0,1\}^k \times \{0,1\}^k$, which will be modeled as random oracles. The protocol is shown in Figure 8.3 and described here.

- **Setup.** The system does not use public-key cryptography, so this algorithm is void.
- **SetupTag.** To create a tag ID, pick uniformly a $k$-bit key $K_{\mathsf{ID}}$ and set $S_{\mathsf{ID}} = K_{\mathsf{ID}}$. When the tag is legitimate, the entry $(\mathsf{ID}, K_{\mathsf{ID}})$ is put in the database.
- **Authentication.** Tag authentication is performed by a challenge-response protocol which works as follow.
  - First, the reader picks a $\alpha$-bit string $a$ and sends it to the tag.
  - The latter computes $c = G(K_{\mathsf{ID}} \| a)$ and replies with this value. In parallel, it updates its state by replacing $K_{\mathsf{ID}}$ by $H(K_{\mathsf{ID}})$.
  - Finally, given a fixed threshold $t$, polynomial in the security parameter, the reader looks in its database for a pair $(\mathsf{ID}, K_{\mathsf{ID}})$ that satisfies the equality $c = G(H^i(S_{\mathsf{ID}}) \| a)$ for some $i \in \llbracket 1, t \rrbracket$. Once the entry is found, the database entry is updated and $K_{\mathsf{ID}}$ is replaced by $G^i(S_{\mathsf{ID}})$. The protocol instance yields ID. If no entry is found, the protocol ends in failure and the reader outputs $\perp$.

**Theorem 8.2**
*If $2^{-\alpha}$ and $2^{-\gamma}$ are negligible in the security parameter $k$, then the scheme depicted in Figure 8.3 describes a weakly correct, secure, and Narrow-Destructive private RFID scheme in the random oracle model.*

| **Tag** | | **System** |
|---|---|---|
| **State:** $pk, \mathsf{ID}, K_{\mathsf{ID}}$ | | **Secret key:** $sk$<br>**Database:** $\{\dots, (\mathsf{ID}, K_{\mathsf{ID}}), \dots\}$ |
| | $\xleftarrow{\quad a \quad}$ | Choose $a \in_R \{0,1\}^\alpha$ |
| $c = \mathsf{Enc}_{pk}(\mathsf{ID} \Vert K_{\mathsf{ID}} \Vert a)$ | $\xrightarrow{\quad c \quad}$ | Check that $\exists(\mathsf{ID}, K_{\mathsf{ID}}) \in \mathsf{DB} : \mathsf{Dec}_{sk}(c) = \mathsf{ID} \Vert K_{\mathsf{ID}} \Vert a$ |
| | | **Output:** ID or $\perp$ if not found |

**Figure 8.4:** A correct, secure, Narrow-Strong, and Forward private RFID authentication protocol based on an IND-CCA2 public-key encryption scheme.

We remark that the scheme's privacy fails as soon as the adversary gains the ability to query a RESULT oracle as it was showed by Juels and Weis [JW07]. To perform the attack, the adversary desynchronizes a tag with the reader by interacting with it $t+1$ times causing a denial of service. Having access to the result of the protocol then allows the adversary to later distinguish that tag from any other legitimate one. The attack is best described by the following algorithm.

```
CREATETAG($\mathsf{ID}_0$)
$\mathsf{vtag}_0 \leftarrow$ DRAWTAG($\mathsf{ID}_0$)
Simulate the reader during $t+1$ consecutive sessions
FREE($\mathsf{vtag}_0$)
CREATETAG($\mathsf{ID}_1$)
$\mathsf{vtag} \leftarrow$ DRAWTAG($\Pr[\mathcal{T}(\mathsf{vtag}) = \mathsf{ID}_0] = \Pr[\mathcal{T}(\mathsf{vtag}) = \mathsf{ID}_1] = \frac{1}{2}$)
$(\pi, \tau) \leftarrow$ EXECUTE($\mathsf{vtag}$)
$b \leftarrow$ RESULT($\pi$)
Output $b$.
```

Therefore, the scheme does not meet Weak privacy.

### 8.4.3 *Narrow-Strong and Forward Privacy Using Public-Key Encryption*

To achieve higher levels of privacy, Vaudenay assumes a public-key cryptosystem (KeyGen, Enc, Dec) implemented in the tags. The protocol in itself is rather simple. It is a two-pass challenge-response protocol defined as follows.

- **Setup.** This algorithm calls upon the encryption scheme's KeyGen with the same security parameter to produce a key pair $(sk, pk)$ that is forwarded as the output.
- **SetupTag.** To create a tag ID, pick uniformly a $k$-bit key $K_{\mathsf{ID}}$ and set $S_{\mathsf{ID}} = (pk, \mathsf{ID}, K_{\mathsf{ID}})$. When the tag is legitimate, the entry $(\mathsf{ID}, K_{\mathsf{ID}})$ is put in the database.
- **Authentication.** The protocol is shown in Figure 8.4 and detailled here.
  - First, the reader picks an $\alpha$-bit string $a$ and sends it to the tag.
  - The latter computes $c = \mathsf{Enc}_{pk}(\mathsf{ID} \Vert K_{\mathsf{ID}} \Vert a)$ and replies with this value.

- Upon reception of $c$, the reader decrypts it, parses the plaintext as $\mathsf{ID}\|K_{\mathsf{ID}}\|a'$, and checks the correctness of the challenge, i.e., whether $a' = a$. In case of failure, the reader aborts and outputs $\perp$. In the other case, it looks in its database for the pair $(\mathsf{ID}, K_{\mathsf{ID}})$ and outputs $\mathsf{ID}$ in case of success. Otherwise, the reader outputs $\perp$.

It is possible to transform this RFID scheme into a system that does not require any database server. This is attained by tweaking the setup and adding a random key $K$ of a PRF to its secret key. Tag creation is also altered so that instead of picking $K_{\mathsf{ID}}$ randomly, it is computed by applying the PRF keyed with $K$ on input $\mathsf{ID}$, that is $\mathsf{SetupTag}$ sets $K_{\mathsf{ID}} = F_K(\mathsf{ID})$. As long as $F$ is a good PRF, the later RFID system is indistinguishable from the former one. Therefore, all the results that follow are still valid for this variant.

**Theorem 8.3**
*Assuming the public-key encryption scheme to be correct and IND-CPA secure, the protocol of Figure 8.4 describes a correct and Narrow-Destructive private RFID scheme. Furthermore, if the encryption scheme is IND-CCA2 secure, then the scheme is secure and Forward private.*

## 8.5 Comparison with the extended-Juels-Weis Model

As we mentionned before, the initial privacy model of Juels and Weis [JW07] considers neither correctness nor security. This hole was filled by Damgård and Pedersen [DP08] which added a definition for completeness and two notions for security, named weak and strong soundness. Essentially, strong soundness is equivalent to our security requirement and weak soundness would be our same security property but restricted to Weak adversaries. Completeness corresponds to our weak correctness. Note that weak soundness is not taken into account by both Vaudenay's and our models as we consider the eventuality that an adversary succeeds in obtaining a tag's secret.

Privacy in the eJW model is based on the notion of indistinguishability: a scheme is supposed to preserve privacy if no adversary can guess the identity of given a tag, secretly chosen among a pair of tags selected by the adversary herself. In this section, we show that Vaudenay's weakest class of privacy with access to the result of the protocol, i.e., the class of Weak adversaries, provides a stronger notion of privacy than the one of Juels and Weis.

**Theorem 8.4**
*If an RFID scheme is weakly-correct, secure, and Weak private in our model then it is correct, sound, and private in the eJW model.*

**Proof.** After reducing eJW's privacy experiment to simple privacy, the proof is straightforward when rewrote in our notation.

```
1: $\textsc{CreateTag}(\mathsf{ID}_1), \ldots, \textsc{CreateTag}(\mathsf{ID}_n)$
2: $\forall i \in [\![1, n]\!] : \mathsf{vtag}_i \leftarrow \textsc{DrawTag}(\mathsf{ID}_i)$
```

```
Interact with $\mathsf{ID}_1, \ldots, \mathsf{ID}_n$ through the LAUNCH, SENDTAG, SENDREADER, and RESULT interfaces.
$\forall i \in [\![1, n]\!] : \textsc{Free}(\mathsf{vtag}_i)$
Pick two integers $x, y \in [\![1, n]\!]$
$\mathsf{vtag} \leftarrow \textsc{DrawTag}(\Pr[\mathcal{T}(\mathsf{vtag}) = \mathsf{ID}_x] = \Pr[\mathcal{T}(\mathsf{vtag}) = \mathsf{ID}_y] = \frac{1}{2})$
$\forall i \in [\![1, n]\!] \setminus \{x, y\} : \mathsf{vtag}_i \leftarrow \textsc{DrawTag}(\mathsf{ID}_i)$
Interact with vtag and all the $\mathsf{vtag}_i$'s through the LAUNCH, SENDTAG, SENDREADER, and RESULT interfaces.
Output a guess $b \in \{x, y\}$.
Get the table $\mathcal{T}$ and output whether $\mathcal{T}(\mathsf{vtag}) = \mathsf{ID}_b$.
```

Recall that a scheme preserves privacy in the sense of Juels and Weis if and only if there exists an adversary $\mathcal{A}$ and a non-negligible advantage $\epsilon$ such that

$$\left|\Pr[\mathcal{A} \to 1] - \frac{1}{2}\right| = \epsilon.$$

At the same time, every blinded adversary wins the privacy experiment with probability $^1/_2$ (That is because a blinded adversary needs to guess the real identity of a tag without having access to any real protocol message). Hence, we deduce that

$$|\Pr[\mathcal{A} \to 1] - \Pr[\mathcal{A}_B \to 1]| = \epsilon,$$

which implies that the scheme does not provide Weak privacy. □

Concerning the other direction, whether privacy in the eJW model implies Weak privacy, we use the fact that any protocol achieving privacy in the zk-privacy model is private in the eJW model [DLYZ10]. Since we show in the next section that the former model does not imply our notion of Weak privacy, the eJW model is strictly weaker than our Weak class of privacy.

## 8.6 ZK-Privacy does not Imply Narrow-Weak Privacy

In this section, we present a generic transformation of a zk-private protocol to another protocol preserving zk-privacy but for which we can exhibit an effective Narrow-Weak adversary. For this transformation, we use pseudo-random functions.

Let us consider a correct, secure, and zk-private RFID authentication scheme augmented in the following way: Along with the original internal states required for authentication, two special tags, denoted 0 and 1 hereafter, are assumed to hold two values $K$ and $\beta$ in their memory. Upon set up, these tags have their $K$ set to a common and random value while their $\beta$ is set to their ID. Furthermore, we define the five special messages for all the tags which implement the protocol of Fig. 8.5.

| **Tag 1**<br>**Augmented State:** $K, \beta$ | | **Tag 2**<br>**Augmented State:** $K, \beta$ |
|---|---|---|
| | $\xleftarrow{L}$ | |
| pick $a$ | $\xrightarrow{R:a}$ | pick $b$ |
| if $x = F_K(a, b)$ then $c \leftarrow \beta$ | $\xleftarrow{L_2:b,x}$ | Compute $x \leftarrow F_K(a, b)$ |
| otherwise pick random $c$ | $\xrightarrow{c}$ | |

**Figure 8.5:** Augmented protocol for RFID tags.

- $[L]$ is a protocol message for the tag upon which the latter responds with a $[R : a]$ message, $a$ being a random value. The value of $a$ is kept until the tag is cleaned, i.e., until the protocol finishes.
- $[R : a]$ is a protocol message for the tag that triggers the following procedure: if the tag has a stored $K$, then it picks a value $b$ at random and compute $x = F_K(a, b)$ where $F_K$ is a pseudo-random function. After that, it erases $K$ and $\beta$ from its memory. The tag then answers by a message $[L_2 : b, x]$. Note that if no $K$ is stored, then the tag does the same with a random $K$. At the end of the procedure, the tag stays clean.
- $[L_2 : b, x]$ is a protocol message for the tag which sent an $[R : a]$ message. If the tag holds a pair $(K, \beta)$ and $x = F_K(a, b)$, it sets $c = \beta$, otherwise it sets $c$ to a random bit, erases $K$ and $\beta$. The tag answers $c$ and is cleaned.
- $[L_S : \beta]$ is special variant of $[L_2 : b, x]$ which makes the tag answer $\beta$ and erase any existing $(K, \beta)$. It also cleans the tag.
- $[R_S : K, \beta, a]$ is a special variant of the $[R : a]$ message in which the tag starts by replacing its current $K$ and $\beta$ by the ones contained in the message. It continues with the procedure of $[R : a]$.

Note that since the augmented messages and associated keys are not used for authentication, the augmented scheme inherits the correctness and security of the original one.

Furthermore, the augmented scheme can be shown to still be zk-private: given an adversary $\mathcal{A}$, we construct a simulator $\mathcal{S}$ as follows: The simulator first simulates $\mathcal{A}$ until he submits his set of tags $\mathcal{C}$. During the simulation, $\mathcal{S}$ has two additional tasks to perform. First, it has to analyze the queries to figure out whether the tags have a key stored or not. Then, it needs to check if an unerased key $K$ in any tag is known by $\mathcal{A}$. That is, if there is corruption or a special query to reset $K$, $K$ becomes known and remains so until it is erased. As soon as the set of tags $\mathcal{C}$ is released, there are several cases to consider.

1. $\mathcal{C}$ is empty.

   The simulation of $\mathcal{A}$ can go on. There is no query from any anonymous tag to simulate.
2. $\mathcal{C}$ contains both tags. However, $\mathcal{A}$ can only play with only one of them. Since the tags must be clean and uncorrupted, either they have erased their state or their key is

unknown.

The simulator picks a random key $K$ and simulate the anonymous tag as if it had this key. Due to the PRF property, this simulation is indistinguishable.

3. $\mathcal{C}$ contains a single tag and its state is known. (Namely: either the other tag was corrupted or its state was erased.)

   The simulation can be performed easily as the state of the anonymous tag is known.

4. $\mathcal{C}$ contains a single tag, its state is unknown and not synchronized. (Namely, either the other tag has erased its state or reset it.)

   The simulator picks a random key $K$ and simulate the anonymous tag as if it had this key. Due to the PRF property, this simulation is indistinguishable as even if the adversary got hold of an $(a, b, x)$ triplet, she cannot force the pair $(a, b)$ to be reused.

5. $\mathcal{C}$ contains a single tag, its state is unknown and is synchronized. Note that $\beta$ is known.

   The simulator acts in the same way except when it comes to interact with the non-anonymous tag : The synchronized key is set to the selected one by using either the $[L_S : \beta]$ or the $[R_S : K, \beta, a]$ variant.

The scheme is not Narrow-Weak private and illustrate an adversary that successfully defeats the privacy of the scheme. Our adversary only creates the two special tags 0 and 1. After that, she draws the two tags in random order, e.i., she does not know who is who. She then makes them play the special protocol which reveals $\beta$. At the end, and after receiving the table $T$, the adversary outputs $1$ if the virtual tag that received the $L_2$ message has identity $\beta$. By construction, this adversary always outputs 1. However, when blinded, the output is $1$ with probability $1/2$. Hence, the adversary is significant and the scheme is not Narrow-Weak private.

Therefore, under the assumption that pseudo-random functions exist, there are zk-private correct and secure schemes which are not Narrow-Weak private. The reason of this is the fact that the notion of zk-privacy does not address concurrency, i.e., the adversary is not allowed to interact with two anonymous tags simultaneously. Since the eJW model [JW07] is weaker than zk-privacy [DLYZ10], this result also applies to their model.

## 8.7 Hermans et al.'s Variant

Recently, Hermans et al. [HPVP11] proposed a variation of Vaudenay's model that did not make use of the notion of blinders and trivial adversaries. Considering these notions to be relatively unused in cryptography, they proposed a more traditional approach based on left-or-right indistinguishability, i.e., indistinguishability between two tags. The model, that we denote HPVP hereafter, retains Vaudenay's definitions of RFID system, correctness and security. However, adversaries for privacy and security differ, the difference lying in the DrawTag interface for adversaries playing the privacy game. Instead of drawing tags randomly sampling following a distribution specified by the adversary, all DrawTag queries get only two tags as input and, according to a bit $b$ chosen a the beginning of the privacy game, the challenger consistently returns either the first or the second tag. Note that the adversary can still choose to target a single tag $\mathsf{ID}$ by specifying the same tags for both arguments of DrawTag, i.e., calling DrawTag$(\mathsf{ID}, \mathsf{ID})$.

**Definition 8.12 (Adversaries against privacy in the HPVP variant)**
*The HPVP model, adversaries are defined as in Vaudenay's model, i.e., following Definition 8.3, except for DrawTag, Free and Corrupt that are altered in the following manner.*

- *The CreateTag interface only creates legitimate tags.*
- *The DrawTag interface takes only two RFID tags for its input and is dependent on a secret bit b. That is, on input two RFID tags* $\mathsf{ID}_0$ *and* $\mathsf{ID}_1$*, DrawTag*$_b(\mathsf{ID}_0, \mathsf{ID}_1) \rightarrow \mathsf{vtag}$ *consistently returns either* $\mathsf{ID}_0$*, if* $b = 0$ *or* $\mathsf{ID}_1$*, if* $b = 1$*. It also inserts the triplet* $(\mathsf{vtag}, \mathsf{ID}_0, \mathsf{ID}_1)$ *in a table* $\mathcal{D}$*. Upon calling Free*$(\mathsf{vtag})$*, the corresponding entry in* $\mathcal{D}$ *is removed. Note that if one of DrawTag's arguments is a tag* $\mathsf{ID}$ *that is listed in* $\mathcal{D}$*, then the interface returns* $\perp$.
- *Corruption queries are made on tag* $\mathsf{ID}$*s and not on virtual identifiers. That is, to corrupt a tag* $\mathsf{ID}$*, an adversary would query Corrupt*$(\mathsf{ID})$ *(Instead of drawing it first and then calling Corrupt with the received* $\mathsf{vtag}$*).*

As for the Vaudenay model, the HPVP variant considers four classes of adversaries (Strong, Destructive, Forward and Weak) and their four Narrow counterparts, that have the same restrictions as the ones listed in Definition 8.4. The privacy game is defined as the inability to guess DrawTag's bit $b$ with a significant advantage. The HPVP model considers two flavors of privacy depending on whether the definition holds for polynomial-time or computationally unbounded adversaries. For simplicity, we consider only polynomial-time adversaries and we only deal with the former type of privacy.

**Definition 8.13 (Computational Privacy in the HPVP Model)**
*Let be the following privacy experiment played by a polynomial-time adversary* $\mathcal{A}$*.*

```
1: Pick a bit $b \in_R \{0,1\}$
2: Invoke SetupReader$(1^k) \rightarrow (sk, pk)$
3: Execute $\mathcal{A}(pk) \rightarrow b'$ with the DrawTag interface parametrized with b.
4: Output whether $b = b'$
```

*The scheme is said to be P private if for every adversary belonging to class P, it holds that*

$$\mathsf{Adv}^{\mathcal{A}}_{\mathsf{priv}}(k) = \Pr[\mathcal{A} \rightarrow 1] - \frac{1}{2} = \mathsf{negl}(k)$$

By being free from the notion of blinders, the variant has the benefit of simplifying security reductions as many cryptographic functionalities are defined in the same terminology. Examples of such functionalities include pseudo-random functions and encryption schemes.

For instance, the proof of the Narrow-Strong privacy of the scheme shown in Fig. 8.4 and instanciated by an IND-CPA public-key encryption scheme is easier to conduct if the adversary accesses ciphertexts coming from one of two chosen tags. (This easily maps to the IND-CPA game where the adversary receives the encryption of one of two messages she submits.) Moreover, they prove that Strong privacy is reachable in their model by using a challenge-response protocol based on an IND-CCA2 public-key encryption scheme.

However, we note that the simplifications introduced to Vaudenay's model causes substantial differences. First, the adversary is not allowed anymore concurrent interaction with two anonymous tags. This leads to the existence of a protocol similar to the one presented in Section 8.6 for ZK-privacy that is private in their model and fails in ours. Second, only known tags can be corrupted, i.e., the adversary is not allowed to get hold of a random tag and learn its state. However, practical attack scenarios do not obey to this restriction. Finally, the HVPV model does not tolerate the existence of illegitimate tags. As we stated before, in practice, these are tags that follow the same protocol specs but belong to another system. We believe that it is important to address the existence of such tags and the possibility of them communicating with a reader not belonging to their system. Moreover, no privacy is possible in the HPVP model if any of the last two restrictions are waived, i.e., if the adversary can corrupt anonymous tags or if illegitimate tags are introduced in the model.

9

# ACHIEVING STRONG PRIVACY

Contents

As it was detailled in Section 8.3.2, Vaudenay's notion of Strong privacy is impossible to achieve. This result is due to the fact that adversaries are able to send queries to which they do already know the answer. From their side, blinders are unable to produce the exact same answer that the RFID system would compute, and that the adversary expects unless they are able to deduce that information from previous queries, which in itself results in a loss of privacy. However, looking back at the proof we described in Section 8.3.2, it becomes apparent that the adversary did not break the privacy of the scheme in the sense that he did not use protocol messages to compute its final statement. Still, this adversary has been shown to be significant. Therefore, we claim that Vaudenay's definitions do not correctly mirror the notion of privacy it aims to capture.

This chapter is devoted to discuss solutions and tweaks in the model that aim to overcome this limitation. We first sketch Ng et al.'s proposal [NSMSN08] of denying the adversary from issuing queries for which "they already know the answer" and show that the formalism given for this statement is not satisfactory.

We then proceed with our fix and argue that it reflects the exact notion of privacy Vaudenay aimed to capture. Our solution consists of merging the blinder and the adversary, i.e., having blinded adversaries simulating protocol messages for themselves. Concretely, this translates into giving to the blinders access to all the adversary's inputs, including its random tape, which was missing from the original definition. We introduce other limitations to the sampling queries of the adversary, rendering them such that they are "aware" of their samplings (Essentially, we require that adversaries can produce a *plausible* guess on the real identity of a drawn tag). The benefit of all these modifications is that Strong privacy becomes achievable using a challenge-response protocol built on a plaintext-aware public-key encryption scheme.

To show that this notion is almost necessary, we give a counter-example to prove that the same protocol instantiated by an IND-CCA2, but not plaintext-aware, cryptosystem is not Strong private.

## 9.1 Ng et al's Proposal: Wise Adversaries

Starting from the observation that the Destructive adversary in the impossibility proof of Strong privacy is already aware of the answer the genuine Result interface would produce and only uses to distinguish in which world she is, Ng et al. [NSMSN08] proposed to fix the model by disallowing such queries. For this sake, they introduced the notion of "wise" adversaries. A wise adversary is defined as an adversary who does not issue oracle queries for which he "knows" the output. The main argument of [NSMSN08] is the following: if the adversary is wise, then he will never ask the interfaces about the outcome of a protocol session in which he was either passive, active, or simulating a tag if he knows the result of the instance. In this scenario, the universal adversary used by Vaudenay against Strong privacy becomes "unwise" and is thus discarded.

Although they claim to keep Vaudenay's framework and definitions, the way to prove privacy is not resolved in [NSMSN08]. Following their definition, an adversary $\mathcal{A}$ making $q$ (any) Oracle accesses is wise if no adversary can achieve the same or a smaller probability of success while making less than $q$ Oracle calls. It turns out that wisdom is a hard notion to manipulate and difficult to prove.

Another issue is whether the notion "wise adversaries" fits in realistic scenarios. One may argue that this kind of adversary seems equivalent in terms of result but, in fact, it is not clear why would an adversary deny himself such advantage. In fact, this comes back to the definition of knowledge: what does it mean for an algorithm to know something.

## 9.2 Our Proposal: Incorporate the Blinder into the Adversary

The solution we propose differs from the one proposed by Ng et al. and the others described in the end of Chapter 8 in that we do not alter the privacy game. In fact, modifying the privacy game in those previous works provoked a loss in the privacy notion captured by the model which unfortunately is not quantified. For instance, we can assume that an adversary defeats privacy by being able to determine if one RFID tag belongs a group of them that had prior communication with the reader. While it is possible that such statement is included in their definitions, it is not clear from their work.

Our proposal is to make the blinder's simulation run inside the adversary. That is, we argue that a blinder, acting for the adversary and not for the system, as Vaudenay's definitions suggest, should be executed by the former. Consequently, the blinder should be given all the adversary's knowledge, and in particular her random tape that was missing from the original definitions.

Before going into the modifications we propose to Vaudenay's definitions, we dedicate the next three sections to introduce new concepts that will be proved to be later useful.

## 9.3 Sampling Algorithms and the ISH Hypothesis

Up to this point, we never fully defined sampling algorithms but merely treated them as algorithms implementing probability distributions. This section looks more deeply in the subject.

**Definition 9.1 (Sampling Algorithm)**
*An efficient sampling algorithm for a probability distribution* $\mathrm{p}$ *is a polynomial-time probabilistic algorithm, in* $k$*, denoted* $\mathsf{Samp}$*, that, on input random coins* $\rho \in \{0,1\}^{\ell(k)}$*, with* $\ell(\cdot)$ *being a polynomial function, outputs vector elements from* $X$ *of dimension* $d(k)$*, with* $d$ *also being a polynomial function, that satisfies*

$$\forall x \in X^d : \quad |\Pr_{\rho}[\mathsf{Samp}(\rho) = x] - p(x)| = \mathsf{negl}(k)$$

The definition above only considers computational closeness from the original distribution. Although we might have considered statistical or perfect distance, the reason of this restriction is that we will only be interested in sampling requests by polynomial-time algorithms. In this context, extending to computational distance can only enlarge the set of distributions that an algorithm can submit without affecting the security proof of the scheme.

We also note that the restriction to polynomial-time algorithms for describing the sampling algorithm is due to security considerations: It is often the case in security reductions that the whole environment adversary+system has to be executed by an adversary playing a classical cryptographic game, such as IND-CCA2 or distinguishing a PRF from a random function. Although Vaudenay overlooked this matter, we find it necessary for the proof of security for simple and weakly-correct RFID schemes and the proofs of privacy of the Weak private protocol based on a PRF, and the Narrow-Strong private one based on an IND-CPA public-key encryption scheme.

However, being able to simulate the adversary and her environment is not always sufficient for the security proof. In many settings, it is the case that the simulator needs to obtain the randomness of the system. For instance, this happens in complex zero-knowledge systems where the simulator would need the random tape of the whole system. Damgård first mentioned this limitation when he considered adaptive corruption in multi-party computation schemes [Dam92]. As a solution he had to restrict the adversary to so-called "good-enough" distributions. A more formal treatment of the problem was subsequently presented by Canetti and Dakdouk [CD08]. Concretely, they proposed the notion of inverse-samplable algorithms which is centered around the idea that for every possible output of an algorithm, it is possible to efficiently find, i.e., in a polynomial number of steps, a randomness that leads to the same output.

In the sequel, we will be interested in a more specific class of sampling algorithms called inverse-sampling algorithms. An algorithm Samp is said to be inverse-samplable if there exists a polynomial-time inversion algorithm which, given a sample $x$ from the output of Samp, obtained using random coins $\rho \in \{0,1\}^{\ell(k)}$, with $\ell(\cdot)$ being a polynomial function, outputs a $\rho_S$ that is consistent with $x$, i.e., it is such that $\mathsf{Samp}(\rho_S) \to x$. Moreover, the choice of $\rho$ has to be such that $(\rho_S, x)$ is computationally indistinguishable from $(\rho, x)$ for a $\rho$ uniformly distributed over $\{0,1\}^{\ell(k)}$. We hereafter state the formal definition of such sampling algorithms, as given by Ishai et al. [IKOS10].

**Definition 9.2 (Inverse-Sampling Algorithm)**
*Given a security parameter $k$, we say that an efficient sampling algorithm* Samp*, in $k$, is inverse-samplable if there exists a polynomial-time inverter algorithm* $\mathsf{Samp}^{-1}$*, in $k$, such that the following two games are indistinguishable*

$$\begin{array}{l|l}
\textit{Real game:} & \textit{Fake Game:} \\
\quad \rho \in_R \{0,1\}^{\ell(k)} & \quad \rho \in_R \{0,1\}^{\ell(k)} \\
\quad x \leftarrow \mathsf{Samp}(\rho) & \quad x \leftarrow \mathsf{Samp}(\rho) \\
 & \quad \rho_S \leftarrow \mathsf{Samp}^{-1}(x) \\
\quad \textit{Output } (\rho, x) & \quad \textit{Output } (\rho_S, x)
\end{array}$$

*That is, for every polynomial-time distinguisher $\mathcal{D}$ we require that*

$$\left|\Pr[\mathcal{D}^{\textit{Real Game}}(1^k) \to 1] - \Pr[\mathcal{D}^{\textit{Fake Game}}(1^k) \to 1]\right| = \mathsf{negl}(k)$$

**Definition 9.3 (Inverse-Sampling Hypothesis)**
*The inverse sampling hypothesis is that for every probability distribution there exists an inverse-samplable algorithm.*

This hypothesis states that for every sampling algorithm $\mathsf{S}_1$, including one-way sampling algorithms, there exists an inverse-sampling algorithm $\mathsf{S}_2$ that can be shown to be indistinguishable from $\mathsf{S}_1$. The analysis of ISH by Ishai et al. [IKOS10] shows that the existence of non-interactively extractable one-way function family ensembles, a generalization of knowledge assumptions, and non-interactive zero-knowledge proof systems for $\mathcal{NP}$ in the common reference string model together imply that ISH does not hold. An interesting side effect of this result is that the existence of plaintext-aware encryption schemes and the validity of the ISH hypothesis are mutually exclusive. This is a direct consequence of the fact that plaintext-aware encryption schemes require knowledge extractors, by definition (cf. Definition 9.5), and that non-interactive zero-knowledge proof systems for $\mathcal{NP}$ in the CRS model can be constructed from any trapdoor one-way permutation [FLS90]. As we will later make use of plaintext-aware encryption schemes, we are obliged to make the assumption that ISH is wrong.

## 9.4 Knowledge Extractors and Non-Falsiable Assumptions

The notion of knowledge and awareness for interactive Turing machines are defined in terms of computations. That is, a machine is said to know $x$ if it is able to compute $f(x)$ for an arbitrarily chosen function $f$. Formalizing this notion has proven to be one of the most difficult tasks of theoretical computer scientists. In the end, the agreed definition is that a Turing machine knows $x$ if there exists another Turing machine that runs in the same complexity class as the former and takes its description along with all its inputs and outputs $x$. This last machine is called a knowledge extractor.

To be more concise, we give a concrete example with extractable one-way functions which were introduced by Canetti and Dakdouk [CD08]. Besides complying to the classical one-wayness property, such a function has to be such that the "only" way for an algorithm to output an element that has a pre-image by this function is to pick an element from the domain of $f$

and apply the function on it. Again, the term "only way" is formalized by requiring the existence, for every algorithm $\mathcal{A}$, of a knowledge extractor that, having access to all $\mathcal{A}$'s knowledge, i.e., its random tape and a reference to the function that $\mathcal{A}$ targeted, either outputs a preimage of $\mathcal{A}$'s output or fails if the later is not in the image of the function. The reason for combining those two notions in one primitive is that it yields a natural abstraction of several knowledge extractor from the literature [Dam92, BP04a, PX09], in much the same way as the notion of one-way function is an abstraction of the Discrete Log assumption. We give the following formal definition taken from [CD08].

**Definition 9.4 (Extractable One-Way Function Family Ensemble)**
*Let $f : K \times D \to R$ be a family of one-way functions with respect to a security parameter $k$. We say that $f$ is an extractable one-way function family if it is one-way and for every PPT algorithm $\mathcal{A}$ that uses $\rho(k)$ random bits, there is a PPT extractor algorithm $\mathcal{A}^\star$ such that*

$$\forall k \in \mathbf{N} : \Pr\left[\begin{array}{c} y \notin \mathsf{Img}(f_\kappa) \\ \vee \\ f(x) = y \end{array} \middle| \begin{array}{c} \kappa \in_R K \\ \rho \in_R \{0,1\}^{\rho(k)} \\ y \leftarrow \mathcal{A}(\kappa; \rho) \\ x \leftarrow \mathcal{A}^\star(\kappa, \rho) \end{array}\right] = 1 - \mathsf{negl}(k)$$

Unfortunately, as for one-way functions, the existence of extractable one-way functions and knowledge extractors can only be assumed (and even independently from the assumption that one-way functions exist).

The first assumption in the literature related to the existence of knowledge extractors is due to Damgård [Dam92] and called the Diffie-Hellman Key (DHK) assumption (it has also been termed the knowledge of exponent assumption by Bellare and Palacio [BP04a]). In short, this assumption states that the only mean for an adversary that is given an element $W$ from a cyclic group in which $g$ is a generator and wants to produce a valid Diffie-Hellman tuple $(W, g^u, W^u)$ is to pick $u$ and that there exists an extractor that given the adversary's input and randomness recovers $u$. Although it was used in numerous applications, it is not clear whether the assumption is true or false. Moreover, the assumption presents the particularity of being as hard to prove than to disprove and has consequently been the target of many criticism [Nao03]. That is, it is insufficient to construct a counter-example to invalidate the DHK assumption as it would be the case for classical computational assumptions such as the discrete logarithm or factoring. In fact, to prove that the assumption does not hold one would need to prove that there exists an adversary for which there is no extractor. Yet, some variants of the DHK assumption were deemed to be false [BP04a]. Defenders of the assumption argue that it is proven to hold in the generic group model [Den06b]. However, much like the random oracle model, some computational assumptions hold in the generic group model but fail as soon as the group is instantiated in any representation [Den02]. That said, much like the random oracle model, no "concrete" example for the separation is currently known.

The DHK assumption was later expanded to cover general subset membership problems by Birkett [Bir10]. He called that generalization the subset witness knowledge (SWK) assumption. Based on this assumption, he was able to extend (and correct some parts of) Dent's proof that the Cramer-Shoup cryptosystem in a Diffie-Hellman group based on the DDH assumption is plaintext-aware [Den06a] to cover more underlying groups and assumptions such as groups in which the quadratic residuosity or the higher residuosity problem is hard.

## 9.5 Plaintext-Awareness

### 9.5.1 *Definitions*

Plaintext-awareness roughly states that if an adversary is able to produce a valid ciphertext different from $\perp$, then she should know the corresponding plaintext $m$. This translates into saying that, for a plaintext-aware encryption scheme, the "only way" for this ciphertext creator to produce a valid ciphertext is to encrypt a known message $m$ with the public key $pk$.

Formalizing this notion has proven to be a non-trivial task and has been the subject of several papers [BDPR98, BP04b, BR95a, BD08b, Den06a]. In the end, several and separate levels of plaintext-awareness were defined, namely, in increasing strength, PA1, and PA2, and their counterparts PA1+ and PA2+.

The difference between PA1 and PA2 lies in the attacker's ability to get hold of ciphertexts for which she does not know the decryption. In the settings of PA2, this ability is implemented by an oracle $\mathcal{P}(\mathsf{aux})$, called plaintext creator, that, on each query, picks a message at random (or possibly according to a distribution partially defined by its input $\mathsf{aux}$) and returns its encryption, i.e., it produces $\mathsf{Enc}_{pk}(\mathcal{P}(\mathsf{aux}))$. Any ciphertext obtained through this oracle is added to a list $\mathsf{CList}$, the list of ciphertexts for which the adversary does not know the corresponding plaintexts. An adversary $\mathcal{A}$, called ciphertext creator, interacts with the plaintext creator and outputs ciphertexts to be submitted to a decryption oracle. The essence of plaintext-awareness is the existence of a polynomial-time algorithm $\mathcal{A}^\star$, whose construction may depend on $\mathcal{A}$, called plaintext extractor that successfully decrypts any ciphertext given by the adversary that was not returned by the plaintext creator. In order to carry out the extraction, $\mathcal{A}^\star$ is given the view of $\mathcal{A}$ (which includes $\mathsf{CList}$ and the random coins of $\mathcal{A}$) and the target ciphertext $c$ to be decrypted. The initial definition of plaintext-awareness implies that $c$ should not be in $\mathsf{CList}$ as the adversary does not "know" the decryption of ciphertexts returned by $\mathcal{P}$. A scheme is thus said to be PA2 plaintext-aware if, for every polynomial-time ciphertext creator $\mathcal{A}$, there exists a plaintext extractor $\mathcal{A}^\star$ that is, for all plaintext creators $\mathcal{P}$, indistinguishable from a decryption oracle. Since PA1 adversaries have no ciphertext creator at disposal, PA1 plaintext-awareness only requires indistinguishability between the knowledge extractor and the decryption oracle with respect to adversaries who have no access to $\mathsf{Enc}_{pk}(\mathcal{P}(\cdot))$.

In order to capture any external knowledge the adversary can access, Dent [Den06a] extended PA1 to PA1+ for adversaries who can get hold of uniformly distributed bits from an external source. Later, Birkett and Dent [BD08b] introduced the analog notion of PA2+ for

PA2 plaintext-awareness. These last two notions were proven to be equivalent under the condition that the encryption scheme is IND-CPA [Bir10].

**Definition 9.5 (Plaintext-Aware Encryption)**
*Let $\mathcal{O}_1$ denote an oracle that on each query returns a single uniformly distributed bit. We say that a public key cryptosystem* $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ *is PA2+ plaintext-aware if, considering a ciphertext creator $\mathcal{A}$, a plaintext extractor $\mathcal{A}^\star$, a plaintext creator $\mathcal{P}$, and a distinguisher $\mathcal{D}$, all being polynomial-time algorithms, we have*

$$\forall \mathcal{A}, \exists \mathcal{A}^\star, \forall \mathcal{P}, \forall \mathcal{D} :$$

$$\Big| \Pr\Big[\mathcal{D}^{\mathcal{A}^{\mathsf{Enc}_{pk}(\mathcal{P}(\cdot)),\mathsf{Dec}_{sk}(\cdot),\mathcal{O}_1}(pk)}(1^k) \to 1 \Big| (sk, pk) \leftarrow \mathsf{KeyGen}(1^k)\Big]$$
$$- \Pr\Big[\mathcal{D}^{\mathcal{A}^{\mathsf{Enc}_{pk}(\mathcal{P}(\cdot)),\mathcal{A}^\star(pk,\cdot,\mathsf{view}_{\mathcal{A}}),\mathcal{O}_1}(pk)}(1^k) \to 1 \Big| (sk, pk) \leftarrow \mathsf{KeyGen}(1^k)\Big]\Big| = \mathsf{negl}(k).$$

*We also consider three variants.*

- *If the previous equality holds when the ciphertext creator $\mathcal{A}$ does not query $\mathcal{O}_1$, then the scheme is said to be PA2 plaintext-aware.*
- *PA1+ plaintext-awareness is the case in which $\mathcal{A}$ is restricted to make no query to the oracle* $\mathsf{Enc}_{pk}(\mathcal{P}(\cdot))$.
- *When both restrictions are put together, i.e., that $\mathcal{A}$ neither queries $\mathcal{O}_1$ nor* $\mathsf{Enc}_{pk}(\mathcal{P}(\cdot))$, *then the scheme is said to be PA1 plaintext-aware.*

### 9.5.2 *Instances of Plaintext-Aware Encryption Schemes*

As we will later use plaintext-aware encryption schemes in a concrete protocol, we present two schemes that satisfy our security requirements.

**The Cramer-Shoup Encryption Scheme** Not only it is the first truly practical construction of an IND-CCA2 public-key encryption scheme [CS98], but the Cramer-Shoup encryption scheme is also the first cryptosystem to have been proven to be PA1+ and PA2 plaintext-aware [Den06a] (In that work, it was proven treating it as as a KEM instanciated in a Diffie-Hellman group) and later generalized to any suitable group [Bir10]. The scheme works as follow.

- **KeyGen.** On input a security parameter $k$, the algorithm picks a generator $g_1$ of a group $G$ with prime order $q$ and a target collision-resistant hash function $\mathsf{TCR} : G^2 \to \mathbf{Z}_q$. After that, it randomly selects $w, x, y, z \in \mathbf{Z}_q$ and let $g_2 = g_1^w$, $e = g_1^x$, $f = g_1^y$, $h = g_1^z$. The public key is defined as $pk = (g_1, g_2, e, f, h)$ and the corresponding secret key is $sk = (w, x, y, z)$.
- **Encapsulation.** This algorithm works by picking $r \in_R \mathbf{Z}_q$ and computing $c_1 = g_1^r$, $c_2 = g_2^r$, $t \leftarrow \mathsf{TCR}(c_1, c_2)$, and $\pi = e^{rt} f^r$. Letting $C = (c_1, c_2, \pi)$ and $K = h^r$, the output is the pair $(C, K)$.

- **Decapsulation.** To extract $K$, parse $C$ as $(c_1, c_2, \pi)$ and recompute $t \leftarrow \mathsf{TCR}(c_1, c_2)$. If $c_2 = c_1^w$ and $\pi = c_1^{xt+y}$ then output $c_1^w$. Otherwise output $\perp$.

**Theorem 9.1 ([Bir10])**
*Suppose the DDH problem is hard in the group $G$ and the hash function* TCR *is target collision resistant. If the group $G$ is a statistically simulatable group on which the DHK assumption holds, then the Cramer-Shoup KEM is PA1+ plaintext-aware.*

**The Kurosawa-Desmedt Scheme** After the publication of the Cramer-Shoup public-key encryption scheme, Kurosawa and Desmedt proposed a similar but more efficient cryptosystem and proved that it is IND-CCA2 secure [KD04] (but the underlying public-key encryption scheme is proven to not be IND-CCA2 secure [CHH$^+$09]). Independently, Jiang and Wang [JW10] and Birkett [Bir10] showed that the scheme is PA2 plaintext aware. Although the last two works considered a more general variant of the Cramer-Shoup cryptosystem based on hash proof systems [CS02], we present a simplest variant using a group in which the DDH assumption can be assumed to be hard.

- **KeyGen.** On input a security parameter, pick two distinct generators $g_1$ and $g_2$ of a group $G$ with prime order $q$ and a target collision-resistant hash function $\mathsf{TCR} : G^2 \to \mathbf{Z}_q$. After that, randomly select $x_1, x_2, y_1, y_2 \in \mathbf{Z}_q$ and let $g = g_1^{x_1} g_2^{x_2}$ and $h = g_1^{y_1} g_2^{y_2}$. The public key is $pk = (g, h, g_1, g_2)$ and the corresponding secret key is $sk = (x_1, x_2, y_1, y_2)$.
- **Encapsulation.** Pick $r \in_R \mathbf{Z}_q$ and compute $c_1 = g_1^r$, $c_2 = g_2^r$, $t = \mathsf{TCR}(c_1, c_2)$, and $K = c_1^r c_2^{rt}$. Let $C = (c_1, c_2)$ and output $(C, K)$.
- **Decapsulation.** To extract $K$, parse $C$ as $(c_1, c_2)$ and recompute $t = \mathsf{TCR}(c_1, c_2)$. Then recover the key $K = c_1^{x_1+ty_1} c_2^{x_2+ty_2}$.

**Theorem 9.2 ([Bir10, JW10])**
*Suppose the DDH problem is hard in the group $G$ and the hash function* TCR *is target collision resistant. If the group $G$ is a statistically simulatable group on which the DHK assumption holds, then the Kurosawa-Desmedt KEM/DEM is PA1+ and PA2 plaintext-aware.*

### 9.5.3 *From PA+ to PA++ Plaintext-Awareness*

We generalize this notion further and define PA1++ and PA2++ plaintext-awareness for adversaries who can submit sampling request to an external oracle. On one side, efficiency considerations restrict the sampling algorithms to be computable in polynomial-time. On the other side, we will require the system composed of the adversary and its randomness oracle to be simulatable by the plaintext extractor, that is because the plaintext extractor will need to be able to find suitable random coins for $\mathcal{O}_S$. However, as we noted in Section 9.3, assuming the existence of plaintext-aware encryption schemes forces us to assume that the ISH hypothesis is false so the restriction on the sampling algorithms becomes effective. In a general sense, we conjecture that it is impossible to achieve plaintext-awareness for adversaries allowed to submit non inverse-sampling algorithms.

**Definition 9.6 (PA1++ and PA2++ Plaintext-Awareness)**
*Starting from the notions of PA1 and PA2 plaintext-awareness, we define two new conditions, PA1++ and PA2++, by adding one randomness oracle $\mathcal{O}_S$. This oracle takes as input the description of an inverse-sampling algorithm, executes it using its own random tape, and returns its output to the ciphertext creator.*

*We say that a public-key encryption scheme is PA1++, respectively PA2++, plaintext-aware if Definition 9.5 holds for PA1, respectively PA2, adversaries having access to the oracle $\mathcal{O}_S$.*

Note that PA1++ (resp. PA2++) plaintext-awareness trivially implies PA1 and PA1+ (resp. PA2 and PA2+) plaintext-awareness, since the ciphertext creator may simply not use the randomness oracle or just query using a sampling algorithm from the uniform distribution over $\{0, 1\}$. Actually, we can even show that these two notions are equivalent, i.e., any scheme that is PA1+ is PA1++ and any scheme that is PA2+ is PA2++.

**Theorem 9.3**
*Suppose a public key encryption scheme is PA1+ (resp. PA2+) plaintext-aware. Then it is PA1++ (resp. PA2++) plaintext-aware.*

**Proof.** We prove the theorem for the case of PA1++. It can be easily modified so that it applies to PA2++. Let $\mathcal{A}$ be a ciphertext creator for the PA1++ plaintext-aware encryption scheme.

We construct a PA1+ ciphertext creator $\mathcal{B}$ as follows: $\mathcal{B}$ takes input $pk$ and simulates $\mathcal{A}$, forwarding all its decryption queries to the decryption oracle. In order to answer $\mathcal{A}$'s queries to the randomness oracle, $\mathcal{B}$ runs the provided sampling algorithm and query its randomness oracle, that we denote $\mathcal{O}_1$, every time a new random bit is asked for. Clearly, $\mathcal{B}$ terminates in polynomial-time if all samplings can be performed in polynomial-time. Remark that $\mathcal{B}$ does not use any internal randomness besides the one used to initialize $\mathcal{A}$.

Since $\mathcal{B}$ is a valid PA1+ ciphertext creator, we can assert the existence of a plaintext extractor $\mathcal{B}^\star$ indistinguishable from a decryption oracle. We use $\mathcal{B}^\star$ to construct a plaintext extractor $\mathcal{A}^\star$ for $\mathcal{A}$. In the following, we assume that $\mathcal{A}^\star$ maintains a state $\mathsf{view}'$ initialized to $\mathsf{view}_{\mathcal{A}}$ that will be used to simulate $\mathcal{B}$'s view. To answer $\mathcal{A}$'s decryption queries, $\mathcal{A}^\star$ proceeds as follow:

1. If $\mathcal{A}$ queried the randomness oracle with an inverse-sampling algorithm $\mathsf{Samp}$ and received $x$ since the last invocation of $\mathcal{A}^\star$, then $\mathcal{A}^\star$ computes $\rho_S \leftarrow \mathsf{Samp}^{-1}(x)$. After that, $\mathcal{A}^\star$ updates the simulated view of $\mathcal{B}$ to include the random bits $\rho_S$, i.e., it sets $\mathsf{view}' \leftarrow \mathsf{view}' \| \rho_S$. Due to the property of inverse-sampling algorithms, $(\rho_S, \mathsf{view}_{\mathcal{A}})$ is indistinguishable from $(\rho, \mathsf{view}_{\mathcal{A}})$, where $\rho$ is the random string returned by $\mathcal{O}_S$ for the sampling request. Thus, a simple induction argument suffices to show that $\mathsf{view}_{\mathcal{B}}$ and $\mathsf{view}'$ are indistinguishable.

   This procedure is repeated for every new sampling query.

2. $\mathcal{A}^\star$ then calls upon $\mathcal{B}^\star(pk, c, \mathsf{view}')$ and forwards its output to $\mathcal{A}$.

Since $\mathsf{view}_{\mathcal{A}}$ is included in $\mathsf{view}'$ and that this last variable is indistinguishable from $\mathsf{view}_{\mathcal{B}}$,

$$\left|\Pr[\mathcal{D}^{\mathcal{A}^{\mathcal{B}^\star(pk,\cdot,\mathsf{view}'),\mathcal{O}_S(pk)}}(1^k) \to 1] - \Pr[\mathcal{D}^{\mathcal{A}^{\mathcal{B}^\star(pk,\cdot,\mathsf{view}_{\mathcal{B}}),\mathcal{O}_S(pk)}}(1^k) \to 1]\right| = \mathsf{negl}(k).$$

Recalling that $\mathcal{B}^\star(pk, \cdot, \mathsf{view}_{\mathcal{B}})$ is indistinguishable from a decryption oracle to $\mathcal{A}$, we deduce that $\mathcal{A}^\star$ is a valid plaintext extractor. In other words,

$$\left|\Pr[\mathcal{D}^{\mathcal{A}^{\mathcal{B}^\star(pk,\cdot,\mathsf{view}'),\mathcal{O}_S(pk)}}(1^k) \to 1] - \Pr[\mathcal{D}^{\mathcal{A}^{\mathsf{Dec}_{sk}(\cdot),\mathcal{O}_S(pk)}}(1^k) \to 1]\right| = \mathsf{negl}(k).$$

This concludes the proof. □

The following corollary is the combination of Theorem 9.3 with the equivalence result between PA2+ and PA2 [Bir10], under the assumption that the scheme is IND-CPA secure.

**Corollary 9.1**
*If an encryption scheme is IND-CPA and PA2 plaintext-aware, then it is PA1++ plaintext-aware.*

## 9.6 Adapting Vaudenay's Definitions

### 9.6.1 *Limiting the Adversary's Sampling Queries*

In the sequel, we will restrict to adversaries who use distributions to the DrawTag such that, at any step, the table $\mathcal{T}$ can be successfully simulated by an algorithm that is only given the view of the adversary as input. That is, we require adversaries to only submit sampling algorithms that are inverse-samplable and allow them to compute a plausible guess for the identity of drawn tags in polynomial-time. We refer to such adversaries as simulatable adversaries.

**Definition 9.7 (Simulatable Adversary)**
*Let $\mathcal{A}$ be an adversary interacting with an RFID system. Let $\mathsf{view}^t_{\mathcal{A}}$ be the view of $\mathcal{A}$ at its $t$-th step and let $\mathcal{T}^t$ denote the table $\mathcal{T}$ of the DrawTag oracle at step $t$ of $\mathcal{A}$. We say that the adversary $\mathcal{A}$ is simulatable if all her sampling algorithms submitted to DrawTag are inverse-samplable and, for all $t$, there exists a polynomial-time algorithm $\mathcal{A}'$, such that $(view^t_{\mathcal{A}}, \mathcal{T}^t)$ and $(view^t_{\mathcal{A}}, \mathcal{A}'(view^t_{\mathcal{A}}))$ are indistinguishable.*

We note that when the adversary only draws one tag at the time, or, in general, a vector of size logarithmic in the security parameter, then our restrictions do not affect the original definition as any sampling algorithm over such a set is inverse-samplable. The difference may arise when the size of the returned vector is polynomial, making the probability space of exponential size. However, it is not clear whether allowing the adversary to specify one-way sampling algorithms make any practical sense.

To illustrate what can a non-simulatable adversary be, assume we have an RFID system composed of $n$ tags with identifiers $\mathsf{ID}_{b,c,i}$, where $i = 1, \ldots, n$ and $b$ is a bit set to 1 when the tag is legitimate and to 0 otherwise. We further assume an adversary who issues DrawTag queries with a sampling algorithm that runs as follows. On input a random tape $\rho$, this algorithm uses an arbitrary function $g$ and a one-way function $f$ to compute $(c_1, \ldots, c_n) = g(\rho)$ and $(b_1, \ldots, b_n) = f(c_1, \ldots, c_n)$. It then draws the $n$ tags with identifier $\mathsf{ID}_{b_i,c_i,i}$, for all $i$. As the view of the adversary only includes $b_1, \ldots, b_n$, it is hard, due to the one-wayness of $f$, to find a consistent $c_1, \ldots, c_n$.

### 9.6.2 *Privacy*

The intuition behind the privacy definition in the Vaudenay model is that any significant adversary against privacy should output a statement *deduced from the interactions between the tags and the system*. Unfortunately, the definition of blinders given by Vaudenay, Definition 8.10, fails to capture any information the adversary may get from other sources and use it to produce its statement. This intrinsic limitation comes from the fact that the blinder, as a separate entity, might not have access to all the adversary's knowledge. Hence, it may be possible for the latter to use that extra information as an advantage against the blinder. The possibility of such senario caused Vaudenay's impossibility result concerning Strong privacy that we detailled in Section 8.3.2.

In our definition hereafter, we correct this limitation by making the blinder being executed by the adversary so that it is aware of any extra information she has in her possession. We formalize this statement by giving the random tape of the adversary to the blinder. For reasons that will be made clearer later, we also restrict the privacy game to simulatable adversaries.

**Definition 9.8 (Blinder)**
*We define a blinder $B$ for an adversary $\mathcal{A}$ as a polynomial-time algorithm which sees the same view as $\mathcal{A}$ (i.e, all the incoming messages and the random tape), records all the adversary's Oracle queries and simulates all the Launch, SendReader, SendTag, Result oracles to $\mathcal{A}$. The blinder does not have access to the reader's tape so does not know the secret key nor the database. A blinded adversary $\mathcal{A}^B$ is an adversary who does not produce any Launch, SendReader, SendTag, Result oracles query but has them simulated by $B$.*

**Definition 9.9 (Privacy and Trivial Adversaries)**
*Consider a two-stage simulatable adversary who starts with an attack phase consisting of* only *oracle queries and some computations then pursuing an analysis phase with no oracle query. In between phases, the adversary receives the hidden table $\mathcal{T}$ of the DrawTag oracle then outputs* true *or* false*. The adversary wins if the output is* true.

*An adversary is said to be trivial if there exists a blinder $B$ for which* $|\Pr[\mathcal{A} \to 1] - \Pr[\mathcal{A}^B \to 1]|$ *is negligible.*

*We say that the RFID scheme is $P$ private if all the adversaries from the class $P$ are trivial.*

Clearly, combining Definitions 9.8 and 9.9, yields a (slightly) weaker privacy notion than the original one by Vaudenay. Since the adversary is not able to hide information from the blinder anymore, its only advantage in winning the privacy game must come from the protocol messages. For this reason, we argue that our proposed definition captures the exact notion of privacy. It is worth mentioning that under this new definition, the proof of the impossibility of Strong privacy does not hold as the blinder in this case "knows" if the adversary is simulating a forged tag or a legitimate one and can consequently predict the outcome of the protocol instance.

Note that all schemes that were shown to achieve a certain level of privacy in the sense of Vaudenay achieves the same level of privacy following our definition. This is because blinders that comply to Definition 8.10 can be seen as a special case of the ones considered in this chapter. In particular, all the results that we presented in Chapter 8 are still valid.

We further note that with these definitions the counter-example for the impossibility of Narrow-Strong privacy and security in the case of mutual authentication given by Armknecht et al. [ASS$^+$10] does not hold anymore. We come back to discuss this result and its implications on mutual authentication protocols in Chapter 10.

## 9.7 IND-CCA2 is not Sufficient for Strong Privacy

Consider the scheme of Figure 9.1 instantiated with an IND-CCA2 public-key encryption scheme that we construct as follows. Starting from an arbitrary IND-CCA2 secure encryption scheme ($\mathsf{KeyGen}^0$, $\mathsf{Enc}^0$, $\mathsf{Dec}^0$), we define another cryptosystem ($\mathsf{KeyGen}$, $\mathsf{Enc}$, $\mathsf{Dec}$) as follows.

- **KeyGen.** Run $(sk^0, pk^0) \leftarrow \mathsf{KeyGen}^0(1^k)$. Pick an RSA modulus $N = pq$, i.e, s.t. $p$ and $q$ are primes, and $y, z \in \mathbf{Z}_N^\star$ such that $\left(\frac{y}{N}\right) = +1$, $\left(\frac{y}{p}\right) = -1$, and $\left(\frac{z}{N}\right) = +1$. The scheme's key pair is $pk = (pk^0, N, y, z)$ and $sk = (sk^0, p)$.
- **Encrypt.** Define $\mathsf{E}_{pk}(b) = y^b r^2 \mod N$ where $b \in \{0,1\}$ and $r \in_R \mathbf{Z}_N^\star$. Pick randomness $\rho$ and compute the ciphertext

  $$\mathsf{Enc}_{pk}(x) = \mathsf{Enc}^0_{pk^0}\left(\mathsf{E}_{pk}(x_0), \ldots, \mathsf{E}_{pk}(x_{n-1})\right)$$

  where $x_0, \ldots, x_{n-1}$ is the binary decomposition of $x$.
- **Decrypt.** Define $\mathsf{D}_{sk}(c) = b$ such that $(-1)^b = \left(\frac{c}{p}\right)$. To decrypt, compute

  $$\mathsf{Dec}_{sk}(c) = \mathsf{D}_{sk}(t_0), \ldots, \mathsf{D}_{sk}(t_{n-1}),$$

where $t_0, \ldots, t_{n-1} = \mathsf{Dec}^0_{sk^0}(c)$.

We can easily see that $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is still IND-CCA2 secure and that, regardless of the properties of the initial scheme, it is not plaintext-aware since, given an integer $z \in \mathbf{Z}^\star_N$, the ciphertext $\mathsf{Enc}^0_{pk^0}\left(z \cdot \mathsf{E}_{pk}(x_0) \bmod N, \ldots, z \cdot \mathsf{E}_{pk}(x_{n-1}) \bmod N\right)$ is, depending on $\left(\frac{z}{p}\right)$, a valid encryption of either $x_0, \ldots, x_{n-1}$ or $\overline{x_0}, \ldots, \overline{x_{n-1}}$. Therefore, the existence of a knowledge extractor induces the existence of a polynomial-time algorithm for distinguishing quadratic residues from non-quadratic residues.

Finally, the following Strong adversary defeats privacy.

```
CreateTag($\mathsf{ID}$)
$\mathsf{vtag} \leftarrow$ DrawTag($\mathsf{ID}$)
$\mathsf{ID}\|K_{\mathsf{ID}} \leftarrow$ Corrupt($\mathsf{vtag}$)
$\pi \leftarrow$ Launch
$a \leftarrow$ SendReader($\emptyset, \pi$)
Set $x = \mathsf{ID}\|K_{\mathsf{ID}}\|a$
$c \leftarrow \mathsf{Enc}^0_{pk}\left(z \cdot \mathsf{E}(x_0), \ldots, z \cdot \mathsf{E}(x_{n-1})\right)$
SendReader($c, \pi$)
$b \leftarrow$ Result($\pi$)
Output $b$
```

Clearly, an adversary outputs $1$ if and only if $\left(\frac{z}{p}\right) = +1$. Therefore, a blinder that follows the same distribution would break the quadratic residuosity problem, i.e., the problem of distinguishing quadratic residues from non-quadratic residues.

## 9.8 Strong Privacy Using Plaintext-Awareness

In this section, we show that using the new definition of blinders, we can achieve Strong privacy using public-key cryptography. For this sake, we make use of the standard definitions of public-key cryptosystems (PKC) and the notion of plaintext-aware encryption schemes.

We consider the same protocol based on a public-key cryptosystem, as depicted in Figure 9.1. In this scheme, the state of the tags is composed of their $\mathsf{ID}$ and a uniformly distributed $\kappa$-bit string $K_{\mathsf{ID}}$. Upon reception of an $\alpha$-bit string challenge $a$, a tag sends the encryption of $\mathsf{ID}\|K_{\mathsf{ID}}\|a$ under the public key $pk$ to the reader. The latter decrypts the received ciphertext using its secret key $sk$ and checks that it is well formed, that $a$ is correctly recovered and that $(\mathsf{ID}, K)$ exists in the database. Note that $\kappa$ and $\alpha$ have to be polynomially bounded.

Although this challenge-response protocol has already been used by Vaudenay [Vau07] to achieve Narrow-Strong privacy under the assumption that the underlying encryption scheme is IND-CPA secure, our result requires PA1+ plaintext-awareness from the encryption scheme. Naturally, since our definition of security is unchanged from the original model, IND-CCA2 security for the encryption scheme is sufficient to prove that the protocol is secure and we use the original result of Vaudenay.

The next theorem establishes the correctness, security, and Strong privacy of the scheme.

| **Tag** | | **System** |
|---|---|---|
| **State:** $pk$, $\mathsf{ID}$, $K_{\mathsf{ID}}$ | | **Secret key:** $sk$ |
| | | **Database:** $\{\dots, (\mathsf{ID}, K_{\mathsf{ID}}), \dots\}$ |
| | $\xleftarrow{\quad a \quad}$ | Choose $a \in_R \{0,1\}^{\alpha}$ |
| $c = \mathsf{Enc}_{pk}(\mathsf{ID} \Vert\, K_{\mathsf{ID}} \Vert a)$ | $\xrightarrow{\quad c \quad}$ | Check that $\exists(\mathsf{ID}, K_{\mathsf{ID}}) \in \mathsf{DB} : \mathsf{Dec}_{sk}(c) = \mathsf{ID} \Vert K_{\mathsf{ID}} \Vert a$ |
| | | **Output:** $\mathsf{ID}$ or $\perp$ if not found |

**Figure 9.1:** A correct, secure, and Strong-private RFID authentication protocol based on an IND-CPA and PA1+ plaintext-aware public-key encryption scheme.

**Theorem 9.4**
*Assume that the public-key encryption scheme used in the RFID scheme of Fig. 9.1 is correct, PA1+ plaintext-aware, and IND-CCA2 secure. If $2^{-\kappa}$ and $2^{-\alpha}$ are negligible, then the scheme is correct, secure, and Strong private.*

## 9.9 Security Proof

### 9.9.1 *Correctness*

Correctness is trivially shown using the encryption scheme's correctness and the fact that the scheme is stateless.

### 9.9.2 *Security*

Since the scheme complies to Definition 8.2 for simple RFID protocols, we can apply simple security using an adversary who creates a single tag $\mathsf{ID}$ and makes $\Psi(sk, a', c', \mathsf{ID}', K_{\mathsf{ID}'})$ queries (with the restriction $\mathsf{ID}' \neq \mathsf{ID}$). These queries consist in checking whether $\mathsf{Dec}_{sk}(c') = \mathsf{ID}' \Vert K_{\mathsf{ID}'} \Vert a'$. We also let $a$ be the output of SENDREADER$(\cdot, \pi)$.

We construct an algorithm $\mathcal{B}$ who receives $pk$ and simulates the RFID system, $\mathcal{A}$, and all oracle queries without $sk$. We further make $\mathcal{B}$ output $\mathcal{A}$'s output. To handle $\mathcal{A}$'s $\Psi$ queries, $\mathcal{B}$ is given an access to a decryption oracle that she queries with $c'$ and later checks whether the plaintext is equal to $\mathsf{ID}' \Vert K_{\mathsf{ID}'} \Vert a'$. We now use a sequence of games in which $S_i$ denotes the event that the adversary wins Game $i$.

*Game 0.* We let this game be the security experiment played by a fixed $\mathcal{A}$ that has her environment simulated by $\mathcal{B}$. Let $S_0$ be the event that the adversary wins the security experiment.

*Game 1.* We first make a change in Game 0 and define Game 1 as being the same except that SENDREADER$(-, \pi)$ never produces an $a$ that was sent before to the SENDTAG interface. In other words, we require that $\mathcal{A}$ never guesses $a$. As $a$ is chosen uniformly, when $\mathcal{A}$ makes $s$ calls to SENDTAG, the probability of this event happening is bounded by $s2^{-\alpha}$ so that

$$|\Pr[S_0] - \Pr[S_1]| \leq s2^{-\alpha}.$$

Since $s$ is polynomially bounded, this probability is negligible when $2^{-\alpha}$ is negligible.

*Game 2.* We define Game 2 in which $\mathcal{A}$ never issues a SENDREADER$(c, \pi)$ query for a $c$ that was obtained from a SENDTAG$(a^\star, \mathsf{vtag})$ query. Since SENDTAG never received $a$, the decryption of the ciphertext the adversary submits must fail to match the $a$ chosen by the reader for session $\pi$ and the RESULT interface answers with 0. Therefore, the adversary always loses the game when this event occurs and we have

$$\Pr[S_1] - \Pr[S_2] = 0$$

*Game 3.* In Game 2 we modify the SENDTAG$(a, \mathsf{vtag})$ interface so that instead of encrypting of $\mathsf{ID}\|K_{\mathsf{ID}}\|a$, it encrypts a random $R$ of the same length. (Recall that no such output is sent to the reader.)

We now construct a hybrid argument to show that $|\Pr[S_3] - \Pr[S_2]|$ is negligible. For that, we first let $\mathcal{B}_3(i)$ be an hybrid blinder that acts as follows: The $i$ first SENDTAG$(a_i, \mathsf{vtag})$ queries submitted by the adversary are treated by encrypting $\mathsf{ID}\|K_{\mathsf{ID}}\|a_i$ and the rest by encrypting random strings. We let $\mathcal{C}$ denote an IND-CCA2 adversary who plays the IND-CCA2 game, simulating $\mathcal{B}_2(i)/\mathcal{B}_2(i+1)$ by submitting $\mathsf{ID}\|K_{\mathsf{ID}}\|a$ (as in $\mathcal{B}_2(i+1)$) and $R$ (as in $\mathcal{B}_2(i)$) to the IND-CCA2 challenger who randomly chooses one of the messages and returns its encryption. $\mathcal{C}$ then continues $\mathcal{B}_2(i)/\mathcal{B}_2(i+1)$'s execution and returns its output. The difference in the output of $\mathcal{B}_2(i)$ and $\mathcal{B}_2(i+1)$ can thus be expressed as a distinguisher advantage for the IND-CCA2 game which is negligible by assumption. Therefore, we find that

$$|\Pr[S_3] - \Pr[S_2]| = \mathsf{negl}(k).$$

At this point, $\mathcal{A}$ is receiving messages that are not related to $K_{\mathsf{ID}}$. Clearly, the only way for her to win the game is to guess $K_{\mathsf{ID}}$ which happens with probability $2^{-\kappa}$. In other words,

$$\Pr[S_3] = 2^{-\kappa}$$

Therefore, the scheme is secure. □

### 9.9.3 *Privacy*

To conduct the proof, we consider a Strong adversary $\mathcal{A}$ and construct a blinder iteratively. That is, we construct a sequence of partial blinders $B_1, \ldots, B_5$ and let $\mathcal{A}_i = \mathcal{A}_{i-1}^{B_i}$ with $\mathcal{A}_0 = \mathcal{A}$. The final blinder for $\mathcal{A}$ is $B = B_1 \circ \cdots \circ B_5$. By showing that the outcome of $\mathcal{A}_i$ and $\mathcal{A}_{i+1}$ are computationally indistinguishable, we deduce that $B$ is indeed a full blinder for $\mathcal{A}$. So, the scheme is Strong private.

We will denote $E$ the event that an adversary $\mathcal{A}$ wins the security experiment, i.e., manages to make the reader accept an uncorrupted tag without matching conversation. We have that $\Pr[E] = \mathsf{negl}(k)$.

*Game 0.* We first fix an adversary $\mathcal{A}_0$ playing the privacy game.

*Game 1.* We let Game 1 denote the privacy game performed by an adversary who simulates every Result on a session $\pi$ with a transcript $(a, c)$, such that $c$ that has been obtained by a previous SendTag$(\mathsf{vtag}, a')$ query.

If $a \neq a'$, we are ensured that $c$ does not decrypt to something containing $a$, so the answer to Result$(\pi)$ must be 0. The simulation is easy and perfect. In the other case, that is, if $a = a'$, the decryption of $c$ will be parsed to a matching challenge $a$ and some entry $\mathsf{ID}\|K_{\mathsf{ID}}$ which is in the database if and only if $\mathsf{vtag}$ is legitimate. Fortunately, the blinder has access to this latter information as it is returned in the response of the DrawTag oracle query drawing $\mathsf{vtag}$. Again, the simulation is easy and perfect. This fully defines $B_1$ and we deduce that

$$\Pr[\mathcal{A}_0 \to 1] = \Pr[\mathcal{A}_0^{B_1} \to 1]$$

We can thus define the adversary $\mathcal{A}_1$ that never queries Result on an instance $\pi$ in which the response $c$ was produced by a previous SendTag query.

*Game 2.* In this game, we make all SendTag queries being simulated by a partial blinder $B_2$. To achieve this, we let $r$ be number of SendTag queries and make a sequence of hybrid blinders $B_2^1, \ldots, B_2^{r+1}$ in which $B_2^i$ simulates the $i - 1$ first SendTag queries. Note that $B_2^1$ does not make any simulation so $\mathcal{A}_1^{B_2^1}$ is exactly $\mathcal{A}_1$. Conversely, $\mathcal{A}^{B_2^{r+1}}$ has all its SendTag queries simulated. We can thus set $B_2 = B_2^{r+1}$. The hybrid $B_2^{i+1}$ simulates the $i$ first encountered SendTag queries by encrypting random strings of same length as $\mathsf{ID}\|K_{\mathsf{ID}}\|a$.

To prove that $\mathcal{A}_1^{B_2^i}$ and $\mathcal{A}_1^{B_2^{i+1}}$ have computationally indistinguishable distributions, we construct an adversary $\mathcal{C}$ playing the IND-CCA2 game. Adversary $\mathcal{C}$ receives the public key and runs $\mathcal{A}_1^{B_2^i}$ or $\mathcal{A}_1^{B_2^{i+1}}$, depending on the bit of the indistinguishability game, while simulating the RFID system, except the $i$-th SendTag query. For that, $\mathcal{C}$ must simulate the environment for $\mathcal{A}_1^{B_2^i}/\mathcal{A}_1^{B_2^{i+1}}$. Since all algorithms and oracles of the scheme, except for Result, do not require the secret key, $\mathcal{C}$ can easily perform the simulation by itself. Regarding the RESULT oracle, $\mathcal{C}$ just queries a decryption oracle and checks whether the decrypted message matches.

The first $i-1$ SENDTAG queries are made to the IND-CCA2 challenger in a real-or-random version. The challenge ciphertext $c$ in the IND-CCA2 game is the answer from the challenger. It is either a real answer (as in the $\mathcal{A}_1^{B_2^i}$ simulation) or a simulated one (as in the $\mathcal{A}_1^{B_2^{i+1}}$ simulation). Note that no RESULT query is made on the session in which the adversary sent $c$ (this case has been taken care of in Game 1). So, $\mathcal{C}$ prefectly simulates either the game for $\mathcal{A}_1^{B_2^i}$ or the game for $\mathcal{A}_1^{B_2^{i+1}}$ and is an IND-CCA2 adversary. Since $\mathcal{C}$ produces the output of $\mathcal{A}_1^{B_2^i}/\mathcal{A}_1^{B_2^{i+1}}$, we obtain that

$$|\Pr[\mathcal{A}_1^{B_2^i} \to 1] - \Pr[\mathcal{A}_1^{B_2^{i+1}} \to 1]| \leq \mathsf{Adv}^{\mathsf{IND-CCA2}}(k),$$

and it results that

$$|\Pr[\mathcal{A}_1 \to 1] - \Pr[\mathcal{A}_1^{B_2} \to 1]| \leq r \cdot \mathsf{Adv}^{\mathsf{IND-CCA2}}(k),$$

which is negligible as $r$ is polynomially bounded and the scheme is IND-CCA2 secure.

At this point, we can legitimately consider an adversary $\mathcal{A}_2$ who makes no SENDTAG queries.

*Game 3.* We now simulate all remaining RESULT queries. To do so, we construct an adversary $\mathcal{E}$ playing the PA1++ game.

This adversary takes the public key then simulates $\mathcal{A}_2$ interacting with the RFID system. Recall that, like in Game 2, the algorithms and oracles of the scheme do not depend on the secret key, except for the RESULT queries that will be treated hereafter. We let $\mathcal{E}$ simulate the RFID system to $\mathcal{A}_2$, handling her queries as follow:

- Assuming w.l.o.g. that session identifiers are not randomized, LAUNCH is deterministically computed by $\mathcal{E}$.
- Upon a CREATETAG($\mathsf{ID}$) query from $\mathcal{A}_2$, $\mathcal{E}$ inserts $(\mathsf{ID}, -)$ in a table $\mathsf{DB}_1$ if the query asks for a legitimate tag. Otherwise, it inserts $(\mathsf{ID}, -)$ in a table $\mathsf{DB}_0$.
- $\mathcal{E}$ simulates SENDREADER queries by asking the oracle $\mathcal{O}_S$ to sample from the uniform distribution over $\{0,1\}^\alpha$. It then forwards the received answer to $\mathcal{A}_2$.
- DRAWTAG($\mathsf{Samp}$) queries are handled by asking the randomness oracle $\mathcal{O}_S$ to sample from the distribution specified by $\mathsf{Samp}$ to get one or more random $\mathsf{ID}$. If any of the returned identifiers corresponds to a drawn tag, $\mathcal{E}$ outputs $\perp$. Otherwise, it generates, deterministically and for each returned $\mathsf{ID}_i$, a fresh $\mathsf{vtag}_i$ and inserts the pair $(\mathsf{vtag}_i, \mathsf{ID}_i)$ into the table $\mathcal{T}$. After that, and for each $\mathsf{ID}_i$, it sets the bit $b_i$ to $1$ if $\mathsf{ID}_i$ is legitimate, and to $0$ otherwise. At last, it returns $((\mathsf{vtag}_1, b_1), \ldots, (\mathsf{vtag}_n, b_n))$ to $\mathcal{A}_2$.

- Corrupt$(\mathsf{vtag})$ makes $\mathcal{E}$ reveal $\mathsf{ID} = \mathcal{T}(\mathsf{vtag})$. Moreover, $\mathcal{E}$ looks for the entry $(\mathsf{ID}, K_{\mathsf{ID}})$ in $\mathsf{DB}_0$ and $\mathsf{DB}_1$. If that corresponding entry contains a $K_{\mathsf{ID}}$ different from $'-'$, then it returns it. Otherwise, it queries $\mathcal{O}_S$ to sample from the uniform distribution over $\{0,1\}^\kappa$ and assigns the answer to $K_{\mathsf{ID}}$. It subsequently updates the entry $(\mathsf{ID}, -)$ to $(\mathsf{ID}, K_{\mathsf{ID}})$ and returns this last pair as its answer. We further assume that whenever the tag $\mathsf{ID}$ is a legitimate one, $\mathcal{E}$ inserts the entry $(\mathsf{ID}, K_{\mathcal{T}(\mathsf{vtag})})$ in a table $\mathcal{T}_{\mathcal{E}}$.
- To simulate the Result$(\pi)$ oracle for an instance $\pi$ with transcript $(a, c)$, $\mathcal{E}$ sends $c$ to the decryption oracle, checks that the recovered plaintext is of the form $\mathsf{ID}\|K_{\mathsf{ID}}\|a$, that it matches a tag state $\mathsf{ID}\|K_{\mathsf{ID}}$ obtained from a previous Corrupt query on a legitimate tag. If this is the case, the answer to Result must be 1, otherwise, the simulated answer is 0. Note that when the output of the $\mathcal{E}$ regarding a Result query is 1, the genuine Result query would also have answered 1. This is because that, knowing a subset of the database through corruption query, $\mathcal{E}$ can effectively predict the answer when the database entry lies in this subset. Errors in the simulation thus occur when $\mathcal{E}$ predicts $0$ and the genuine Result query would also have outputted 1 in a session without matching conversation. Clearly, the failure of one of $\mathcal{E}$'s simulations corresponds to the happening of the event that $\mathcal{A}_2$ wins the security game, that we denoted $E$ in the previous section. In other words,

$$\left|\Pr\left[\mathcal{A}_2 \to 1\right] - \Pr\left[\mathcal{A}_2^{\mathcal{E}} \to 1\right]\right| \leq \Pr[E]$$

Since we assumed the encryption scheme to be PA1++ plaintext-aware, we can use the plaintext extractor $\mathcal{E}^\star$ of $\mathcal{E}$ to replace the decryption oracle without significantly altering the outcome distribution. However, $\mathcal{E}^\star$ requires the view of $\mathcal{E}$ instead of the view of $\mathcal{A}_2$, so we cannot use it as an extractor for $\mathcal{A}_2$. Fortunately, it is possible to reconstruct that view given the adversary's random tape and its queries. At first, we note that $\mathcal{E}$'s random coins are only used to initialize $\mathcal{A}_2$. Furthermore, all the randomness $\mathcal{E}$ obtains from $\mathcal{O}_S$ to process Corrupt queries is revealed to $\mathcal{A}_2$. Moreover, since $\mathcal{A}_2$ is simulatable, we can use the algorithm $\mathcal{A}_2'$, induced by Definition 9.7, to reconstruct, from $\mathcal{A}_2$'s view, a table $\mathcal{T}'$ indistinguishable from $\mathcal{T}$. Since this table lists all the mappings between real and virtual identifiers, it is straightforward to reconstruct a randomness for $\mathcal{E}$ that she received to process the DrawTag queries using the $\mathsf{Samp}^{-1}$ algorithms corresponding to the sampling queries of $\mathcal{A}_2$. We let this whole operation be carried by a polynomial-time algorithm $\mathcal{V}$ that takes as input the view of $\mathcal{A}_2$ and uses $\mathcal{A}_2'$ to reconstruct a view of $\mathcal{E}$, i.e., it is such that $\mathcal{V}(\mathsf{view}_{\mathcal{A}_2})$ and $\mathsf{view}_{\mathcal{E}}$ are indistinguishable. It follows that $\mathcal{E}^\star(pk, \cdot, \mathcal{V}(\mathsf{view}_{\mathcal{A}_2}))$ and $\mathcal{E}^\star(pk, \cdot, \mathsf{view}_{\mathcal{E}})$ are indistinguishable.

At this point, we are able to define $B_3$, the partial blinder for Result queries. Similar to $\mathcal{E}$, we assume that $B_3$ maintains a table $\mathcal{T}_{B_3}$ containing a list of pairs $(\mathsf{ID}, K_{\mathsf{ID}})$

for corrupted legitimate tags. In order to simulate a RESULT query on an instance $\pi$ of transcript $(a, c)$, the blinder proceeds as follow.

a) The blinder calls upon $\mathcal{E}^{\star}(pk, c, \mathcal{V}(\mathsf{view}_{\mathcal{A}_2}))$ to get $\mathsf{Dec}_{sk}(c) = \mathsf{ID}\|K_{\mathsf{ID}}\|a'$.

b) After that, it verifies that $a = a'$ and outputs $0$ in case of failure. Otherwise, it continues.

c) Then, it outputs $1$ if the pair $\mathsf{ID}\|K_{\mathsf{ID}}$ is listed in $\mathcal{T}_{B_3}$. Otherwise, it outputs $0$.

The probability that Step 1 fails can be expressed as a distinguisher advantage of the PA1++ game or between $\mathcal{V}(\mathsf{view}_{\mathcal{A}_2})$ and $\mathsf{view}_{\mathcal{E}}$, so

$$\left|\Pr\left[\mathcal{A}_2^{B_3} \to 1\right] - \Pr\left[\mathcal{A}_2^{\mathcal{E}} \to 1\right]\right| \le \mathsf{Adv}^{\mathsf{PA1{+}{+}}}(k) + \mathsf{negl}(k).$$

At the same time, Step 3 fails when the event $E$ occurs, so using triangle inequalities we conclude that

$$\left|\Pr\left[\mathcal{A}_2 \to 1\right] - \Pr\left[\mathcal{A}_2^{B_3} \to 1\right]\right| \le \mathsf{Adv}^{\mathsf{PA1{+}{+}}}(k) + \Pr[E] + \mathsf{negl}(k).$$

Recalling that $E$ occurs with negligible probability and that the scheme is PA1++ plaintext-aware, the quantity above becomes negligible. Hence, $B_3$ describes a successful blinder for the RESULT oracle.

*Game 4.* In this game, we get rid of SENDREADER queries. This can easily be achieved by constructing a blinder $B_4$ that returns uniformly distributed values from the set $\{0, 1\}^{\alpha}$. Clearly, simulation is perfect as both distributions are perfectly indistinguishable. Hence,

$$\Pr\left[\mathcal{A}_3 \to 1\right] = \Pr[\mathcal{A}_3^{B_4} \to 1]$$

*Game 5.* Finally, we have an adversary $\mathcal{A}_4$ who only produces LAUNCH queries. Since these are generated deterministically, we can trivially simulate the LAUNCH oracle using a blinder $B_5$. It follows that

$$\Pr\left[\mathcal{A}_4 \to 1\right] = \Pr[\mathcal{A}_4^{B_5} \to 1]$$

In the end, we have obtained an adversary $\mathcal{A}_5 = \mathcal{A}^B$, with $B = B_1 \circ \cdots \circ B_5$, who does not produce any oracle query that is such that

$$\left|\Pr[\mathcal{A} \to 1] - \Pr[\mathcal{A}^B \to 1]\right| = \mathsf{negl}(k)$$

The scheme is thus Strong private. □

## 9.10 Perspective and Future Development

To overcome the limitations put on sampling queries, we may use a stronger notion of plaintext-awareness by adding auxiliary inputs to the adversary in the plaintext-awareness definitions. If an encryption scheme could ever be proved to be auxiliary-PA1+ , then we would be able to achieve Strong privacy against adversaries who have no restriction on the choice of the efficient sampling algorithms to use.

The difficulty in achieving this notion lies in how to simulate the avdersary's behavior if it depends on the system's randomness. In such cases, the extractor would need to be able to execute the adversasary "in the same way" it is executed in real conditions. Unfortunately, giving to the extractor the history of queries of the adversary cannot help

# 10

# THE CASE OF MUTUAL AUTHENTICATION

Contents

After studying unilateral authentication protocols for RFIDs, we turn to the case of mutual authentication where, in addition to the tag proving its identity to the reader, the latter is also required to authenticate itself. This feature can be useful is many situations, especially if the tag embeds a detection mechanism allowing it to trigger an alarm when its number of unsuccessful authentications exceeds a certain threshold.

Our starting point is the paper of Paise and Vaudenay [PV08] which extended Vaudenay's model to mutual authentication. Unfortunately, Armknecht, Sadeghi, Scafuro, Visconti, and Wachsmann showed that some of their results are not sound [ASS$^+$10]. So, in order to overcome the limitations induced by Paise and Vaudenay's definitions, we consider a less restrictive, but meaningful, notion of security for the readers for which the analysis of Armknecht et al. does not hold. Concretely, we restrict the security game to adversaries whose goal is to make an uncorrupted tag accept the reader without any matching conversation. Previously, Paise and Vaudenay considered this notion for all tags, corrupted and uncorrupted ones. We show that in these settings the results of the PV model concerning Weak privacy using a PRF and Narrow-Destructive privacy in the random oracle model are valid. However, we show that regardless of which security notion we take, the protocol based on an IND-CPA public-key encryption scheme fails to be Narrow-Strong private.

We extend our definitions with blinders having access to the adversary's random tape to mutual authentication and show that secure Strong privacy with mutual authentication is achievable. As it was the case for simple tag authentication, we will use plaintext-aware encryption schemes to achieve our goal.

## 10.1 Enriching The Definitions

We first adapt the definitions of Chapters 8 and 9 to cover the case of mutual authentication. We leave the extension of the definition of security for an RFD system to Section 10.2 and relate it to the results of Armknecht et al.

### 10.1.1 *RFID System with Mutual Authentication*

The definition of an RFID system with mutual authentication is very similar to Definition 8.1 with the difference that we introduce an output for the tag that is one bit equal to $1$ when the latter authenticates the reader and $0$ otherwise.

**Definition 10.1 (RFID System with Mutual Authentication)**
*An RFID system with mutual authentication is composed of the following three algorithms*

1. $\mathsf{SetupReader}(1^k) \rightarrow (sk, pk)$. *This first probabilistic polynomial-time algorithm is used to initialize the reader. As such, it takes as input a security parameter $k$ and outputs a*

*pair of secret/public key $(sk, pk)$ for the reader (if no public-key cryptography is used in the RFID scheme then $pk$ is set to $\bot$).*

2. $\mathsf{SetupTag}_{pk}(\mathsf{ID}) \rightarrow (K_{\mathsf{ID}}, S_{\mathsf{ID}})$. *This probabilistic polynomial-time algorithm creates a tag with unique identifier* ID. *The state $S_{\mathsf{ID}}$ is stored inside the tag while an entry $(\mathsf{ID}, K_{\mathsf{ID}})$ is inserted in the server's database* DB *when the created tag is legitimate.*

3. *A two-party game run between a reader and a tag* ID*, each one of them following an interactive polynomial-time probabilistic algorithm, come to complete the definition. Apart from the random bits, the algorithm for the tag takes as input the state $S_{\mathsf{ID}}$ while the algorithm for the reader takes as input the database* DB*, and the reader's secret key $sk$. In the end, the reader ends up with a tape* Output*, set to $\bot$ if the instance failed from its perspective, and the tag* ID *gets a tape* $\mathsf{Output}_{\mathsf{ID}}$ *that is set as follows:*

   - *After* ID *completes a session in which reader authentication succeeds, it sets* $\mathsf{Output}_{\mathsf{ID}} = 1$.
   - *If reader authentication failed, then* $\mathsf{Output}_{\mathsf{ID}} = 0$.
   - *When the tag receives a message for a new protocol session message , it sets* $\mathsf{Output}_{\mathsf{ID}} = 0$.

   *A protocol execution with* ID *is called succeeded if it has a matching conversation with the reader with output* ID. *For the ease of notation, we shall denote by* $\mathsf{Output}(\pi)$ *the output of the protocol instance with identifier $\pi$ and we write* $\mathsf{Output}_{\mathsf{ID}}(\tau)$ *to refer to the output of the tag after having completed a protocol instance with transcript $\tau$.*

For mutual authentication protocols, we also define the class of simple protocols that comply to Definition 8.2. Adversaries are defined like in the case of unilateral authentication, i.e., following Definition 8.3. We also retain Definition 8.4 for the classification of adversaries in Weak, Forward, Destructive, and Strong classes and their Narrow counterparts.

### 10.1.2 *Correctness*

Correctness in its two variants also needs to be adapted by requiring the tag to authenticate the reader every time both have a matching conversation, except with negligible probability. That is, we have the following two definitions.

**Definition 10.2 (Correctness of an RFID Scheme with Mutual Authentication)**
*We say that an RFID scheme is correct if an undisturbed protocol instance run between the reader and a legitimate tag* ID *selected according to any distribution, even after a Strong adversary interacted with the system, results in the reader outputting* ID *and the tag outputting* $1$*, except with negligible probability. When* ID *is illegitimate, the reader's and tag's outputs regarding the instance has to be $\bot$ and $0$ respectively, except with negligible probability.*

**Definition 10.3 (Weak Correctness of Simple RFID Schemes with Mutual Authentication)** *We say that a simple RFID system with mutual authentication is weakly-correct if there exists an efficiently computable predicate $\Psi'(\mathsf{ID}, t)$ such that t is the number previously completed successive sessions after the last one in which* ID *accepted the reader. The predicate is such that if it yields* $1$*, then a complete undisturbed run instance between* ID *and the reader results in both parties accepting, i.e., the reader outputting* ID *and the tag* $1$*, except with negligible probability. If the predicate returns* $0$*, then the instance's output is* ID *for the reader and* $0$ *for the tag, except with negligible probability.*

### 10.1.3 *Privacy*

Concerning privacy, we use the definitions we proposed in Chapter 9, i.e., Definitions 9.8 and 9.9.

## 10.2 Defining Security for the Tags

On one hand, privacy, as a measure of information leakage from protocol messages, is unaffected by whether the protocol consists of tag-to-reader or mutual authentication. On the other hand, extending the definitions of correctness and weak correctness is rather straightforward.

Thus, the only property that remains to be adapted is security. Unilateral authentication protocols have a security notion consisting in the inability of any Strong polynomial-time adversary of making the reader accept a tag that did not have matching conversation. Security for mutual authentication protocols would basically need to duplicate the requirements for both sides. However, we remark that almost all protocols enabling mutual authentication are such that the reader authenticates itself by proving that it has retrieved the partner tag's secret and ID (Hence, tag authentication happens first). Consequently, the question of whether to restrict security on the tag side to only uncorrupted tags is irrelevant for this class of authentication protocols.

For this reason, Paise and Vaudenay used the unrestrictive version of the definition of security. However, Armknecht et al. showed that using this definition leads to incompatibilities with the notion of Narrow-Strong privacy: No protocol can be both secure in the stronger sense and Narrow-Strong private. Their result therefore invalidated Paise and Vaudenay's proof of achieving Narrow-Strong privacy and security for the tags. Moreover, it appears that the proofs of security for their Narrow-Destructive and Weak private protocols is compromised under this definition. Nevertheless, Paise and Vaudenay's results are valid under the more restrictive definition. Since we will mainly use the weakest notion and use the strongest one only for the Strong private protocol of Section 10.7, we shall call secure a scheme that is secure for the reader and for the tags in the strongest case strongly secure while secure refers to a scheme that is secure for the reader and only secure for uncorrupted tags.

**Definition 10.4 (Security for the tags)**
*We say that an RFID scheme is secure for the tags if no Strong adversary can make an uncorrupted tag* ID *output* $1$ *for a session that did not have matching conversation with any of the reader's protocol sessions, except with negligible probability. The tag* ID *is called the target tag.*

**Definition 10.5 (Security of RFID Schemes with Mutual Authentication)**
*An RFID scheme is secure if it is secure for both the tags, following Definition 10.4 and for the reader, following Definition 8.8.*

In some case, security can be proven with respect to adversaries who can even corrupt the target tag. We shall refer to protocols satisfying this notion as strongly-secure.

We can also show that the definition of security simplifies when one restricts to simple and weakly-correct RFID scheme.

**Definition 10.6 (Simple Security for the Tags)**
*For simple and weakly-correct RFID schemes with mutual authentication, we consider the following simplified security game for the tags for adversaries who are given access to an oracle* $\mathcal{O}_\Psi$ *who checks the predicate* $\Psi(sk, \cdot, \cdot, \cdot)$.

```
1: $(sk, pk) \leftarrow \mathsf{SetupReader}(1^k)$
2: $\textsc{CreateTag}^1(\mathsf{ID})$
3: $\mathsf{vtag} \leftarrow \textsc{DrawTag}(\mathsf{ID})$
4: Run $\mathcal{A}^{\mathcal{O}_\Psi}$ interacting with Launch, SendReader, and SendTag.
5: Let $\mathsf{Output}_{\mathsf{ID}}$ be the current output of the tag.
6: Output $1$ if $\mathsf{Output}_{\mathsf{ID}} = 1$ and the last instance of tag ID had no matching conversation with
   the reader.
```

*The scheme is said to be simply secure for the tags if the winning probability of any adversary playing the simple security experiment is negligible in the security parameter.*

**Definition 10.7 (Security of an RFID Scheme with Mutual Authentication)**
*An RFID scheme is said to be simply secure if it is simply secure for the reader and for the tags.*

We have the following lemma which mirrors Lemma 8.1 for the case of one-way authentication.

**Lemma 10.1**
*For simple and weakly-correct RFID schemes, simple security implies security.*

**Proof.** The case of tag authentication follows from Lemma 8.1 so we only need to prove simple security for the tags. That is, we consider an adversary playing the security game for the tags and reduce it to an adversary who plays the simple security game for the tags.

*Game 0.* This denotes the original security game played by a fixed Strong adversary $\mathcal{A}$. We let $S_0$ be the event that $\mathcal{A}$ succeeds. Recall that $\mathcal{A}$ has access to all interfaces. We assume, w.l.o.g., that $\mathcal{A}$ stops as soon as it wins the security game, i.e., one tag $\mathsf{ID}$ has $\mathsf{Output}_{\mathsf{ID}} = 1$ after completing a session without matching conversation.

*Game 1.* We add a new condition for $\mathcal{A}$ to win by requiring it to correctly guess the target tag $\mathsf{ID}$ when created and the time at which $\mathsf{Output}_{\mathsf{ID}} = 1$ after a session without matching conversation. If $S_3$ is the event that the adversary wins this game, $n$ are the number of tags created, and $q$ the number of sessions $\mathsf{ID}$ completed, we have

$$\Pr[S_2] \geq \frac{1}{nq} \Pr[S_1]$$

*Game 2.* We now simulate the creation of all tags except the target one. That is, we process all CreateTag queries with a parameter different from $\mathsf{ID}$ in the following way. $\mathcal{A}$ calls on $\mathsf{SetupTag}_{pk}(\cdot)$ to generate the tag state and the key for the database. If the query concerns a legitimate tag, $\mathcal{A}$ inserts the entry into a list of legitimate tags $\mathsf{Tags}_1$. pair $(\mathsf{ID}_\star, S_\star)$ into a list $\mathsf{Tags}_0$. computed using $sk$, $\mathcal{A}_2$ can easily simulate SendTag Since $\mathcal{A}$ has knowledge of all states of the tags, she can simulate all SendTag queries related to any tag, except $\mathsf{ID}$ that is forwarded to the SendTag interface (Recall that $\mathcal{A}$ draws tags herself so she knows the real $\mathsf{ID}$ of every tag). The simulation is thus perfect.

We also need to show that $\mathsf{Output}$, and thus Result, can be simulated. To determine the outcome of a protocol session, $\mathcal{A}$ tests queries $\mathcal{O}_\Psi$ on every entry except $(\mathsf{ID}, K_{\mathsf{ID}})$ to determine which entry satisfies $\Psi$. As for $(\mathsf{ID}, K_{\mathsf{ID}})$, $\mathcal{A}$ assumes that $\Psi$ would answer $0$ if the instance does not have matching conversation with that tag. Otherwise, it assumes it to be $1$. Therefore, when the predicate tested with $(\mathsf{ID}, K_{\mathsf{ID}})$ would have yield 0, $\mathcal{A}$ perfectly simulates $\mathsf{Output}$ (The rest of the protocol messages do not depend on $K_{\mathsf{ID}}$ if $\mathsf{ID}$ has not been identified). If the predicate would have answered $1$ with $(\mathsf{ID}, K_{\mathsf{ID}})$ and without matching conversation, it should already have been the target session and this is addressed with another selection in Game 2. So, simulation is perfect and we find that

$$\Pr[S_4] = \Pr[S_3]$$

Note that the adversary submits its SendReader query if its simulated output is $\mathsf{ID}$ so that the database entry can be correctly updated.

Finally, we notice that Game 3 is described by the simple security experiment. We therefore conclude that simple security for simple and weakly-correct RFID schemes implies security. □

## 10.3 Limitations of Mutual Authentication

Several possibilities can be considered to obtain reader authentication. In short, we have three options: having both devices authenticating at the same time, having the tag authenticating before the tag and vice-versa.

Unfortunately, the first option, which is clearly the best, cannot be achieved. That is, as a fair-exchange protocol is unrealizable without the involvement of a trusted third party, we cannot design solutions in which both a tag and the reader authenticate themselves "at the same time".

We turn to the case of the reader authenticating itself before the tag does. Clearly, this approach has the benefit that tags would a priori know whether they are communicating with a legitimate entity or an attacker and limit the later. However, authenticating the reader can only be done through a primitive dedicated to that, i.e., either a MAC or a digital signature and each solution suffers from disadvantages. Regarding MACs, since the reader does not a priori know which tag it is authenticating to, it has to share a single symmetric key with all the tags that belong to the system. In these settings, it is not hard to see that such a scheme fails to be secure as any adversary who succeeds in tampering with one tag becomes able to authenticate to other tags as the reader. On their side, digital signatures raise a privacy issue. That is, a protocol in which the reader issues digital signatures can never achieve privacy as the adversary knows its public-key. Therefore, it is easy for her to distinguish the real messages from the ones produced by the blinder (as long as the signature scheme is unforgeable) by running the signature's verification algorithm.

We are only left with the second option of making the reader authenticate itself after the tag did it. The benefit from this approach is that the reader, having recovered its partner's identity and associated secret, can make use of both information to securely authenticate itself. However, this must come at the cost of adding messages to the protocol. In fact, Paise and Vaudenay used that paradigm of enriching a two-message protocol to add a third "confirmation message" that is sent by the reader. As it turns out, the reader cannot be sure on whether it was successfully authenticated or not, which is a classical fairness problem.

Not only that, but having more than two messages for mutual authentication schemes also brings a disadvantage with regards to Forward adversaries. According to Definition 8.3, corruption leaks the entire state of the tag to the attacker, including all the temporary variables it has in store. Unfortunately, this model of corruption is too strong: No scheme achieves reader security and Narrow-Forward privacy at the same time. Indeed, there exists a generic attack in which an attacker blocks the last message of one protocol session and later corrupts a tag to simulate its answer on the blocked message. It the tag would have accepted the reader, then it must correspond to the anonymous tag that the attacker had in the first place. The attack is best described in the following steps.

| Tag |  | System |
|---|---|---|
| **State:** $pk$, ID, $K_{\text{ID}}$ |  | **Secret key:** $sk$<br>**Database:** $\{\dots, (\text{ID}, K_{\text{ID}}), \dots\}$ |
| Choose $b \in_R \{0,1\}^\beta$ | $\xleftarrow{\quad a \quad}$ | Choose $a \in_R \{0,1\}^\alpha$ |
| $c = \text{Enc}_{pk}(\text{ID}\|\| K_{\text{ID}}\|\|a\|\|)$ | $\xrightarrow{\quad c \quad}$ | Check that $\exists(\text{ID}, K_{\text{ID}}) \in \text{DB} : \text{Dec}_{sk}(c) = \text{ID}\|\|K_{\text{ID}}\|\|a\|\|b^\star$ |
|  | $\xleftarrow{\quad b^\star \quad}$ | If not found, set $b^\star \in_R \{0,1\}^\beta$ |
| **Output:** $b = b^\star$ |  | **Output:** ID or $\perp$ if not found |

**Figure 10.1:** A Public-Key Based Mutual Authentication Protocol.

1: $\textsc{CreateTag}(\text{ID}_0)$, $\textsc{CreateTag}(\text{ID}_1)$
2: vtag $\leftarrow$ $\textsc{DrawTag}(\Pr[\mathcal{T}(\text{vtag}) = \text{ID}_{b \in \{0,1\}}] = \frac{1}{2})$
3: Execute a protocol between vtag and the reader but stop before the last $\textsc{SendTag}(\text{vtag}, m)$ query and store $m$.
4: $\textsc{Free}(\text{vtag})$
5: $\text{vtag}_0 \leftarrow \textsc{DrawTag}(\text{ID}_0)$
6: $\text{vtag}_1 \leftarrow \textsc{DrawTag}(\text{ID}_1)$
7: $S_0 \leftarrow \textsc{Corrupt}(\text{vtag}_0)$
8: $S_1 \leftarrow \textsc{Corrupt}(\text{vtag}_1)$
9: Set a bit $b$ such that simulating a tag of state $S_b$ with the incoming message $m$ leads to output $1$ (if no or both $S_b$ work, set $b$ to a random bit)
10: Get $\mathcal{T}$ and output whether $\mathcal{T}(\text{vtag}) = \text{ID}_b$

Due to correctness of reader authentication, the tag outputs $1$ with negligible probability when fed with message different from $m$ or with a non-final message $m$ and $0$ with negligible probability when fed with the final message $m$. So, if $p$ is the probability for guessing the last query right, the adversary wins with probability close to $p$.

For any blinded adversary, tags run no protocol so there is a negligible probability for getting an m leading to $1$, the probability for winning is close to $1$ . Hence, the advantage is $p/2$ which is non-negligible for any blinder. So, the adversary is significant.

To overcome this limitation, Paise and Vaudenay chose to alter the Free interface and make it erase the tag's temporary memory. As temporary memory typically needs a power source to be maintained, it is reasonable to assume that its contents fade away when a passive tag is not powered by the reader. For semi-active and active tags, we have to ensure that this content is automatically and safely erased after some delay without receiving the expected message.

## 10.4 Public-Key Based Mutual Authentication RFID Scheme

The rest of this chapter is dedicated at studying the privacy of the enriched public-key based protocol of Section 8.4.3 that we also used for Strong privacy in Section 9.8. In short, we alter the protocol in that we make the tag generate a random $\beta$-bit string $b$ and add it to the message that is encrypted and sent to the reader. Using its secret key, the reader recovers $b$ and sends it back to the tag. The latter authenticates the reader if the value it receives is consistent with its $b$. The scheme in more details is defined as follows.

1. **Setup.** This algorithm calls upon the encryption scheme's KeyGen with the same security parameter to produce a key pair $(sk, pk)$ that is forwarded as the output.
2. **SetupTag.** To create a tag ID, pick uniformly a $k$-bit key $K_{\mathsf{ID}}$ and set $S_{\mathsf{ID}} = (pk, \mathsf{ID}, K_{\mathsf{ID}})$. When the tag is legitimate, the entry $(\mathsf{ID}, K_{\mathsf{ID}})$ is put in the database.
3. **Authentication.** As shown in Figure 10.1, the protocol runs as follow.
   a) First, the reader picks a $\alpha$-bit string $a$ and sends it to the tag.
   b) The later picks a random $\beta$-bit string $b$ and computes $c = \mathsf{Enc}_{pk}(\mathsf{ID}\|K_{\mathsf{ID}}\|a\|b)$ and replies with this value.
   c) Upon reception of $c$, the reader decrypts it, parses the plaintext as $\mathsf{ID}\|K_{\mathsf{ID}}\|a'\|b'$, and checks the correctness of the challenge, i.e., whether $a' = a$. In case of failure, the reader aborts, outputs $\perp$. In the other case, it looks in its database for the pair $(\mathsf{ID}, K_{\mathsf{ID}})$, outputs ID and sends $b'$ in case of success. Otherwise, the reader outputs $\perp$. Note that anytime the authentication for the reader fails, it sends a uniformly chosen $\beta$-bit string $b$ to the tag.
   d) Finally, the tag authenticates the reader, i.e., outputs $1$, if $b' = b$. Otherwise, it outputs $0$.

In the next section, we challenge Paise and Vaudenay's result concerning the Narrow-Strong privacy of the scheme when the encryption scheme is IND-CPA secure. At the same time, we prove that it still meets Narrow-Forward privacy and thus Forward privacy if the scheme is IND-CCA2 secure. After that, we instantiate the protocol with an IND-CCA2 and PA1+ plaintext-aware encryption scheme and show that the resulting protocol is Strong private.

## 10.5 IND-CCA2 Security is Insufficient for Narrow-Destructive Privacy

In this section, we show an attack against the privacy of the IND-CPA based protocol, depicted in Figure 10.1 but instantiated with an IND-CCA2 secure encryption scheme, that Paise and Vaudenay proved to be Narrow-Strong private. Intuitively, the reason why this protocol cannot achieve privacy is that, due to the $b$ message, the reader is acting as a partial decryption oracle for the queries of the adversary. Therefore, the results of Paise and Vaudenay regarding the Narrow-Strong privacy of their scheme does not hold. We stress that this result is not valid when the blinder has access to the adversary's randomness. The following theorem formalizes the attack.

**Theorem 10.1**
*There exists an effective Narrow-Destructive adversary against the RFID scheme of Figure 10.1 when the encryption scheme is only IND-CPA secure and blinders are not given the adversary's random tape.*

**Proof.** Let us assume an adversary $\mathcal{A}$ who creates a single tag and simulates it to the reader. The adversary then outputs 1 if the verification from the tag perspective succeeded. In other words, we consider the following adversary,

```
$\textsc{CreateTag}(\mathsf{ID})$
$\mathsf{vtag} \leftarrow \textsc{DrawTag}(\mathsf{ID})$
$(\mathsf{ID}, K_{\mathsf{ID}}) \leftarrow \textsc{Corrupt}(\mathsf{vtag})$
$\textsc{Launch} \to \pi$
$\textsc{SendReader}(\pi, -) \to a$
pick $b \in_R \{0,1\}^\beta$
compute $c \leftarrow \mathsf{Enc}_{pk}(\mathsf{ID}\|K_{\mathsf{ID}}\|a\|b)$
$\textsc{SendReader}(\pi, c) \to \hat{b}$
Output $\neg(b \oplus \hat{b})$
```

As the reader correctly decrypts and recovers $b$, the adversary outputs 1 with probability 1. On the other hand, Narrow-Destructive privacy implies the existence of a blinder $B$ such that $\mathcal{A}^B$ outputs 1 with probability roughly equal to 1.

The blinder $B$ is an algorithm which works as follow. It first receives a public key $pk$ and a pair $(\mathsf{ID}, K_{\mathsf{ID}})$ to generate a bit-string $a$. After that, it gets a ciphertext $c$ whose underlying plaintext is partially known and returns the decryption of the unknown part. In other words, $B$ is an algorithm which works in two phases. In the first phase, it takes as input a public key $pk$ and a pair $(\mathsf{ID}, K_{\mathsf{ID}})$ and returns a randomly chosen $a$. In the second phase, the blinder receives the encryption of $\mathsf{ID}\|K_{\mathsf{ID}}\|a\|b$ denoted $c$, where $b$ is a $\beta$-bit string, and outputs $b$.

We now show that the existence of such blinder leads to a chosen plaintext attack against the public-key encryption scheme. We consider an adversary playing the IND-CPA game. It first picks a random $\mathsf{ID}$ and runs $\mathsf{SetupTag}_{pk}(\mathsf{ID}) \to (K_{\mathsf{ID}}, S_{\mathsf{ID}})$. It then feeds $B$ with the pair $(\mathsf{ID}, K_{\mathsf{ID}})$ along with the public key $pk$ and gets $a$ in return. After that it pick two $\beta$-bit strings $b_0$ and $b_1$ and submits the messages $\mathsf{ID}\|K_{\mathsf{ID}}\|a\|b_0$ and $\mathsf{ID}\|K_{\mathsf{ID}}\|a\|b_1$ to the IND-CPA challenger who tosses a coin $\delta$ and returns $c \leftarrow \mathsf{Enc}_{pk}(\mathsf{ID}\|K_{\mathsf{ID}}\|a\|b_\delta)$. $\mathcal{A}$ forwards that ciphertext to $B$ and retrieves $b_\delta$. Finally, it outputs $\delta$. Clearly, since the blinder recovers the correct $b_\delta$ with probability $1 - \mathsf{negl}$, the adversary's advantage is equal to $1/2 - \mathsf{negl}$ so the scheme is not IND-CPA secure. □

Note that this counterexample does not hold when the blinder is given the random coins of the adversary. Indeed, a blinder would generate $b$ like $\mathcal{A}$ and output $b$. Nevertheless, the point of this result is to show that Paise and Vaudenay's proof of Narrow-Strong privacy is flawed.

## 10.6 Narrow-Forward Privacy from IND-CCA2 Security

Despite not being Narrow-Strong private, we can still prove that the scheme of Figure 10.1 is Narrow-Forward private when instantiated with an IND-CPA secure public-key encryption scheme as the next theorem says. Nevertheless, it is rather easy to find that the resulting scheme is not secure. Instead, we prove that security is satisfied when the encryption scheme is IND-CCA2 secure and it also follows that the protocol becomes Forward private.

**Theorem 10.2**
*If the encryption scheme of Figure 10.1 is IND-CPA secure then the scheme is correct and Narrow-Forward private. Furthermore, if the cryptosystem is IND-CCA2 secure, then the scheme is secure and Forward private.*

We divide our proof in four parts. In the first part, we show that the scheme is correct and secure for the reader. We then demonstrate that it is secure for the tags. Finally, we prove that it is Narrow-Forward private. We conclude that it is Forward private using Lemma 8.2.

### 10.6.1 *Correctness and Security for the Reader*

Correctness is trivially induced by the correctness of the public-key encryption. Regarding security for the reader, it follows from Theorem 9.4 based on IND-CCA2 security.

### 10.6.2 *Security for the Tags*

We let the security experiment played by a fixed $\mathcal{A}$ that has her environment simulated by $\mathcal{B}$. The later has access to a decryption oracle that it uses to simulate the queries that require the secret key, namely SendReader$(c, \pi)$ and Result$(\pi)$. For that, it just queries the decryption oracle with $c$, gets a bit-string that it matches against $\mathsf{ID}\|K_{\mathsf{ID}}\|a$ and returns the last $\beta$ bits of the recovered plaintext in case of success. Otherwise, it returns a random bit-string. The same procedure is used to decide on the success of a protocol session.

Again, we let $S_i$ be the event that the adversary wins the security experiment in Game $i$.

*Game 0.* Let $S_0$ be the event that the adversary wins the security experiment. Note that the adversary does not issue a SendReader$(c, \pi)$ on the target session $\pi$ that induces a matching conversation. (This is the unique value that makes the tag accept and the adversary win so getting it from SendReader results in a matching conversation.)

*Game 1.* We make a change in Game 0 and define Game 1 as being the same except that all queries Sendreader$(-, \pi)$ never produce an $a$ that was sent before to the SendTag interface. In other words, we require that $\mathcal{A}$ never guesses $a$. As $a$ is chosen uniformly, when $\mathcal{A}$ makes $s$ calls to SendTag, the probability of this event happening is bounded by $s2^{-\alpha}$ so that

$$|\Pr[S_0] - \Pr[S_1]| \leq s2^{-\alpha}.$$

Since $s$ is polynomially bounded, this probability is negligible when $2^{-\alpha}$ is negligible.

*Game 2.* We now modify the way $\mathcal{B}$ handles SENDREADER$(c, \pi)$ queries in instances that have either no matching conversation but $c$ was the output of a SENDTAG query or a matching conversation with an illegitimate tag. For those, $\mathcal{B}$ returns uniformly distributed $b^{\star}$'s.

Simulation is perfect as illegitimate tags get rejected with probability $1$ (The database does not contain their corresponding entry) and ciphertexts that embed a reader challenge different from the one of the instance provoke the failure of the comparison after decryption. In other words,

$$\Pr[S_2] = \Pr[S_1]$$

*Game 3.* We further adapt $\mathcal{B}$'s behavior regarding SENDREADER$(c, \pi)$ queries for sessions that have matching conversation with legitimate tags. Since $\mathcal{B}$ is also handling SENDTAG queries, it knows the plaintext corresponding to the $c$ sent in an instance with matching conversation. We thus modify $\mathcal{B}$ so that it keeps a table of pairs $(a, b)$ for every ciphertext produced for a legitimate tag. This way, $\mathcal{B}$ does not need to access its decryption oracle for $c$'s that were produced by legitimate tags in matching sessions. Clearly, we have that

$$\Pr[S_3] = \Pr[S_2]$$

*Game 4.* We now alter the SENDTAG$(a, \mathsf{vtag})$ interface so that instead of computing the encryption of $\mathsf{ID}\|K_{\mathsf{ID}}\|a\|b$, it encrypts a random $R$ of the same length. (Recall that no such output is sent by $\mathcal{B}$ to the decryption oracle.)

We now construct a hybrid argument to show that $|\Pr[S_4] - \Pr[S_3]|$ is negligible. We construct the hybrids as follow: $\mathcal{B}(i)$ is an algorithm simulating $\mathcal{A}$ for which the $i$ first SENDTAG$(a_i, \mathsf{vtag})$ queries are treated by picking a random $b_i$ and encrypting $\mathsf{ID}\|K_{\mathsf{ID}}\|a_i\|b_i$. The rest of the queries are processed by encrypting random strings. We let $\mathcal{C}$ denote an adversary playing the IND-CCA2 game, simulating $\mathcal{B}_2(i)/\mathcal{B}_2(i+1)$, that submits $\mathsf{ID}\|K_{\mathsf{ID}}\|a_i\|b_i$ (as in $\mathcal{B}_2(i+1)$) and $R$ (as in $\mathcal{B}_2(i)$) to the IND-CCA2 challenger who randomly chooses one of the messages and returns its encryption. $\mathcal{C}$ then continues $\mathcal{B}_2(i)/\mathcal{B}_2(i+1)$'s execution and returns its output. The difference in the output of $\mathcal{B}_2(i)$ and $\mathcal{B}_2(i+1)$ can be expressed as a distinguisher advantage for the IND-CCA2 game which is negligible by assumption. Therefore, we find that

$$|\Pr[S_4] - \Pr[S_3]| = \mathsf{negl}(k).$$

At this point, $\mathcal{A}$ is receiving messages that are unrelated to $b$. Therefore, the only way for her to win the game is to guess $b$ which happens with probability $2^{-\beta}$. Therefore, the scheme is secure. □

### 10.6.3 *Privacy*

To prove privacy, we reduce a fixed Narrow-Forward adversary to a one playing against the corresponding one-way authentication protocol, i.e., the same protocol without the last message and reader authentication. Recall that this protocol is Narrow-Strong private. Therefore, we only need to construct a blinder for the SendReader$(c, \cdot) \to b$ interface. However, keeping the soundness of the proof requires us to split this simulation in two steps: We first take care of the case in which an instance fails. We then proceed as in

Basically, the blinder for SendReader returns uniformly distributed $b$'s. To show that this simulation is indistinguishable from the $b$ sent by the reader, we proceed in a number of games. We denote by $S_i$ the event that the adversary wins Game $i$.

*Game 0.* We let this game be the original privacy game played by a Narrow-Forward adversary $\mathcal{A}$. Recall that privacy requires $\mathcal{A}$'s SendTag and SendReader, in its two variants, queries to be simulated.

*Game 1.* We first eliminate the case in which the adversary submits a $c$ that was not the answer of any SendTag query. Since the transcript of the instance would have no matching conversation, security ensures that the reader outputs $\perp$ and chooses a random $b^\star$ for its answer. We find that

$$|\Pr[S_1] - \Pr[S_0]| \leq \Pr[E]$$

Therefore, we make $B$ outputting random $\beta$-bit strings.

*Game 2.* We proceed similarly for the case in which $c$ is the output of a SendTag$(a^\star, \mathsf{vtag})$ query in which $\mathsf{vtag}$ is an illegitimate tag (This information is yield by DrawTag.) or $a^\star$ was not sent by the first SendReader query of the same session, i.e., the conversation $(a^\star, c)$ is not matching. Since decryption yields an $a$ that is different from the one sent for the session, authentication fails with probability $1$ so the reader outputs a uniformly distributed $b^\star$. In this case, the blinder's simulation is perfect.

*Game 3.* At last, we have an adversary who only sends $c$'s that were produced by legitimate tags on sessions with matching conversation. Consequently, the answer from the SendReader interface will consist of a $b^\star$ that is equal to the $b$ that was picked by the tag. Clearly, the adversary has no information on this $b$ except that it is contained in the ciphertext $c$. More formally, we use the IND-CPA property of the encryption scheme to change $c$ to a random value. In other words, we construct a blinder for both the SendTag interface and the remaining queries to the SendReader interface.

We let $B(i)$ be the hybrid blinder for which the $i$ first queries SendTag$(a, \mathsf{vtag}) \to c$ and the eventual subsequent SendReader$(c, \cdot) \to b^\star$ queries are handled by setting $c$ to be the encryption of a random $r$ of the same length as $\mathsf{ID}\|K_{\mathsf{ID}}\|a\|b$ and $b^\star$ is picked randomly while the rest of the queries are processed in the usual way.

We now construct an adversary $\mathcal{C}$ playing the encryption scheme's IND-CPA game. As it receives the public-key, $\mathcal{C}$ simulates the whole RFID system and runs either $\mathcal{A}^{B(i)}$ or $\mathcal{A}^{B(i+1)}$ (Remark that no query requires the secret key). The first $i-1$ SendTag queries are made to the IND-CCA2 challenger in a real-or-random version. The challenge ciphertext $c$ in the IND-CCA2 game is the answer from the challenger. It is either a real answer (as in the $\mathcal{A}_1^{B_2^i}$ simulation) or a simulated one (as in the $\mathcal{A}_1^{B_2^{i+1}}$ simulation). Note that no Result query is made on the session in which the adversary sent $c$ (this case has been taken care of in Game 1). So, $\mathcal{C}$ prefectly simulates either the game for $\mathcal{A}_1^{B_2^i}$ or the game for $\mathcal{A}_1^{B_2^{i+1}}$ and is an IND-CCA2 adversary. Since $\mathcal{C}$ produces the output of $\mathcal{A}_1^{B_2^i}/\mathcal{A}_1^{B_2^{i+1}}$, we obtain that

$$|\Pr[\mathcal{A}_1^{B_2^i} \to 1] - \Pr[\mathcal{A}_1^{B_2^{i+1}} \to 1]| \leq \mathsf{Adv}^{\mathsf{IND-CCA2}}(k),$$

*Game 4.* We now simulate the remaining Launch and SendReader queries. Regarding the former interface, the session identifiers are assumed to be deterministically generated so that it can be perfectly simulated. As for the latter one, we construct a blinder that returns uniformly distributed $\alpha$-bit strings. It is not hard to see that this simulation is perfect.

**Forward Privacy** Finally, Forward privacy follows from correctness and Narrow-Forward privacy using Lemma 8.2.

## 10.7 Strong Privacy with Mutual Authentication

We now consider the same protocol of Section 10.4 with the public-key encryption scheme being PA1+ plaintext-aware and IND-CC2 secure (As we detailled in Section 9.5.2, the Cramer-Shoup and Kurosawa-Desmetd encryption schemes meet this level of plaintext-awareness and security) and show that the resulting scheme is Strong private.

**Theorem 10.3**
*Assume that the public-key encryption scheme used in the RFID scheme of Figure 10.1 is correct, PA1+ plaintext-aware, and IND-CCA2 secure. If $2^{-\alpha}$, $2^{-\beta}$, and $2^{-\kappa}$ are negligible, then the scheme is correct, secure, and Strong private.*

The security proof is rather similar to the one of Theorem 9.4 but for the sake of completeness we include it in here.

We split the proofs into three parts. In the first part, we argue that the scheme is correct and secure for the reader using results from the corresponding one-way authentication protocol. We then prove security for the tags and finally Strong privacy.

### 10.7.1 *Correctness and Security*

Correctness is trivially induced by the correctness of the public-key encryption. Regarding security, it follows from Theorem 10.2.

### 10.7.2 *Strong Privacy*

To conduct the proof, we consider a Strong adversary $\mathcal{A}$ and construct a blinder iteratively. That is, we construct a sequence of partial blinders $B_1, \dots, B_5$ and let $\mathcal{A}_i = \mathcal{A}_{i-1}^{B_i}$ with $\mathcal{A}_0 = \mathcal{A}$. The final blinder for $\mathcal{A}$ is $B = B_1 \circ \cdots \circ B_5$. By showing that the outcome of $\mathcal{A}_i$ and $\mathcal{A}_{i+1}$ are computationally indistinguishable, we deduce that $B$ is indeed a full blinder for $\mathcal{A}$. So, the scheme is Strong private.

The outline of the proof is as follows. We first eliminate the case of instances matching conversations whose result is trivially $1$ or $0$ depending on whether the tag is a legitimate one.

*Game 0.* We first fix an adversary $\mathcal{A}_0$ playing the privacy game.

*Game 1.* We let Game 1 denote the privacy game performed by an adversary who simulates every RESULT on a session $\pi$ with a transcript $(a, c)$, such that $c$ that has been obtained by a previous SENDTAG$(\mathsf{vtag}, a')$ query.

If $a \neq a'$, $c$ does not decrypt to something containing $a$, so the answer to RESULT$(\pi)$ must be $0$. The simulation is easy and perfect. In the other case, that is, if $a = a'$, the decryption of $c$ will be parsed to a matching challenge $a$ and some entry $\mathsf{ID}\|K_{\mathsf{ID}}$ which is in the database if and only if vtag is legitimate. Fortunately, the blinder has access to this latter information as it is returned in the response of the DRAWTAG oracle query drawing vtag. Again, the simulation is easy and perfect. This fully defines $B_1$ and we deduce that

$$\Pr[\mathcal{A}_0 \to 1] = \Pr[\mathcal{A}_0^{B_1} \to 1]$$

Clearly, the outcome of $\mathcal{A}_0$ and $\mathcal{A}_1$ have identical distributions. We can thus define the adversary $\mathcal{A}_1$ that never queries RESULT on an instance $\pi$ in which the response $c$ was produced by a previous SENDTAG query.

*Game 2.* We let Game 1 denote the privacy game performed by an adversary who simulates every RESULT on a session $\pi$ with a transcript $(a, c)$, such that $c$ that has been obtained by a previous SENDTAG$(\mathsf{vtag}, a')$ query.

Assume that the reader in instance $\pi$ produced a challenge $a$. If $a \neq a'$, we are ensured that $c$ does not decrypt to something containing $a$, so the answer to RESULT$(\pi)$ must be 0. The simulation is easy and perfect. In the other case, that is, if $a = a'$, the decryption of $c$ will be parsed to a matching challenge $a$ and some entry $\mathsf{ID}\|K_{\mathsf{ID}}$ which is in the database if and only if vtag is legitimate. Fortunately, the blinder has access to this latter information as it is returned in the response of the DrawTag oracle query drawing vtag. Again, the simulation is easy and perfect. This fully defines $B_1$ and we deduce that

$$\Pr[\mathcal{A}_0 \to 1] = \Pr[\mathcal{A}_0^{B_1} \to 1]$$

Clearly, the outcome of $\mathcal{A}_0$ and $\mathcal{A}_1$ have identical distributions. We can thus define the adversary $\mathcal{A}_1$ that never queries Result on an instance $\pi$ in which the response $c$ was produced by a previous SendTag query.

*Game 3.* In this game, we make all SendTag queries being simulated by a partial blinder $B_2$. To achieve this, we let $r$ be number of SendTag queries and make a sequence of hybrid blinders $B_2^1, \ldots, B_2^{r+1}$ in which $B_2^i$ simulates the $i-1$ first SendTag queries. Note that $B_2^1$ does not make any simulation so $\mathcal{A}_1^{B_2^1}$ is exactly $\mathcal{A}_1$ and that $B_2^{r+1}$ is a partial blinder the SendTag oracle that we set to be $B_2$. The hybrid $B_2^{i+1}$ simulates the $i$ first encountered SendTag queries by encrypting random strings of same length as $\mathsf{ID}\|K_{\mathsf{ID}}\|a$.

To prove that $\mathcal{A}_1^{B_2^i}$ and $\mathcal{A}_1^{B_2^{i+1}}$ have computationally indistinguishable distributions, we construct an adversary $\mathcal{C}$ playing the IND-CCA2 game. Adversary $\mathcal{C}$ receives the public key and runs $\mathcal{A}_1^{B_2^i}$ or $\mathcal{A}_1^{B_2^{i+1}}$, depending on the bit of the indistinguishability game, while simulating the RFID system, except the $i$-th SendTag query. For that, $\mathcal{C}$ must simulate the environment for $\mathcal{A}_1^{B_2^i}/\mathcal{A}_1^{B_2^{i+1}}$. Since all algorithms and oracles of the scheme, except for Result, do not require the secret key, $\mathcal{C}$ can easily perform the simulation by itself. Regarding the Result oracle, $\mathcal{C}$ just queries a decryption oracle and checks whether the decrypted message matches.

The first $i-1$ SendTag queries are made to the IND-CCA2 challenger in a real-or-random version. The challenge ciphertext $c$ in the IND-CCA2 game is the answer from the challenger. It is either a real answer (as in the $\mathcal{A}_1^{B_2^i}$ simulation) or a simulated one (as in the $\mathcal{A}_1^{B_2^{i+1}}$ simulation). Note that no Result query is made on the session in which the adversary sent $c$ (this case has been taken care of in Game 1). So, $\mathcal{C}$ prefectly simulates either the game for $\mathcal{A}_1^{B_2^i}$ or the game for $\mathcal{A}_1^{B_2^{i+1}}$ and is an IND-CCA2 adversary. Since $\mathcal{C}$ produces the output of $\mathcal{A}_1^{B_2^i}/\mathcal{A}_1^{B_2^{i+1}}$, we obtain that

$$|\Pr[\mathcal{A}_1^{B_2^i} \to 1] - \Pr[\mathcal{A}_1^{B_2^{i+1}} \to 1]| \leq \mathsf{Adv}^{\mathsf{IND-CCA2}}(k),$$

and it results that

$$|\Pr[\mathcal{A}_1 \to 1] - \Pr[\mathcal{A}_1^{B_2} \to 1]| \leq r \cdot \mathsf{Adv}^{\mathsf{IND-CCA2}}(k),$$

which is negligible as $r$ is polynomially bounded and the scheme is IND-CCA2 secure.

At this point, we can legitimately consider an adversary $\mathcal{A}_2$ who makes no SendTag queries.

*Game 4.* We now simulate all remaining RESULT queries. To do so, we construct an adversary $\mathcal{E}$ playing the PA1++ game.

This adversary takes the public key then simulates $\mathcal{A}_2$ interacting with the RFID system. Recall that, like in Game 2, the algorithms and oracles of the scheme do not depend on the secret key, except for the RESULT queries that will be treated hereafter. We let $\mathcal{E}$ simulate the RFID system to $\mathcal{A}_2$, handling her queries as follow:

- Assuming w.l.o.g. that session identifiers are not randomized, LAUNCH is deterministically computed by $\mathcal{E}$.
- Upon a CREATETAG(ID) query from $\mathcal{A}_2$, $\mathcal{E}$ inserts $(\mathsf{ID}, -)$ in a table $\mathsf{DB}_1$ if the query asks for a legitimate tag. Otherwise, it inserts $(\mathsf{ID}, -)$ in a table $\mathsf{DB}_0$. oracle implementing the uniform distribution over $\{0,1\}^\kappa$ and assigns the answer to $K_{\mathsf{ID}}$.
- $\mathcal{E}$ simulates SENDREADER queries by asking the oracle $\mathcal{O}_S$ to sample from the uniform distribution over $\{0,1\}^\alpha$. It then forwards the received answer to $\mathcal{A}_2$.
- DRAWTAG(Samp) queries are handled by asking the randomness oracle $\mathcal{O}_S$ to sample from the distribution specified by Samp to get one or more random ID. If any of the returned identifiers corresponds to a drawn tag, $\mathcal{E}$ outputs $\perp$. Otherwise, it generates, deterministically and for each returned $\mathsf{ID}_i$, a fresh $\mathsf{vtag}_i$ and inserts the pair $(\mathsf{vtag}_i, \mathsf{ID}_i)$ in the table $\mathcal{T}$. After that, it sets the bit $b_i$ to $1$ if $\mathsf{ID}_i$ is legitimate, or to 0 otherwise. At last, it returns $(\mathsf{vtag}_1, b_1, \ldots, \mathsf{vtag}_n, b_n)$ to $\mathcal{A}_2$.
- CORRUPT(vtag) makes $\mathcal{E}$ reveal $\mathsf{ID} = \mathcal{T}(\mathsf{vtag})$. Moreover, $\mathcal{E}$ looks for the entry $(\mathsf{ID}, K_{\mathsf{ID}})$ in $\mathsf{DB}_0$ and $\mathsf{DB}_1$. If that corresponding entry contains a $K_{\mathsf{ID}}$ different from $'-'$, then it returns it. Otherwise, it queries $\mathcal{O}_S$ to sample from the uniform distribution over $\{0,1\}^\kappa$ and assigns the answer to $K_{\mathsf{ID}}$. It subsequently updates the entry $(\mathsf{ID}, -)$ to $(\mathsf{ID}, K_{\mathsf{ID}})$ and returns this last pair as its answer. $\mathcal{E}$ received from $\mathcal{O}_S$ for processing the CREATETAG(ID) query, are also returned. We further assume that whenever the tag ID is a legitimate one, $\mathcal{E}$ inserts the entry $(\mathsf{ID}, K_{\mathcal{T}(\mathsf{vtag})})$ in a table $\mathcal{T}_{\mathcal{E}}$.
- To simulate the RESULT$(\pi)$ oracle for a reader instance $\pi$ with transcript $(a, c)$, $\mathcal{E}$ sends $c$ to the decryption oracle, checks that the recovered plaintext is of the form $\mathsf{ID}\|K_{\mathsf{ID}}\|a$, that it matches a tag state $\mathsf{ID}\|K_{\mathsf{ID}}$ obtained from a previous CORRUPT(vtag) query, and finally that the corresponding tag vtag is legitimate. If this is the case, the answer to RESULT must be $1$, otherwise, the simulated answer is 0. Note that when the output of the $\mathcal{E}$ regarding a RESULT query is $1$, the genuine RESULT query would also have answered $1$. This is because that, knowing a subset of the database through corruption query, $\mathcal{E}$ can effectively predict the answer when the database entry lies in this subset. Errors in the simulation occur when $\mathcal{E}$ predicts $0$ and the genuine RESULT

query would also have outputted $1$ in a session without matching conversation. Clearly, the failure of one of $\mathcal{E}$'s simulations corresponds to the happening of the event that $\mathcal{A}_2$ wins the security game, that we denoted $E$ in the previous section. In other words,

$$\left|\Pr\left[\mathcal{A}_2 \to 1\right] - \Pr\left[\mathcal{A}_2^{\mathcal{E}} \to 1\right]\right| \leq \Pr[E]$$

This simulation is almost perfect since $E$ occurs with negligible probability. Since we assumed the encryption scheme to be PA1++ plaintext-aware, we can use the plaintext extractor $\mathcal{E}^\star$ of $\mathcal{E}$ to replace the decryption oracle without significantly altering the outcome distribution. However, $\mathcal{E}^\star$ requires the view of $\mathcal{E}$ instead of the view of $\mathcal{A}_2$, so we cannot use it as an extractor for $\mathcal{A}_2$. Fortunately, it is possible to reconstruct that view given the adversary's random tape and its queries. At first, we note that $\mathcal{E}$'s random coins are only used to initialize $\mathcal{A}_2$. Furthermore, all the randomness $\mathcal{E}$ obtains from $\mathcal{O}_S$ to process CORRUPT queries is revealed to $\mathcal{A}_2$. Moreover, since $\mathcal{A}_2$ is simulatable, we can use the algorithm $\mathcal{A}_2'$, induced by Definition 9.7, to reconstruct, from $\mathcal{A}_2$'s view, a table $\mathcal{T}'$ indistinguishable from $\mathcal{T}$. $\mathcal{E}$ received from $\mathcal{O}_S$ using all the $\mathsf{Samp}^{-1}$ algorithms. the real $\mathsf{ID}$ of all the $\mathsf{vtag}$ and . Since this table lists all the mappings between real and virtual identifiers, it is straightforward to reconstruct a randomness for $\mathcal{E}$ that she received to process the DRAWTAG queries using the $\mathsf{Samp}^{-1}$ algorithms corresponding to the sampling queries of $\mathcal{A}_2$. We let this whole operation be carried by a polynomial-time algorithm $\mathcal{V}$ that takes as input the view of $\mathcal{A}_2$ and uses $\mathcal{A}_2'$ to reconstruct a view of $\mathcal{E}$, i.e., it is such that $\mathcal{V}(\mathsf{view}_{\mathcal{A}_2})$ and $\mathsf{view}_{\mathcal{E}}$ are indistinguishable. It follows that $\mathcal{E}^\star(pk, \cdot, \mathcal{V}(\mathsf{view}_{\mathcal{A}_2}))$ and $\mathcal{E}^\star(pk, \cdot, \mathsf{view}_{\mathcal{E}})$ are indistinguishable.

interfaces CREATETAG, SENDREADER, and DRAWTAG is obtained through the randomness oracle. Moreover, all this randomness is given to the adversary through the SENDREADER and CORRUPT oracles (Here we make the assumption that the adversary corrupts all the RFID tags of the system. It easily generalizes to the general case as the randomness needed for the creation of uncorrupted tags is not effectively used by $\mathcal{E}$.). from the view of $\mathcal{A}$.

At this point, we are able to define $B_3$, the partial blinder for RESULT queries. adversary makes $r$ RESULT queries and define the successive blinders $B(1), \ldots, B(r+1)$ such that $B(i)$ simulates the $i-1$ first RESULT queries. Again, $B(1)$ does not simulate any query while $B(r+1)$ is a partial blinder for the RESULT oracle. Similarly to $\mathcal{E}$, we assume that $B_3$ maintains a table $\mathcal{T}_{B_3}$ containing a list of pairs $(\mathsf{ID}, K_{\mathsf{ID}})$ for corrupted legitimate tags. In order to simulate a RESULT query on an instance $\pi$ of transcript $(a, c)$, the blinder proceed as follow.

a) First, the blinder calls upon $\mathcal{E}^\star(pk, c, \mathcal{V}(\mathsf{view}_{\mathcal{A}_2}))$ to get $\mathsf{Dec}_{sk}(c) = \mathsf{ID}\|K_{\mathsf{ID}}\|a'$.

b) Then it verifies that $a = a'$ and outputs $0$ in case of failure. Otherwise, it continues.

c) At last, it outputs $1$ if the pair $\mathsf{ID}||K_{\mathsf{ID}}$ is listed in $\mathcal{T}_{B_3}$. Otherwise, it outputs $0$.

The probability that Step 1 fails can be expressed as a distinguisher advantage of the PA1++ game or between $\mathcal{V}(\mathsf{view}_{\mathcal{A}_2})$ and $\mathsf{view}_{\mathcal{E}}$, so

$$\left|\Pr\left[\mathcal{A}_2^{B_3} \to 1\right] - \Pr\left[\mathcal{A}_2^{\mathcal{E}} \to 1\right]\right| \leq \mathsf{Adv}^{\mathsf{PA1++}}(k) + \mathsf{negl}(k).$$

At the same time, Step 3 fails when the event $E$ occurs, so using triangle inequalities we conclude that

$$\left|\Pr\left[\mathcal{A}_2 \to 1\right] - \Pr\left[\mathcal{A}_2^{B_3} \to 1\right]\right| \leq \mathsf{Adv}^{\mathsf{PA1++}}(k) + \Pr[E] + \mathsf{negl}(k).$$

Recalling that $E$ occurs with negligible probability and that the scheme is PA1++ plaintext-aware, the quantity above becomes negligible. Hence, $B_3$ describes a successful blinder for the RESULT oracle. In the following, we denote $\mathcal{A}_2^{B_3}$ by $\mathcal{A}_3$.

*Game 5.* We now alter the game so that no adversary issues SENDREADER$(c, \cdot)$ queries. For that, we define a blinder $B_4$ for $\mathcal{A}_3$ that returns uniformly distributed values from the set $\{0,1\}^{\beta}$. Recall that no SENDTAG query has been issued by the adversary so $c$ must have been produced by the adversary. We proceed as in Game 3 and make $B_4$ follow the same strategy of $B_3$ to recover $c$'s decryption and decide of the outcome of the protocol session. If $B_3$'s decision for the result of the session is success, then $B_4$ outputs the last $\beta$ bits of $c$'s decryption corresponding to $b$. Otherwise, it picks a random $b$ and returns it.

To show that the simulation is indistinguishable, we notice that when $B_4$ gets the correct decryption of $c$ and $B_3$ correctly computes RESULT$(\cdot)$, then it can perfectly simulate the reader. Therefore,

$$\begin{aligned}\left|\Pr\left[\mathcal{A}_3 \to 1\right] - \Pr\left[\mathcal{A}_3^{B_4} \to 1\right]\right| &\leq \left|\Pr\left[\mathcal{A}_2 \to 1\right] - \Pr\left[\mathcal{A}_2^{B_3} \to 1\right]\right| \\ &= \mathsf{negl}(k)\end{aligned}$$

*Game 6.* In this game, we eliminate the remaining SENDREADER queries. This can easily be achieved by constructing a blinder $B_5$ that returns uniformly distributed values from the set $\{0,1\}^{\alpha}$. Clearly, simulation is perfect as both distributions are perfectly indistinguishable. Hence,

$$\Pr\left[\mathcal{A}_4 \to 1\right] = \Pr[\mathcal{A}_4^{B_5} \to 1]$$

*Game 7.* Finally, we have an adversary $\mathcal{A}_5$ who only produces LAUNCH queries. Since these are generated deterministically, we can trivially simulate the LAUNCH oracle using a blinder $B_6$. It follows that

$$\Pr\left[\mathcal{A}_5 \to 1\right] = \Pr[\mathcal{A}_5^{B_6} \to 1]$$

In the end, we have obtained an adversary $\mathcal{A}_6 = \mathcal{A}^B$, with $B = B_1 \circ \cdots \circ B_6$, who does not produce any oracle query that is such that

$$\left|\Pr[\mathcal{A} \to 1] - \Pr[\mathcal{A}^B \to 1]\right| = \mathsf{negl}(k)$$

The scheme is thus Strong private.

11

# CONCLUSION

Contents

The RFID technology is promising with several upcoming evolutions that will hopefully lead them to a widespread development and a general consensus on their benefits. In particular, two aspects on which current RFID tags should be improved are security and privacy. While the specific constraints put on these lightweight devices denied the use of classical cryptographic primitives, we presented an assessment of the security of two original designs, HB$^{\#}$ and SQUASH. The second part of the thesis was dedicated to studying the level of privacy RFID tags can offer.

The main contributions of this thesis are summarized in the list below.

1. We showed that the HB$^{\#}$ protocol is insecure against man-in-the-middle attacks. This gave a negative answer to a conjecture by Gilbert et al. that claimed otherwise.
2. We invalidated SQUASH's security argument by mounting an attack against its earlier variant, SQUASH-0, that stands on the same security assumption. Although our attack does not compromise the security of SQUASH's final proposal, it showed that its security is unrelated to factoring.
3. To emphasize the need for a framework assessing privacy and the importance of studying protocols in such a framework, we illustrated how several authentication protocols dedicated to RFID tags compromise privacy. The list of these protocols include ProbIP, MARP, Auth2, YA-TRAP, YA-TRAP+, O-TRAP, RIPP-FS, and the Lim-Kwon protocol.
4. We also argued that protocols proven private in the UC-based model of Le, Burmester and de Meideros, are still vulnerable to privacy attacks that have a practical sense. We took for examples, O-FRAP and O-FRAKE.
5. We reformulated Vaudenay's definition of privacy. We also incorporated two flavors of correctness, depending on whether it is ensured in an absolute or contextual sense. We also clarified the way adversaries formally select tags.
6. We studied the relation of Vaudenay's model with the extended-Juels-Weis privacy model and the the ZK-privacy model. We did that by illustrating protocols that can be proven to be private in their model, but fail to meet our standard notion of privacy.
7. We also analyzed variants of Vaudenay's privacy model that were meant to either simplify the definitions, such as the HPVP model, or to make Strong privacy possible such as the proposal of Ng et al. We showed that the former model fails to capture real-world attackers capabilities. We also argued that the notion of wise adversaries proposed by Ng et al. fails to justify in practical attack scenarios.
8. We corrected Vaudenay's definition of privacy and showed that with the new definition Strong privacy is achievable. We then used encryptions schemes' notion of plaintext-awareness to instantiate a protocol achieving this level of privacy.
9. We illustrated a separation between two notions of security for encryption schemes, namely IND-CCA2 on one side and IND-CPA coupled with PA2 on the other side.

Albeit Bellare and Palacio showed that the two latter notions together imply the former, it was not clear whether plaintext-awareness could serve any purpose that IND-CCA2 security could not meet. We proved that using an IND-CCA2 secure encryption scheme does not yield a Strong private protocol and instantiating the same protocol with an IND-CPA secure and PA2 plaintext-aware encryption scheme results in a Strong private protocol.

10. We extended our results to protocols offering mutual authentication, i.e., in which the reader is also required to authenticate to a tag. In these settings, we showed that our definition of privacy invalidates the results obtained by Armknecht et al. concerning the PV model. While they have demonstrated that no secure protocol can be Narrow-Strong private with Vaudenay's definitions, we showed that Strong privacy, in our settings, with mutual authentication is achievable.
11. We proposed a tradeoff for Forward privacy by lowering its requirements by a small margin to allow protocols using lightweight cryptography to achieve a certain form of forward privacy. Concretely, these protocols ensure the privacy of all the tags' actions that occurred before their secrets leaked to the adversary except for the last session in which a tag was involved before corruption if it did not end properly.

## 11.1 The Security of RFID Primitives

The first part of thesis was dedicated to analyzing the security of dedicated cryptographic primitives for RFID tags.

### 11.1.1 *Our Contributions*

Regarding the security of primitives dedicated to RFID tags, we mainly gave two contributions in analyzing the security of the protocol HB# and the message authentication code SQUASH.

**The Security of the HB# Protocol.** We first challenged the conjecture establishing the security of HB# against man-in-the-middle adversaries. We showed that if an adversary can alter all messages transiting through the wireless channel set between a tag and a reader and if she has access to the result of each protocol session, she can recover the tag's secret without tampering. We provided complexity analysis of the attack and showed a bound on the parameter that separated the case in which the attack is asymptotically polynomial from exponential. The first parameter set proposed by the authors of HB# fell into the case in which the attack is polynomial and we showed that the tag's secret can be retrieved by solving a system of linear equations after disturbing messages in $2^{20}$ protocol instances. The attack complexity for parameter set II is higher as the attack is exponential in the security parameter. Nevertheless, we were still able to retrieve the secret after disturbing $2^{35}$ protocol sessions and solving a system of linear equations.

Possible fixes to render HB$^{\#}$ immune to man-in-the-middle attacks were also analyzed. We looked at the possibility of lowering the error threshold or to bound the number of errors the prover introduces in its answer so that it always gets accepted. Unfortunately, both solutions turned out to be also vulnerable to variants of the attacks on HB$^{\#}$.

**SQUASH.** The second primitive we studied is the message authentication code SQUASH. We concentrated on its security arguments and its connection to the Rabin encryption scheme. We separated the security of the two primitives by mounting an attack against the earlier version of SQUASH that enables an adversary who has access to an oracle returning the MAC of messages it receives to recover the secret key. This attack scenario readily applies to challenge-response protocols based on a MAC where the challenger, i.e., the reader in the context of RFIDs, sends a message to the prover, i.e., the tag in the context of RFIDs, which replies with the MAC of the received message.

In the end, our attack strategy allows us to recover SQUASH's secret keys using $2^{10}$ messages if the modulus $2^{1277} - 1$ is used for Rabin's function. Replacing SQUASH-128's NLFSR with a linear one, we were able to recover the secret key using $64$ queries to the MAC oracle.

### 11.1.2 *Further Work*

Although new protocols based on the LPN problem were proposed and even proven secure against man-in-the-middle attacks, they rely on other components than simple XOR operations. For instance, the MAC constructed by Kiltz et al. uses a secret pairwise independent permutation which in itself needs a large secret key to be added to the one for LPN problem. Basically, the reason for the introduction of this component is to break the linearity of the protocol and thwart the kind of attacks we succeeded in mounting. In this sense, the MAC and subsequent protocol they propose is not entirely built around the LPN problem as HB-related protocols are. Although linearity provides nice implementation properties, our attack proved it to be a bad feature for security. Therefore, in order to obtain a secure version of an HB protocol, it is necessary to design a variant that uses non-linear components. For efficiency purposes, the perfect protocol would not rely on any other primitive than the LPN problem.

This linearity property is also at the center of our attack on SQUASH. As a consequence, it is very probable that SQUASH could be broken if a linear approximation of the mixing function could be found. Still, as we have shown that SQUASH's security is unrelated to Rabin's, it would be interesting to compare SQUASH's security with a version stripped from Rabin's squaring, i.e., a MAC that outputs a window of bits from an NLFSR initialized with the key and a message.

## 11.2 Privacy in RFID Protocols

The second part of this thesis was devoted to developing our privacy model for RFID systems.

### 11.2.1 *Our Contributions*

In this part of the dissertation, we concentrated on privacy issues related to RFID systems and developed a model for assessing which level of privacy, if any, an RFID authentication protocol achieves.

**The Need for a Privacy Model.** The first step toward proposing our privacy model was to emphasize the need for having one. For that, we used a basic ad hoc model, inspired by the literature of key-exchange protocols and the work of Bellare, Pointcheval, and Rogaway. That model only captured the notion of unlinkability, i.e., that a protocol is private if no adversary can give a relation between tags that were involved in protocol instances with the reader. In line with cryptography's classical adversarial models, the adversary is assumed to have full control over the communication channel.

Despite working with an incomplete model, we were able to use it to show that several protocols, namely ProbIP, MARP, Auth2, YA-TRAP, YA-TRAP+, O-TRAP, RIPP-FS, and the Lim-Kwon protocol, fail to be privacy concealing.

**The Shortcomings of the LBdM Privacy Model.** We also used that model to study two protocols that were proven to be forward private in the LBdM model, namely O-FRAP and O-FRAKE. As it turns out, both protocols fail to meet this notion in our model, hereby raising doubts on the pertinence of the LBdM model.

**Vaudenay's Privacy Model.** We clarified some notions in Vaudenay's privacy model such as how tags are selected by the adversary. For that, Vaudenay used a vague term of distribution that is queried to an interface. We formalized this capability by saying that the adversary submits the description of a sampling algorithm which running time is bounded by a polynomial in the security parameter. We also gave two definitions for the correctness of an RFID protocol. The stronger notion states that whatever happens in a system, a tag running an undisturbed protocol instance with the reader will end up being authenticated by the latter. The weaker version, proposed to reflect on several protocols proposed in the literature, requires correctness to only hold if the tag has not been involved in more than a certain number of consecutive sessions without being authenticated by the reader.

**Relation with Other Models and Variants** We compared Vaudenay's model with several other privacy models dedicated to RFIDs. We looked at the relationship between Vaudenay's privacy model and both the Juels-Weis and zk-privacy model. We showed that the former is superseeded by one of the weakest adversarial classes of Vaudenay's model. The latter was also shown to have issues related to concurrent attacks, i.e., privacy attacks in which the adversary interacts with more than one tag to compromise privacy. Using that, we were able to show the existence of authentication protocols that would be considered as private in the sense of zk-privacy but fail to be so in Vaudenay's model.

We also studied the HPVP variant of Vaudenay's model that was meant to simplify its formulation while retaining its semantic. However, we found that the variant cannot stand the existence of more than one RFID system, i.e., it does not tolerate the possibility of tags that do not belong to that RFID system. Vaudenay's model takes this eventuality into account. Moreover, adversaries cannot tamper with unknown tags: This means that the model denies the possibility of an adversary getting a random tag on which she has no information and extracting its secret. This prohibition fails to justify in practical scenario attacks.

**The Exact Notion of Privacy.** We argued that Vaudenay's definition of privacy is too strong for the notion it aims to formalize and that this mismatch is the cause of the impossibility of Strong privacy. We corrected this definition by requiring the entity responsible for producing fake messages to the adversary that would be unnoticeable to the adversary to have access to all the adversary's knowledge. In particular, this includes her random tape, which was missing from Vaudenay's definition. With our new definitions, Vaudenay's impossibility results does not hold and we showed that it is possible to achieve the strongest notion of privacy using a secure and plaintext-aware encryption scheme. At the same time, we proved that an IND-CCA2 secure encryption is insufficient for Strong privacy.

We also discussed an earlier attempt to obtain Strong privacy due to Ng et al., which introduced the rather artificial class of wise adversaries that do not issue queries for which they already know the answer. Besides being hard to define and manipulate, the notion of wise adversaries is hard to motivate.

**From Unilateral to Mutual Authentication.** We extended our results to cover protocols which offer mutual authentication. In this regard, the corrected formulation of privacy discards the results of Armknecht et al. relative to the impossibility of achieving Narrow-Strong privacy. Moreover, we show that Strong privacy with mutual authentication is achievable using plaintext-aware encryption schemes.

We also revisited Paise and Vaudenay's Narrow-Strong private protocol, which security theorem was invalidated by Arkmecht et al., and showed that the protocol is still Forward private.

### 11.2.2 *Further Work*

Further extensions to the model for diverse concrete senario can explored. For instance, we assume in our model that the adversary is able to learn the outcome of authentication. However, several deployed RFID applications give more information to the adversary in leaking the *identifier* of the tag it authenticated. Although this is an undeniable privacy loss, we could study the impact of learning such an information on other ones that the adversary can try to obtain without relying on the reader.

An extension towards addressing read/write only tags can also be envisaged for that these types of tags are still the most commonly used in real-world applications. Such type of tags only provide two interfaces that can be remotely accessed, one for reading the contents of its memory and another one to set it to a value specified in the command. It is rather easy to see that from a classical cryptographic point of view, no privacy can be achieved if the tag performs no computations unless we assume that the adversary does not completely control the communication channels. The goal of the model in here would be to measure the best privacy protection such tags can offer.

## 11.3 Final Notes

The notion of blinder is a powerful tool for assessing the privacy of an RFID system. Yet, we believe that it could be used for other types of cryptographic protocols. At first, it could be used in key exchange protocols. For the similarities these latter share with RFID protocols, it would be rather straightforward to translate the definitions from one setting to the other. The paradigm could also be used to deployed key establishment Internet protocols such as SSL/TLS. Similarly, we can strengthen zero-knowledge protocols with requiring that not only the verifier learns anything from a protocol execution but also that no other party can deduce any information.

On another side, the definitions of plaintext-awareness may be improved to make the knowledge extractor able to filter the eventual auxiliary information the ciphertext creator gets.

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF DEFINITIONS

## List of Tables

# BIBLIOGRAPHY

[ABF$^+$08] Ali Can Atici, Lejla Batina, Junfeng Fan, Ingrid Verbauwhede, and Siddika Berna Örs. Low-cost implementations of NTRU for pervasive security. In *19th IEEE International Conference on Application-Specific Systems, Architectures and Processors, ASAP 2008, July 2-4, 2008, Leuven, Belgium*, pages 79–84. IEEE Computer Society, 2008. 4

[ACPS09] Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Shai Halevi, editor, *Advances in Cryptology - CRYPTO 2009, 29th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2009. Proceedings*, volume 5677 of *Lecture Notes in Computer Science*, pages 595–618. Springer, 2009. 31

[ADO06] 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, 12th International Workshop, SAC 2005, Kingston, ON, Canada, August 11-12, 2005, Revised Selected Papers*, volume 3897 of *Lecture Notes in Computer Science*, pages 291–306. Springer, 2006. 114

[AGV09] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In Omer Reingold, editor, *Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, San Francisco, CA, USA, March 15-17, 2009. Proceedings*, volume 5444 of *Lecture Notes in Computer Science*, pages 474–495. Springer, 2009. 31

[AHMNP10] Jean-Philippe Aumasson, Luca Henzen, Willi Meier, and María Naya-Plasencia. Quark: A lightweight hash. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, volume 6225 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 2010. 4

[AO05] Gildas Avoine and Philippe Oechslin. RFID traceability: A multilayer problem. In Andrew S. Patrick and Moti Yung, editors, *Financial Cryptography and Data Security, 9th International Conference, FC 2005, Roseau, The Commonwealth of Dominica, February 28 - March 3, 2005, Revised Papers*, volume 3570 of *Lecture Notes in Computer Science*, pages 125–140. Springer, 2005. 7

[ASS$^+$10] Frederik Armknecht, Ahmad-Reza Sadeghi, Alessandra Scafuro, Ivan Visconti, and Christian Wachsmann. Impossibility results for RFID privacy notions. *Transactions on Computational Science XI - Special Issue on Security in Computing, Part II*, 6480:39–63, 2010. 8, 157, 168

[Avo05] Gildas Avoine. *Cryptography in radio frequency identification and fair exchange protocols.* PhD thesis, Thèse N° 3407, Lausanne, Switzerland, 2005. Available at http://library.epfl.ch/en/theses/?nr=3407. 114

[BBEG09] Côme Berbain, Olivier Billet, Jonathan Etrog, and Henri Gilbert. An efficient forward private RFID protocol. In Ehab Al-Shaer, Somesh Jha, and Angelos D. Keromytis, editors, *Proceedings of the 2009 ACM Conference on Computer and Communications Security, CCS 2009, Chicago, Illinois, USA, November 9-13, 2009*, pages 43–53. ACM, 2009.

[BCD06] Julien Bringer, Hervé Chabanne, and Emmanuelle Dottax. HB$^{++}$: a lightweight authentication protocol secure against some attacks. In *Second International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU 2006), 29 June 2006, Lyon, France*, pages 28–33. IEEE Computer Society, 2006. 36

[BCK96] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying hash functions for message authentication. In Neal Koblitz, editor, *Advances in Cryptology - CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, volume 1109 of *Lecture Notes in Computer Science*, pages 1–15. Springer, 1996. 16, 100

[BD08a] Steve Babbage and Matthew Dodd. The MICKEY stream ciphers. In Matthew J. B. Robshaw and Olivier Billet, editors, *New Stream Cipher Designs - The eSTREAM Finalists*, volume 4986 of *Lecture Notes in Computer Science*, pages 191–209. Springer, 2008. 4

[BD08b] James Birkett and Alexander W. Dent. Relations among notions of plaintext awareness. In Ronald Cramer, editor, *Public Key Cryptography - PKC 2008, 11th International Workshop on Practice and Theory in Public-Key Cryptography, Barcelona, Spain, March 9-12, 2008. Proceedings*, volume 4939 of *Lecture Notes in Computer Science*, pages 47–64. Springer, 2008. 151

[BDJR97] Mihir Bellare, Anand Desai, E. Jokipii, and Phillip Rogaway. A concrete security treatment of symmetric encryption. In *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997, Proceedings*, pages 394–403, 1997. 15

[BDPR98] 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, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings*, volume 1462 of *Lecture Notes in Computer Science*, pages 26–45. Springer, 1998. 151

[Ben] Boycott benetton no RFID tracking chips in clothing! http://www.boycottbenetton.com/. 7

[BHK+99] John Black, Shai Halevi, Hugo Krawczyk, Ted Krovetz, and Phillip Rogaway. UMAC: Fast and secure message authentication. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 216–233. Springer, 1999. 16

[Bir10] James Birkett. *On Plaintext-Aware Public-Key Encryption Schemes*. PhD thesis, Royal Holloway, University of London, 2010. Available at http://www.ma.rhul.ac.uk/static/techrep/2010/RHUL-MA-2010-11.pdf. 150, 152, 153, 155

[BKL+07] Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: An ultra-lightweight block cipher. In Pascal Paillier and Ingrid Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007, 9th International Workshop, Vienna, Austria, September 10-13, 2007, Proceedings*, volume 4727 of *Lecture Notes in Computer Science*, pages 450–466. Springer, 2007. 4

[BKL+11] Andrey Bogdanov, Miroslav Knezevic, Gregor Leander, Deniz Toz, Kerem Varici, and Ingrid Verbauwhede. SPONGENT: A lightweight hash function. In Bart Preneel and Tsuyoshi Takagi, editors, *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, volume 6917 of *Lecture Notes in Computer Science*, pages 312–325. Springer, 2011. 4

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM*, 50(4):506–519, 2003. 30

[BMT78] Elwyn R. Berlekampa, Robert J. McEliece, and Andehnk C. A. Van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24, 1978. 29

[Bon01] Dan Boneh. Simplified OAEP for the RSA and Rabin functions. In Joe Kilian, editor, *Advances in Cryptology - CRYPTO 2001, 21st Annual International Cryptology Conference, Santa Barbara, California, USA, August 19-23,* *2001, Proceedings*, volume 2139 of *Lecture Notes in Computer Science*, pages 275–291. Springer, 2001. 20

[BP04a] Mihir Bellare and Adriana Palacio. The knowledge-of-exponent assumptions and 3-round zero-knowledge protocols. In Matthew K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, 24th Annual International Cryptology-Conference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, pages 273–289. Springer, 2004. 150

[BP04b] Mihir Bellare and Adriana Palacio. Towards plaintext-aware public-key encryption without random oracles. In Pil Joong Lee, editor, *Advances in Cryptology - ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5-9, 2004, Proceedings*, volume 3329 of *Lecture Notes in Computer Science*, pages 48–62. Springer, 2004. 151

[BPR00] Mihir Bellare, David Pointcheval, and Phillip Rogaway. Authenticated key exchange secure against dictionary attacks. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000, International Conference on the Theory and Application of Cryptographic Techniques, Bruges, Belgium, May 14-18, 2000, Proceeding*, volume 1807 of *Lecture Notes in Computer Science*, pages 139–155. Springer, 2000. 88

[BR93] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. In *CCS 1993, Proceedings of the 1st ACM Conference on Computer and Communications Security, November 3-5, 1993, Fairfax, Virginia, USA*, pages 62–73, 1993. 21

[BR95a] Mihir Bellare and Phillip Rogaway. Optimal asymmetric encryption. In Alfredo De Santis, editor, *Advances in Cryptology - EUROCRYPT 1994, Workshop on the Theory and Application of Cryptographic Techniques, Perugia, Italy, May 9-12, 1994, Proceedings*, volume 950 of *Lecture Notes in Computer Science*, pages 92–111. Springer, 1995. 20, 151

[BR95b] Mihir Bellare and Phillip Rogaway. Provably secure session key distribution: the three party case. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing, 29 May-1 June 1995, Las Vegas, Nevada, USA*, pages 57–66. ACM, 1995.

[BR02] John Black and Phillip Rogaway. A block-cipher mode of operation for parallelizable message authentication. In Lars R. Knudsen, editor, *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, volume 2332 of *Lecture Notes in Computer Science*, pages 384–397. Springer, 2002. 16

[BRW04] Mihir Bellare, Phillip Rogaway, and David Wagner. The EAX mode of operation. In Bimal K. Roy and Willi Meier, editors, *Fast Software Encryption, 11th International Workshop, FSE 2004, Delhi, India, February 5-7, 2004, Revised Papers*, volume 3017 of *Lecture Notes in Computer Science*, pages 389–407. Springer, 2004. 16

[BY03] Mihir Bellare and Bennet S. Yee. Forward-security in private-key cryptography. In Marc Joye, editor, *Topics in Cryptology - CT-RSA 2003, The Cryptographers' Track at the RSA Conference 2003, San Francisco, CA, USA, April 13-17, 2003, Proceedings*, volume 2612 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2003.

[Can00] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. Cryptology ePrint Archive, Report 2000/067, 2000. http://eprint.iacr.org/. 90, 108

[CAS] Consumers against supermarket privacy invasion and numbering (CASPIAN). Anti-RFID Campaign webpage available at http://www.spychips.com/. 7

[CCGS06] Benoît Calmels, Sébastien Canard, Marc Girault, and Hervé Sibert. Low-cost cryptography for privacy in RFID systems. In Josep Domingo-Ferrer, Joachim Posegga, and Daniel Schreckling, editors, *Smart Card Research and Advanced Applications, 7th IFIP WG 8.8/11.2 International Conference, CARDIS 2006, Tarragona, Spain, April 19-21, 2006, Proceedings*, volume 3928 of *Lecture Notes in Computer Science*, pages 237–251. Springer, 2006. 4

[CD08] Ran Canetti and Ronny Ramzi Dakdouk. Extractable perfectly one-way functions. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations*, volume 5126 of *Lecture Notes in Computer Science*, pages 449–460. Springer, 2008. 148, 149, 150

[CDK09] Christophe De Cannière, Orr Dunkelman, and Miroslav Knezevic. KATAN and KTANTAN - a family of small and efficient hardware-oriented block ciphers. In Christophe Clavier and Kris Gaj, editors, *Cryptographic Hardware and Embedded Systems - CHES 2009, 11th International Workshop, Lausanne, Switzerland, September 6-9, 2009, Proceedings*, volume 5747 of *Lecture Notes in Computer Science*, pages 272–288. Springer, 2009. 4

[CGH98] Ran Canetti, Oded Goldreich, and Shai Halevi. The random oracle methodology, revisited (preliminary version). In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 209–218, 1998. 22

[CHH$^+$09] Seung Geol Choi, Javier Herranz, Dennis Hofheinz, Jung Yeon Hwang, Eike Kiltz, Dong Hoon Lee, and Moti Yung. The kurosawa-desmedt key encapsulation is not chosen-ciphertext secure. *Information Processing Letters*, 109(16):897–901, 2009. 153

[CHKP10] David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 523–552. Springer, 2010. 31

[CHS05] Ran Canetti, Shai Halevi, and Michael Steiner. Hardness amplification of weakly verifiable puzzles. In Joe Kilian, editor, *Theory of Cryptography, Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12, 2005, Proceedings*, volume 3378 of *Lecture Notes in Computer Science*, pages 17–33. Springer, 2005. 45, 46, 199

[CKS03] James M. Crawford, Michael J. Kearns, and Robert E. Schapire. The minimal disagreement parity problem as a hard satisfiability problem. Technical report, Computational Intelligence Research Laboratory, University of Oregon, 2003. 29

[Coo71] Stephen A. Cook. The complexity of theorem-proving procedures. In *Conference Record of Third Annual ACM Symposium on Theory of Computing, 1971, Shaker Heights, Ohio, USA*, pages 151–158. ACM, 1971. 91

[CP08] Christophe De Cannière and Bart Preneel. Trivium. In Matthew J. B. Robshaw and Olivier Billet, editors, *New Stream Cipher Designs - The eSTREAM Finalists*, volume 4986 of *Lecture Notes in Computer Science*, pages 244–266. Springer, 2008. 4

[CPMS07] Mauro Conti, Roberto Di Pietro, Luigi V. Mancini, and Angelo Spognardi. RIPP-FS: An RFID identification, privacy preserving protocol with forward secrecy. In *Fifth Annual IEEE International Conference on Pervasive Computing and Communications - Workshops (PerCom Workshops 2007), 19-23 March 2007, White Plains, New York, USA*, pages 229–234. IEEE Computer Society, 2007. 88, 105

[CS98] Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *Advances in Cryptology - CRYPTO 1998, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings*, volume 1462 of *Lecture Notes in Computer Science*, pages 13–25. Springer, 1998. 20, 152

[CS02] Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *Advances in Cryptology - EUROCRYPT 2002, International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 - May 2, 2002, Proceedings*, volume 2332 of *Lecture Notes in Computer Science*, pages 45–64. Springer, 2002. 153

[CS07] Claude Castelluccia and Mate Soos. Secret Shuffing: A Novel Approach to RFID Private Identification. In *Conference on RFID Security*, pages 169–180, Malaga, Spain, 2007. 7, 88, 91

[CS10] Baudoin Collard and François-Xavier Standaert. Multi-trail statistical saturation attacks. In Jianying Zhou and Moti Yung, editors, *Applied Cryptography and Network Security, 8th International Conference, ACNS 2010, Beijing, China, June 22-25, 2010. Proceedings*, volume 6123 of *Lecture Notes in Computer Science*, pages 123–138, 2010. 4

[CTIN08] Jose Carrijo, Rafael Tonicelli, Hideki Imai, and Anderson C. A. Nascimento. A novel probabilistic passive attack on the protocols HB and HB+. Cryptology ePrint Archive, Report 2008/231, 2008. 31

[CW77] Larry Carter and Mark N. Wegman. Universal classes of hash functions (extended abstract). In *Conference Record of the Ninth Annual ACM Symposium on Theory of Computing, 2-4 May 1977, Boulder, Colorado, USA*, pages 106–112. ACM, 1977. 17

[CW79] Larry Carter and Mark N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, 1979. 48

[CW90] Don Coppersmith and Shmuel Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9(3):251–280, 1990. 53

[Dam92] Ivan Damgård. Towards practical public key systems secure against chosen ciphertext attacks. In Joan Feigenbaum, editor, *Advances in Cryptology - CRYPTO 1991, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*, volume 576 of *Lecture Notes in Computer Science*, pages 445–456. Springer, 1992. 148, 150

[Den02] Alexander W. Dent. Adapting the weaknesses of the random oracle model to the generic group model. In Yuliang Zheng, editor, *Advances in Cryptology - ASIACRYPT 2002, 8th International Conference on the Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, December 1-5, 2002, Proceedings*, volume 2501 of *Lecture Notes in Computer Science*, pages 100–109. Springer, 2002. 150

[Den06a] Alexander W. Dent. The cramer-shoup encryption scheme is plaintext aware in the standard model. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June*

*1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 289–307. Springer, 2006. 151, 152

[Den06b] Alexander W. Dent. The hardness of the DHK problem in the generic group model. Cryptology ePrint Archive, Report 2006/156, 2006. http://eprint.iacr.org/. 150

[DH76] Whitfield Diffie and Martin Hellman. New directions in cryptography. *Information Theory, IEEE Transactions on Information Theory*, 22(6):644–654, 1976. 1

[DK07] Dang Nguyen Duc and Kwangjo Kim. Securing HB$^+$ against GRS man-in-the-middle attack. In *Institute of Electronics, Information and Communication Engineers, Symposium on Cryptography and Information Security, Jan. 23-26, 2007, Sasebo, Japan*, page 123, 2007. 37

[DLYZ10] 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, *Computer Security - ESORICS 2010, 15th European Symposium on Research in Computer Security, Athens, Greece, September 20-22, 2010. Proceedings*, volume 6345 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2010. 8, 114, 118, 122, 127, 139, 141

[DP08] Ivan Damgård and Michael Ostergaard Pedersen. RFID security: Tradeoffs between security and efficiency. In Tal Malkin, editor, *Topics in Cryptology - CT-RSA 2008, The Cryptographers' Track at the RSA Conference 2008, San Francisco, CA, USA, April 8-11, 2008. Proceedings*, volume 4964 of *Lecture Notes in Computer Science*, pages 318–332. Springer, 2008. 114, 116, 122, 138

[DR02] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer, 2002. 14

[DS09] Itai Dinur and Adi Shamir. Cube attacks on tweakable black box polynomials. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, volume 5479 of *Lecture Notes in Computer Science*, pages 278–299. Springer, 2009. 4

[DS11] Itai Dinur and Adi Shamir. Breaking grain-128 with dynamic cube attacks. In Antoine Joux, editor, *Fast Software Encryption - 18th International Workshop, FSE 2011, Lyngby, Denmark, February 13-16, 2011, Revised Selected Papers*, volume 6733 of *Lecture Notes in Computer Science*, pages 167–187. Springer, 2011. 4

[Elg85] Taher Elgamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and David Chaum, editors, *Advances* *in Cryptology, Proceedings of CRYPTO 1984, Santa Barbara, California, USA, August 19-22, 1984, Proceedings*, volume 196 of *Lecture Notes in Computer Science*, pages 10–18. Springer, 1985. 19

[FBV09] Junfeng Fan, Lejla Batina, and Ingrid Verbauwhede. Light-weight implementation options for curve-based cryptography: HECC is also ready for RFID. In *Proceedings of the 4th International Conference for Internet Technology and Secured Transactions, ICITST 2009, London, UK, November 9-12, 2009*, pages 1–6. IEEE, 2009. 4

[FDW04] Martin Feldhofer, Sandra Dominikus, and Johannes Wolkerstorfer. Strong authentication for RFID systems using the AES algorithm. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, volume 3156 of *Lecture Notes in Computer Science*, pages 357–370. Springer, 2004. 3, 135

[FLS90] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple non-interactive zero knowledge proofs based on a single random string (extended abstract). In *31st Annual Symposium on Foundations of Computer Science, 22-24 October 1990, St. Louis, Missouri, USA*, volume I, pages 308–317. IEEE, 1990. 149

[FR06] Martin Feldhofer and Christian Rechberger. A case against currently used hash functions in RFID protocols. In Robert Meersman, Zahir Tari, and Pilar Herrero, editors, *On the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops, OTM Confederated International Workshops and Posters, AWeSOMe, CAMS, COMINF, IS, KSinBIT, MIOS-CIAO, MONET, OnToContent, ORM, PerSys, OTM Academy Doctoral Consortium, RDDS, SWWS, and SeBGIS 2006, Montpellier, France, October 29 - November 3, 2006. Proceedings, Part I*, volume 4277 of *Lecture Notes in Computer Science*, pages 372–381. Springer, 2006. 3

[GCvDD03] Blaise Gassend, Dwaine E. Clarke, Marten van Dijk, and Srinivas Devadas. Delay-based circuit authentication and applications. In *Proceedings of the 2003 ACM Symposium on Applied Computing (SAC), March 9-12, 2003, Melbourne, FL, USA*, pages 294–301. ACM, 2003. 39

[GM82] Shafi Goldwasser and Silvio Micali. Probabilistic encryption and how to play mental poker keeping secret all partial information. In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing, 5-7 May 1982, San Francisco, California, USA*, pages 365–377. ACM, 1982. 19, 23

[GMR85] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof-systems (extended abstract). In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing, 6-8 May 1985, Providence, Rhode Island, USA*, pages 291–304. ACM, 1985. 118

[GMR89] Shafi Goldwasser, Silvio Micali, and Charles Rackoff. The knowledge complexity of interactive proof systems. *SIAM Journal on Computing*, 18(1):186–208, 1989. 118

[GMZZ08] Zbigniew Golebiewski, Krzysztof Majcher, Filip Zagorski, and Marcin Zawada. Practical attacks on HB and HB+ protocols. Cryptology ePrint Archive, Report 2008/241, 2008. 31

[Gol00] Oded Goldreich. *Foundations of Cryptography: Basic Tools*. Cambridge University Press, 2000.

[GPP11] Jian Guo, Thomas Peyrin, and Axel Poschmann. The PHOTON family of lightweight hash functions. In Phillip Rogaway, editor, *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, volume 6841 of *Lecture Notes in Computer Science*, pages 222–239. Springer, 2011. 4

[GPPR11] Jian Guo, Thomas Peyrin, Axel Poschmann, and Matthew J. B. Robshaw. The LED block cipher. In Bart Preneel and Tsuyoshi Takagi, editors, *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, volume 6917 of *Lecture Notes in Computer Science*, pages 326–341. Springer, 2011. 4

[GPS06] Marc Girault, Guillaume Poupard, and Jacques Stern. On the fly authentication and signature schemes based on groups of unknown order. *Journal of Cryptology*, 19(4):463–487, 2006. 4

[GPV08] Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 197–206. ACM, 2008. 31

[GRS05] Henri Gilbert, Matt Robshaw, and Hervé Sibert. Active attack against $HB^+$: a provably secure lightweight authentication protocol. *IEEE Electronics Letters*, 41(21):1169–1170, 2005. 5, 35

[GRS08a] Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin. Good variants of $HB^+$ are hard to find. In Gene Tsudik, editor, *Financial Cryptography and Data Security, 12th International Conference, Cozumel, Mexico, January 28-30, 2008. To appear*, Lecture Notes in Computer Science. Springer, 2008. 37, 38

[GRS08b] 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, 27th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Istanbul, Turkey, April 13-17, 2008. Proceedings*, volume 4965 of *Lecture Notes in Computer Science*, pages 361–378. Springer, 2008. 4, 5, 40, 44, 46, 48

[GRS08c] Henri Gilbert, Matthew J.B. Robshaw, and Yannick Seurin. HB$^{\#}$: Increasing the security and efficiency of HB$^{+}$, full version. Cryptology ePrint Archive, Report 2008/028, 2008. 48

[Gün90] Christoph G. Günther. An identity-based key-exchange protocol. In Jean-Jacques Quisquater and Joos Vandewalle, editors, *Advances in Cryptology - EUROCRYPT 1989, Workshop on the Theory and Application of of Cryptographic Techniques, Houthalen, Belgium, April 10-13, 1989, Proceedings*, volume 434 of *Lecture Notes in Computer Science*, pages 29–37. Springer, 1990.

[HAHH06] Panu Hämäläinen, Timo Alho, Marko Hännikäinen, and Timo D. Hämäläinen. Design and implementation of low-area and low-power AES encryption hardware core. In *Ninth Euromicro Conference on Digital System Design: Architectures, Methods and Tools (DSD 2006), 30 August - 1 September 2006, Dubrovnik, Croatia*, pages 577–583. IEEE Computer Society, 2006. 3

[Hås97] Johan Håstad. Some optimal inapproximability results. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 1–10. ACM, 1997. 29

[Hås01] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001. 29

[HB01] Nicholas J. Hopper and Manuel Blum. Secure human identification protocols. In Colin Boyd, editor, *Advances in Cryptology - ASIACRYPT 2001, 7th International Conference on the Theory and Application of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001, Proceedings*, volume 2248 of *Lecture Notes in Computer Science*, pages 52–66. Springer, 2001. 5, 32

[HFW11] Michael Hutter, Martin Feldhofer, and Johannes Wolkerstorfer. A cryptographic processor for low-resource devices: Canning ECDSA and AES like sardines. In Claudio Agostino Ardagna and Jianying Zhou, editors, *Information Security Theory and Practice. Security and Privacy of Mobile Devices in Wireless Communication - 5th IFIP WG 11.2 International Workshop, WISTP 2011, Heraklion, Crete, Greece, June 1-3, 2011. Proceedings*, volume 6633 of *Lecture Notes in Computer Science*, pages 144–159. Springer, 2011. 4

[HJMM08] Martin Hell, Thomas Johansson, Alexander Maximov, and Willi Meier. The Grain family of stream ciphers. In Matthew J. B. Robshaw and Olivier Billet, editors, *New Stream Cipher Designs - The eSTREAM Finalists*, volume 4986 of *Lecture Notes in Computer Science*, pages 179–190. Springer, 2008. 4, 6, 71

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In Joe Buhler, editor, *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer, 1998. 4

[HPVP11] Jens Hermans, Andreas Pashalidis, Frederik Vercauteren, and Bart Preneel. A new RFID privacy model. In Vijay Atluri and Claudia Diaz, editors, *Computer Security - ESORICS 2011, 16th European Symposium on Research in Computer Security, Leuven, Belgium, September 12-14, 2011. Proceedings*, Lecture Notes in Computer Science, page To Appear. Springer, 2011. 122, 141

[HS08] 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, *Applied Cryptography and Network Security, 6th International Conference, ACNS 2008, New York, NY, USA, June 3-6, 2008. Proceedings*, volume 5037 of *Lecture Notes in Computer Science*, pages 346–365, 2008. 38

[HSH+08] J. Alex Halderman, Seth D. Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph A. Calandrino, Ariel J. Feldman, Jacob Appelbaum, and Edward W. Felten. Lest we remember: Cold boot attacks on encryption keys. In Paul C. van Oorschot, editor, *Proceedings of the 17th USENIX Security Symposium, July 28-August 1, 2008, San Jose, CA, USA*, pages 45–60. USENIX Association, 2008. 125

[HWF09] Daniel M. Hein, Johannes Wolkerstorfer, and Norbert Felber. ECC is ready for RFID - a proof in silicon. In Roberto Maria Avanzi, Liam Keliher, and Francesco Sica, editors, *Selected Areas in Cryptography, 15th International Workshop, SAC 2008, Sackville, New Brunswick, Canada, August 14-15, Revised Selected Papers*, volume 5381 of *Lecture Notes in Computer Science*, pages 401–413. Springer, 2009. 4

[IK03] Tetsu Iwata and Kaoru Kurosawa. OMAC: One-key CBC MAC. In Thomas Johansson, editor, *Fast Software Encryption, 10th International Workshop, FSE 2003, Lund, Sweden, February 24-26, 2003, Revised Papers*, volume 2887 of *Lecture Notes in Computer Science*, pages 129–153. Springer, 2003. 16

[IKOS10] Yuval Ishai, Abishek Kumarasubramanian, Claudio Orlandi, and Amit Sahai. On invertible sampling and adaptive security. In Masayuki Abe, editor, *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, volume 6477 of *Lecture Notes in Computer Science*, pages 466–482. Springer, 2010. 148, 149

[ILL89] Russell Impagliazzo, Leonid A. Levin, and Michael Luby. Pseudo-random generation from one-way functions (extended abstracts). In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing, 15-17 May 1989, Seattle, Washington, USA*, pages 12–24. ACM, 1989. 17

[Jav08] Java card platform specification 2.2.2. Available online at http://java.sun.com/javacard/3.0/specs.jsp, 2008. 1

[JV04] Pascal Junod and Serge Vaudenay. FOX : A new family of block ciphers. In Helena Handschuh and M. Anwar Hasan, editors, *Selected Areas in Cryptography, 11th International Workshop, SAC 2004, Waterloo, Canada, August 9-10, 2004, Revised Selected Papers*, volume 3357 of *Lecture Notes in Computer Science*, pages 114–129. Springer, 2004. 14

[JW05a] Ari Juels and Stephen A. Weis. Authenticating pervasive devices with human protocols. In Victor Shoup, editor, *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*, pages 293–308. Springer, 2005. 3, 5, 34, 35

[JW05b] Ari Juels and Stephen A. Weis. Authenticating pervasive devices with human protocols (full version). Available online at http://saweis.net/pdfs/lpn-paper.pdf, 2005. 36

[JW07] Ari Juels and Stephen A. Weis. Defining strong privacy for rfid. In *Fifth Annual IEEE International Conference on Pervasive Computing and Communications - Workshops (PerCom Workshops 2007), 19-23 March 2007, White Plains, New York, USA*, pages 342–347. IEEE Computer Society, 2007. 8, 88, 97, 101, 104, 114, 119, 122, 137, 138, 141

[JW09] Ari Juels and Stephen A. Weis. Defining strong privacy for RFID. *ACM Transactions on Information and System Security*, 13(1), 2009. 114

[JW10] Shaoquan Jiang and Huaxiong Wang. Plaintext-awareness of hybrid encryption. In Josef Pieprzyk, editor, *Topics in Cryptology - CT-RSA 2010, The Cryptographers' Track at the RSA Conference 2010, San Francisco, CA, USA, March 1-5, 2010. Proceedings*, volume 5985 of *Lecture Notes in Computer Science*, pages 57–72. Springer, 2010. 153

[Kar72] Richard M. Karp. Reducibility among combinatorial problems. In Raymond E. Miller and James W. Thatcher, editors, *Proceedings of a symposium on the Complexity of Computer Computations, held March 20-22, 1972, at the IBM Thomas J. Watson Research Center, Yorktown Heights, New York*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972. 29

[KCS11] Stéphanie Kerckhof, Baudoin Collard, and François-Xavier Standaert. FP-GA implementation of a statistical saturation attack against PRESENT. In Abderrahmane Nitaj and David Pointcheval, editors, *Progress in Cryptology - AFRICACRYPT 2011 - 4th International Conference on Cryptology in Africa, Dakar, Senegal, July 5-7, 2011. Proceedings*, volume 6737 of *Lecture Notes in Computer Science*, pages 100–116. Springer, 2011. 4

[KD04] Kaoru Kurosawa and Yvo Desmedt. A new paradigm of hybrid encryption scheme. In Matthew K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, Cali-*

*fornia, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, pages 426–442. Springer, 2004. 153

[Kea98] Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, 1998. 30

[KL07] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography*. Cryptography and Network Security Series. Chapman & Hall/CRC, 2007.

[KLPR10] Lars R. Knudsen, Gregor Leander, Axel Poschmann, and Matthew J. B. Robshaw. PRINTcipher: A block cipher for ic-printing. In Stefan Mangard and François-Xavier Standaert, editors, *Cryptographic Hardware and Embedded Systems, CHES 2010, 12th International Workshop, Santa Barbara, CA, USA, August 17-20, 2010. Proceedings*, volume 6225 of *Lecture Notes in Computer Science*, pages 16–32. Springer, 2010. 4

[KPC$^+$11] Eike Kiltz, Krzysztof Pietrzak, David Cash, Abhishek Jain, and Daniele Venturi. Efficient authentication from hard learning problems. In Kenneth G. Paterson, editor, *Advances in Cryptology - EUROCRYPT 2011 - 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, Estonia, May 15-19, 2011. Proceedings*, volume 6632 of *Lecture Notes in Computer Science*, pages 7–26. Springer, 2011. 31, 65, 94

[Kra05] Hugo Krawczyk. HMQV: A high-performance secure diffie-hellman protocol. Cryptology ePrint Archive, Report 2005/176, 2005. http://eprint.iacr.org/.

[KS99] Aviad Kipnis and Adi Shamir. Cryptanalysis of the HFE public key cryptosystem by relinearization. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO 1999, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 19–30. Springer, 1999. 72

[KS06a] 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, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 73–87. Springer, 2006. 34, 35

[KS06b] Jonathan Katz and Adam Smith. Analyzing the HB and HB+ protocols in the "large error" case. Cryptology ePrint Archive, Report 2006/326, 2006. 35

[KYK06] Soo-Cheol Kim, Sang-Soo Yeo, and Sung Kwon Kim. MARP: Mobile agent for RFID privacy protection. In Josep Domingo-Ferrer, Joachim Posegga, and Daniel Schreckling, editors, *Smart Card Research and Advanced Applications, 7th IFIP WG 8.8/11.2 International Conference, CARDIS 2006, Tar-*

*ragona, Spain, April 19-21, 2006, Proceedings*, volume 3928 of *Lecture Notes in Computer Science*, pages 300–312. Springer, 2006. 7, 88, 94

[LAAZ11] Gregor Leander, Mohamed Ahmed Abdelraheem, Hoda AlKhzaimi, and Erik Zenner. A cryptanalysis of PRINTcipher: The invariant subspace attack. In Phillip Rogaway, editor, *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, volume 6841 of *Lecture Notes in Computer Science*, pages 206–221. Springer, 2011. 4

[LBdM06] Tri Van Le, Mike Burmester, and Breno de Medeiros. Provably secure ubiquitous systems: Universally composable RFID authentication protocols. In *Securecomm Workshops, 2006*, pages 1–9, 2006. 7, 88, 99, 102, 108, 110

[LBdM07] Tri Van Le, Mike Burmester, and Breno de Medeiros. Universally composable and forward-secure rfid authentication and authenticated key exchange. In Feng Bao and Steven Miller, editors, *Proceedings of the 2007 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2007, Singapore, March 20-22, 2007*, pages 242–252. ACM, 2007. 8, 90

[Lev08] Eric Levieil. *Contributions à l'étude cryptographique de protocoles et de primitives à clé secrète.* PhD thesis, Université Paris 7, 2008. 30, 31

[LF06] Éric Levieil and Pierre-Alain Fouque. An improved LPN algorithm. In Roberto De Prisco and Moti Yung, editors, *Security and Cryptography for Networks, 5th International Conference, SCN 2006, Maiori, Italy, September 6-8, 2006, Proceedings*, volume 4116 of *Lecture Notes in Computer Science*, pages 348–359. Springer, 2006. 31

[LK06] Chae Hoon Lim and Taekyoung Kwon. Strong and robust RFID authentication enabling perfect ownership transfer. In Peng Ning, Sihan Qing, and Ninghui Li, editors, *Information and Communications Security, 8th International Conference, ICICS 2006, Raleigh, NC, USA, December 4-7, 2006, Proceedings*, volume 4307 of *Lecture Notes in Computer Science*, pages 1–20. Springer, 2006. 7, 106, 108

[LM91] Xuejia Lai and James L. Massey. A proposal for a new block encryption standard. In Ivan Damgård, editor, *Advances in Cryptology - EUROCRYPT 1990, Workshop on the Theory and Application of of Cryptographic Techniques, Aarhus, Denmark, May 21-24, 1990, Proceedings*, volume 473 of *Lecture Notes in Computer Science*, pages 389–404. Springer, 1991. 14

[LP08] Sven Laur and Sylvain Pasini. SAS-based group authentication and key agreement protocols. In Ronald Cramer, editor, *Public Key Cryptography - PKC 2008, 11th International Workshop on Practice and Theory in Public-Key Cryptography, Barcelona, Spain, March 9-12, 2008. Proceedings*, volume 4939 of *Lecture Notes in Computer Science*, pages 197–213. Springer, 2008. 33

[LPPS07] Gregor Leander, Christof Paar, Axel Poschmann, and Kai Schramm. New lightweight DES variants. In Alex Biryukov, editor, *Fast Software Encryption, 14th International Workshop, FSE 2007, Luxembourg, Luxembourg, March 26-28, 2007*, volume 4593 of *Lecture Notes in Computer Science*, pages 196–210. Springer, 2007. 3

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In Henri Gilbert, editor, *Advances in Cryptology - EUROCRYPT 2010, 29th Annual International Conference on the Theory and Applications of Cryptographic Techniques, French Riviera, May 30 - June 3, 2010. Proceedings*, volume 6110 of *Lecture Notes in Computer Science*, pages 1–23. Springer, 2010. 31

[Lyu05] Vadim Lyubashevsky. The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem. In Chandra Chekuri, Klaus Jansen, José D. P. Rolim, and Luca Trevisan, editors, *Approximation, Randomization and Combinatorial Optimization, Algorithms and Techniques, 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2005 and 9th InternationalWorkshop on Randomization and Computation, RANDOM 2005, Berkeley, CA, USA, August 22-24, 2005, Proceedings*, volume 3624 of *Lecture Notes in Computer Science*, pages 378–389. Springer, 2005. 31

[MW04] 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, *Proceedings of the 11th ACM Conference on Computer and Communications Security, CCS 2004, Washington, DC, USA, October 25-29, 2004*, pages 210–219. ACM, 2004. 114, 116

[Nao03] Moni Naor. On cryptographic assumptions and challenges. In Dan Boneh, editor, *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings*, volume 2729 of *Lecture Notes in Computer Science*, pages 96–109. Springer, 2003. 150

[NIS99] NIST. FIPS publication 46-3: Data encryption standard (DES). Technical report, National Institute of Standards and Technology (NIST), 1999. 14

[NIS02] NIST. Fips publication 180-2: Secure hash standard. Technical report, National Institute of Standards and Technology (NIST), August 2002. 17

[NSMSN08] Ching Yu Ng, Willy Susilo, Yi Mu, and Reihaneh Safavi-Naini. RFID privacy models revisited. In Sushil Jajodia and Javier López, editors, *Computer Security - ESORICS 2008, 13th European Symposium on Research in Computer Security, Málaga, Spain, October 6-8, 2008. Proceedings*, volume 5283 of *Lecture Notes in Computer Science*, pages 251–266. Springer, 2008. 146, 147

[NY90] Moni Naor and Moti Yung. Public-key cryptosystems provably secure against chosen ciphertext attacks. In *Proceedings of the Twenty Second Annual ACM Symposium on Theory of Computing, 14-16 May 1990, Baltimore, Maryland, USA*, pages 427–437. ACM, 1990. 20

[oD] US Department of Defense. Military marking for shipment and storage. Available online at http://www.acq.osd.mil/log/rfid/MIL-STD-129PCH4.pdf. 2

[OOV08] 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, 14th International Conference on the Theory and Application of Cryptology and Information Security, Melbourne, Australia, December 7-11, 2008. Proceedings*, volume 5350 of *Lecture Notes in Computer Science*, pages 108–124. Springer, 2008. 5, 39, 42

[OP08a] Khaled Ouafi and Raphael C.-W. Phan. Privacy of recent RFID authentication protocols. In Liqun Chen, Yi Mu, and Willy Susilo, editors, *Information Security Practice and Experience, 4th International Conference, ISPEC 2008, Sydney, Australia, April 21-23, 2008, Proceedings*, volume 4991 of *Lecture Notes in Computer Science*, pages 263–277. Springer, 2008. 8, 88

[OP08b] 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, *Applied Cryptography and Network Security, 6th International Conference, ACNS 2008, New York, NY, USA, June 3-6, 2008. Proceedings*, volume 5037 of *Lecture Notes in Computer Science*, pages 479–489, 2008. 8, 88

[OPSW10] Khaled Ouafi, Raphael C.-W. Phan, Doug Stinson, and Jiang Wu. Privacy analysis of forward and backward untraceable RFID authentication schemes. *Wireless Personal Communications*, pages 1–13, 2010. 8

[oR63] Paul Erd os and Alfrèd Rényi. On two problems of information theory. *Publ. Math. Inst. Hung. Acad. Sci.*, 8(21):229–243, 1963. 56

[OSK05] Miyako Ohkubo, Koutarou Suzuki, and Shingo Kinoshita. RFID privacy issues and technical challenges. *Communications of the ACM*, 48(9):66–71, 2005. 94, 104

[OV09] Khaled Ouafi and Serge Vaudenay. Smashing SQUASH-0. In Antoine Joux, editor, *Advances in Cryptology - EUROCRYPT 2009, 28th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cologne, Germany, April 26-30, 2009. Proceedings*, volume 5479 of *Lecture Notes in Computer Science*, pages 300–312. Springer, 2009. 6, 68

[Pai99] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *Advances in Cryptology - EUROCRYPT*

*1999, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 1999. 19

[Pei09] Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In Michael Mitzenmacher, editor, *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 333–342. ACM, 2009. 31

[Pfi88] Birgit Pfitzmann. *Learning from Good and Bad Data*. The Springer International Series in Engineering and Computer Science. Springer, 1988. 30

[Pie10] Krzysztof Pietrzak. Subspace LWE. Manuscript available at http://homepages.cwi.nl/~pietrzak/publications/SLWE.pdf, 2010. 31

[PKC07] PKCS #11: Cryptographic token interface standard. Available online at http://www.rsa.com/rsalabs/node.asp?id=2133, 2007. 1

[PV06] Sylvain Pasini and Serge Vaudenay. SAS-based authenticated key agreement. In Moti Yung, Yevgeniy Dodis, Aggelos Kiayias, and Tal Malkin, editors, *Public Key Cryptography - PKC 2006, 9th International Conference on Theory and Practice of Public-Key Cryptography, New York, NY, USA, April 24-26, 2006, Proceedings*, volume 3958 of *Lecture Notes in Computer Science*, pages 395–409. Springer, 2006. 33

[PV08] Radu-Ioan Paise and Serge Vaudenay. Mutual authentication in RFID: security and privacy. In Masayuki Abe and Virgil D. Gligor, editors, *Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2008, Tokyo, Japan, March 18-20, 2008*, pages 292–299. ACM, 2008. 168

[PVW08] Chris Peikert, Vinod Vaikuntanathan, and Brent Waters. A framework for efficient and composable oblivious transfer. In David Wagner, editor, *Advances in Cryptology - CRYPTO 2008, 28th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2008. Proceedings*, volume 5157 of *Lecture Notes in Computer Science*, pages 554–571. Springer, 2008. 31

[PW08] Chris Peikert and Brent Waters. Lossy trapdoor functions and their applications. In Cynthia Dwork, editor, *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 187–196. ACM, 2008. 31

[PX09] Manoj Prabhakaran and Rui Xue. Statistically hiding sets. In Marc Fischlin, editor, *Topics in Cryptology - CT-RSA 2009, The Cryptographers' Track at the RSA Conference 2009, San Francisco, CA, USA, April 20-24, 2009. Proceedings*, volume 5473 of *Lecture Notes in Computer Science*, pages 100–116. Springer, 2009. 150

[Rab79] Michael O. Rabin. Digitalized signatures and public-key functions as intractable as factorization. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA, 1979. 5, 22

[Reg05] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 84–93. ACM, 2005. 30, 31

[RS92] 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, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*, volume 576 of *Lecture Notes in Computer Science*, pages 433–444. Springer, 1992. 20

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978. 22

[RSS$^+$10] Ulrich Rührmair, Frank Sehnke, Jan Sölter, Gideon Dror, Srinivas Devadas, and Jürgen Schmidhuber. Modeling attacks on physical unclonable functions. In Ehab Al-Shaer, Angelos D. Keromytis, and Vitaly Shmatikov, editors, *Proceedings of the 17th ACM Conference on Computer and Communications Security, CCS 2010, Chicago, Illinois, USA, October 4-8, 2010*, pages 237–249. ACM, 2010. 39

[Rud92] Steven Rudich. The use of interaction in public cryptosystems (extended abstract). In Joan Feigenbaum, editor, *Advances in Cryptology - CRYPTO 1991, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*, volume 576 of *Lecture Notes in Computer Science*, pages 242–251. Springer, 1992.

[Sch90a] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO 1989, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252. Springer, 1990. 1

[Sch90b] Claus-Peter Schnorr. Efficient identification and signatures for smart cards (abstract). In Jean-Jacques Quisquater and Joos Vandewalle, editors, *Advances in Cryptology - EUROCRYPT 1989, Workshop on the Theory and Application of of Cryptographic Techniques, Houthalen, Belgium, April 10-13, 1989, Proceedings*, volume 434 of *Lecture Notes in Computer Science*, pages 688–689. Springer, 1990. 1

[Seu09] Yannick Seurin. *Primitives et protocoles cryptographiques à sécurité prouvée.*

PhD thesis, Université de Versailles Saint-Quentin-en-Yvelines, 2009. Available online at http://yannickseurin.free.fr/pubs/these_Yannick_Seurin.pdf (In French). 39, 45, 48

[Sha49] Claude E. Shannon. Communication theory of secrecy systems. *Bell System Technical Journal*, 28(4):656–715, 1949. 14

[Sha84] Adi Shamir. A polynomial-time algorithm for breaking the basic merkle-hellman cryptosystem. *IEEE Transactions on Information Theory*, 30(5):699–704, 1984. 91

[Sha95] Adi Shamir. Memory efficient variants of public-key schemes for smart card applications. In Alfredo De Santis, editor, *Advances in Cryptology - EUROCRYPT 1994, Workshop on the Theory and Application of Cryptographic Techniques, Perugia, Italy, May 9-12, 1994, Proceedings*, volume 950 of *Lecture Notes in Computer Science*, pages 445–449. Springer, 1995. 68

[Sha07] Adi Shamir. SQUASH: A new one-way hash function with provable security properties for highly constrained devices such as RFID tags., 2007. Invited lecture to the RFID Security'07 Workshop. Slides available from http://mailman.few.vu.nl/pipermail/rfidsecuritylist/2007-August/000001.html. 4, 5, 6, 68

[Sha08] Adi Shamir. SQUASH - a new MAC with provable security properties for highly constrained devices such as RFID tags. In Kaisa Nyberg, editor, *Fast Software Encryption, 15th International Workshop, FSE 2008, Lausanne, Switzerland, February 10-13, 2008, Revised Selected Papers*, volume 5086 of *Lecture Notes in Computer Science*, pages 144–157. Springer, 2008. 4, 5, 6, 71

[Sho04] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Report 2004/332, 2004. http://eprint.iacr.org/. 24

[Str69] Volker Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 14(3):354–356, 1969. 53

[TSL07] Chiu Chiang Tan, Bo Sheng, and Qun Li. Severless search and authentication protocols for RFID. In *Fifth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom 2007), 19-23 March 2007, White Plains, New York, USA*, pages 3–12. IEEE Computer Society, 2007. 7, 88, 97

[Tsu06] Gene Tsudik. YA-TRAP: Yet another trivial RFID authentication protocol. In *4th IEEE Conference on Pervasive Computing and Communications Workshops (PerCom 2006 Workshops), 13-17 March 2006, Pisa, Italy*, pages 640–643. IEEE Computer Society, 2006. 7, 99

[Vau98] Serge Vaudenay. Cryptanalysis of the chor-rivest cryptosystem. In Hugo Krawczyk, editor, *Advances in Cryptology - CRYPTO '98, 18th Annual International Cryptology Conference, Santa Barbara, California, USA, August 23-27, 1998, Proceedings*, volume 1462 of *Lecture Notes in Computer Science*, pages 243–256. Springer, 1998. 91

[Vau03] Serge Vaudenay. Decorrelation: A theory for block cipher security. *Journal of Cryptology*, 16(4):249–286, 2003. 15

[Vau05a] Serge Vaudenay. *A classical introduction to cryptography - applications for communications security*. Springer, 2005.

[Vau05b] Serge Vaudenay. Secure communications over insecure channels based on short authenticated strings. In Victor Shoup, editor, *Advances in Cryptology - CRYPTO 2005: 25th Annual International Cryptology Conference, Santa Barbara, California, USA, August 14-18, 2005, Proceedings*, volume 3621 of *Lecture Notes in Computer Science*, pages 309–326. Springer, 2005. 33

[Vau06] Serge Vaudenay. RFID privacy based on public-key cryptography. In Min Surp Rhee and Byoungcheon Lee, editors, *Information Security and Cryptology - ICISC 2006, 9th International Conference, Busan, Korea, November 30 - December 1, 2006, Proceedings*, volume 4296 of *Lecture Notes in Computer Science*, pages 1–6. Springer, 2006. 90

[Vau07] Serge Vaudenay. On privacy of RFID. In Kaoru Kurosawa, editor, *To Appear in Advances in Cryptology - ASIACRYPT 2007, 13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, December 2-6, 2007, Proceedings*, Lecture Notes in Computer Science. Springer, 2007. 8, 90, 102, 122, 123, 135, 136, 158

# CURRICULUM VITÆ

Khaled Ouafi was born in Paris, France, in 1984. After attending primary school and part of secondary school in Paris, he moved to Bordj Bou Arreridj, Algeria, in 1996 where he finished his secondary school in Arabic. Five years later, he obtained a scientific baccalaureate with a major in Math and Physics (*"Baccalauréat de l'enseignement supérieur filière sciences exactes"*) with mention *"Très Bien"* and ranked 1st at the regional level and 56th at the national level. This achievement qualified him for an excellence scholarship for studies abroad awarded by the President of Algeria and the Ministry of Higher Education and Scientific Research.

In consequence, he moved to Switzerland and joined EPFL where he was enrolled in the preparatory year, Cours de Mathématiques Spéciales. He then spent one year studying communication systems before switching to computer science in which he successively obtained a Bachelor and Master in Computer Science in 2005 and 2007. His Master thesis was supervised by Prof. Serge Vaudenay and entitled *"Fail-StopSignatures and their Applications"*. In parallel to his studies, he has worked as an independent web designer and was part of a team responsible for organizing a stand in the expo *"Le Comptoir Suisse 2007"*. He was also an active member of several student associations such as Unipoly, Junior Entreprise, and Forum EPFL.

Since 2007, he is a PhD student and full-time research assistant at EPFL's Laboratory of Security and Cryptography (LASEC) led by Prof. Serge Vaudenay where, in parallel to working on completing his PhD thesis, he is involved in other activities, including teaching and IT system administration.