SECURITY AND PRIVACY IN RFID SYSTEMS

by

SÜLEYMAN KARDAŞ

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Sabancı University

June, 2014

SECURITY AND PRIVACY IN RFID SYSTEMS

APPROVED BY:

Assoc. Prof. Dr. Albert Levi (Thesis Supervisor)	.....
Assoc. Prof. Dr. Erkay Savaş	.....
Assoc. Prof. Dr. Cem Güneri	.....
Assist. Prof. Dr. Cemal Yilmaz	.....
Prof. Dr. Gildas Avoine (INSA Rennes France & UCL Belgium)	.....

DATE OF APPROVAL: .....

To the my sweet baby Esra and my wife Leyla...

SECURITY AND PRIVACY IN RFID SYSTEMS

Süleyman Kardaş
Computer Science and Engineering
Ph.D. Thesis, 2014
Thesis Supervisor: Assoc. Prof. Dr. Albert Levi

Keywords: RFID, Security, Privacy, Distance Bounding Problem, Physically Unclonable Functions

Abstract

RFID is a leading technology that has been rapidly deployed in several daily life applications such as payment, access control, ticketing, e-passport, supply-chain, etc. An RFID tag is an electronic label that can be attached to an object/individual in order to identify or track the object/individual through radio waves. Security and privacy are two major concerns in several applications as the tags are required to provide a proof of identity. The RFID tags are generally not tamper-resistant against strong adversarial attacks. They also have limited computational resources. Therefore, the design of a privacy preserving and cost-effective RFID authentication protocol is a very challenging task for industrial applications. Moreover, RFID systems are also vulnerable to relay attacks (i.e., mafia, terrorist and distance frauds) when they are used for authentication purposes. Distance bounding protocols are particularly designed as a countermeasure against these attacks. These protocols aim to ensure that the tags are in a bounded area by measuring the round-trip delays during a rapid challenge-response exchange of short authentication messages. Several RFID distance bounding protocols have been proposed recently in the literature. However, none of them provides

the ideal security against the terrorist fraud. Besides, the requirements of low resources and inefficient data management trigger to make use of cloud computing technology in RFID authentication systems. However, as more and more information on individuals and companies is placed in the cloud, concerns about data safety and privacy raise. Therefore, while integrating cloud services into RFID authentication systems, the privacy of tag owner against the cloud must also be taken into account.

Motivated by this need, this dissertation contributes to the design of algorithms and protocols aimed at dealing with the issues explained above. First of all, we introduce two privacy models for RFID authentication protocols based on Physically Unclonable Functions (PUF). We propose several authentication protocols in order to demonstrate these models. Moreover, we study distance bounding protocols having bit-wise fast phases and no final signature. We give analysis for the optimal security limits of the distance bounding protocols. Furthermore, we propose a novel RFID distance bounding protocol based on PUFs and it satisfies the highest security levels. Finally, we provide a new security and privacy model for integrating cloud computing into RFID systems. For the sake of demonstration of this model, we also propose two RFID authentication protocols that require various computational resources and provide different privacy levels.

RFID SİSTEMLERİNDE GÜVENLİK VE MAHREMİYET

Süleyman Kardaş
Bilgisayar Bilimi ve Mühendisliği
Doktora Tezi, 2014
Tez Danışmanı: Doçent Dr. Albert Levi

Anahtar Sözcükler: RFID, Güvenlik, Mahremiyet, Mesafe Sınırlama Problemi, Fiziksel Klonlanamayan Fonksiyonlar

Özet

Radyo Frekanslı Kimlik Tanımlama (RFID) teknolojisi, son zamanlarda günlük hayatımızdaki bir çok uygulamalarda kullanılmaktadır. Özellikle pasaportlarda, ödeme sistemlerinde, giriş/çıkış kontrollerinde, tedarik zincirinde vb. uygulamalarda kullanılmaktadır. RFID etiketleri nesne veya canlılar üzerinde yerleştirilen bir çip olup radyo frekansı aracılığı ile kimlik tanımlamaya ve takip edilmeye olanak sağlar. Kimlik doğrulama gerektiren uygulamalarda güvenlik ve mahremiyet iki önemli sorundur. Öte yandan, RFID etiketleri güçlü fiziksel saldırılara karşı dayanıklı değildirler ve sınırlı hesaplama kaynaklarına sahiptirler. Bu nedenle, endüstriyel uygulamalar için mahremiyet odaklı, güvenli ve maliyet etkin bir doğrulama mekanizması tasarlarmak çok zor bir iştir. Ayrıca, RFID sistemleri kimlik doğrulama amaçlı kullanıldığında aktarma saldırılarına (yani mafya, terörist ve dolandırıcılık saldırıları) açıktır. Mesafe sınırlama protokolleri özellikle bu saldırılara karşı bir önlem olarak tasarlanmıştır. Bu protokollerde, etiketler ile okuyucu arasında hızlı bir sorgu/cevap işleminde mesajların gidiş-dönüş gecikme süreleri ölçülerek etiketlerin dar ve sınırlı bir alan içerisinde kimlik doğrulama yapmaları hedeflenmektedir. Son zamanlarda, literatürde bir çok

RFID mesafe sınırlayıcı protokolleri sunuldu, ancak bunların hiçbiri terörist dolandırıcılığa karşı ideal bir güvenlik çözümü sunmamaktadır.

Öte yandan, okuyucu ve sunucu tarafında kaynakların yetersiz olması durumunda güvenli ve verimli bir kimlik doğrulama protokolünü tasarımı inşa etmek zorlaşmaktadır. Bulut bilişim bu soruna etkili bir çözüm sağlamak için umut verici bir teknoloji olarak karşımıza çıkmaktadır. Bulut bilişimde birey ve şirketler hakkında belge ve dokümanların sayısı arttıkça ve bu bilgilerin bulut bilişimde korunması gerekliliği endişelerini arttırmaktadır. RFID kimlik doğrulama sistemleri içine bulut hizmetlerini entegre ederken, bulut bilişime karşı RFID etiket sahibinin mahremiyetinin korunması da dikkate alınmalıdır .

Bu motivasyonla, bu doktora tezi, yukarıda belirtilen problemlere çözüm olmak amacı ile güvenli ve mahremiyet odaklı RFID protokollerin tasarımlarına katkıda bulunmaktadır. Öncelikle, Klonlanamayan fonksiyonlara (PUF) dayalı iki farklı RFID mahremiyet modeli önerildi. Modellerin uygulanabilirliği için çeşitli kimlik doğrulama protokolleri önerildi. Ayrıca, mesafe sınırlama protokolleri üzerinde katkılar yapıldı. PUF fonksiyonlar kullanılarak yeni bir RFID mesafe sınırlayıcı protokolü önerildi ve bu protokol ile en yüksek güvenlik seviyelerinin nasıl sağlandığı gösterildi. Son olarak, RFID sistemleri içine bulut bilişim teknolojilerinin entegre edilmesi için yeni bir güvenlik ve mahremiyet modeli tanımlandı ve bu modelin pratikte uygulanabilir olduğunu göstermek için iki farklı protokol önerildi.

ACKNOWLEDGMENTS

I am deeply grateful to my supervisor Prof. Dr. Albert Levi, who has guided me with his invaluable suggestions and criticisms, and encouraged me a lot in my academic life. It was a great pleasure for me to have a chance of working with him.

I would like to sincerely thank to my doctoral committee Erkay Savaş, Cem Güneri, Cemal Yılmaz and Gildas Avoine for their invaluable time and suggestions. I am also thankful to my friends, especially Muhammed Ali Bingöl, Serkan Celik and Mehmet Sabir Kiraz for their always being supportive and accountable. I also want to thank Atakan Arslan, Ertuğrul Murat, Ziya Alper Genç and all my colleagues in TUBITAK BILGEM UEKAE for their support and strong friendship.

Last but not least, I am deeply indebted to my family for their unagging care, trust and support. My father's commitment to lifelong learning, growth and hard work has nurtured and inspired me throughout my life. My mother's endless love and faith in me have always been a beacon of confidence for me. Most of all, my lovely wife deserves special acknowledgment. Her perseverance, devotion and sacrifices enabled me and motivated me to focus on the research I am interested in. I am eternally grateful and wonderfully blessed to have her as my wife.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

INTRODUCTION

This chapter firstly presents the motivations for the challenges that are faced in RFID systems. Then, the structure and organization of the dissertation are outlined. Finally, it briefly discusses our contributions for handling these challenges.

1.1 Motivations

Radio Frequency IDentification (RFID) technology has received increasing attention as an emerging solution for remotely identifying and/or authenticating objects or individuals with the help of RFID tags. A typical RFID system generally consists of tags, i.e., a microcircuit with an antenna, readers, which allow to remotely query the tags, and a back-end server that manages all the information related to each tag. In simplest terms, the working principle of an RFID system is that a tag transfers its coded data when queried by a reader. The reader conveys the packets collected from the tag to back-end server in order to perform the identification and/or authentication process.

Recently, RFID technology has been rapidly deployed in several daily-

life applications such as payment, access control, ticketing, e-passport, etc. The communication between tags and readers runs on an insecure wireless channel. The security and privacy are definitely two critical concerns in those applications since the tags are generally required to provide a proof of identity in most applications. The most conspicuous privacy risk is *tracking* of the tag owner. In this case, the tracker can obtain and abuse tag owners' profile. Therefore, an RFID system should provide confidentiality of the tag identity along with privacy of the tag owner.

Mitigating these problems requires researchers to design identification and authentication protocols that include cryptographic mechanisms. On the other hand, most of RFID tags have limited memory and computational capability; therefore, the existing privacy-preserving mechanisms, which require high computational costs, are not applicable to many restricted RFID systems. Furthermore, most of RFID tags are not tamper resistant against strong adversarial attacks. Namely, physical attacks on tag's chip allow the adversary to learn the secrets stored in the tag. Thus, the design of a privacy preserving and cost-effective RFID authentication protocol is a challenging task. To fulfill these needs, several authentication mechanisms have been proposed in the literature [1–17].

Moreover, having a security and privacy model for RFID systems is essential for making formal security analysis of RFID authentication protocols. A large number of frameworks have been proposed to formalize security and privacy in the context of RFID system [18–27]. The shortcomings of these frameworks are addressed in [28].

Furthermore, typical RFID systems are also vulnerable to relay attacks when they are used for authentication purposes. Distance bounding protocols are particularly designed as a countermeasure against relay attacks.

These protocols aim to ensure that the tags are in a bounded area by measuring the round-trip delays during a rapid challenge-response exchange of short authentication messages. Several RFID distance bounding protocols have been proposed in the literature. However, none of them provides ideal security against the terrorist fraud, who collaborates with the tag's owner.

On the one hand, in some applications multiple tag reading points may be required to track the products throughout the workplace. For scalability reasons, in some systems, multiple databases can be established which is costly and it is difficult to merge them in-house. Moreover, such systems may have synchronization and data consistency problems if managed poorly. Furthermore, in order to make use of the benefits of RFID, retailers will need to upgrade their IT infrastructure in a number of areas, and their interfaces with other businesses should be closer. Outsourcing background systems and database management to the cloud is a promising alternative to the these issues. However, the verification of tagged items by RFID systems provides full traceability from sender (e.g. manufacturer) to receiver by maintaining a single database placed in a cloud computing. This provides assurance that a product has been shipped and delivered. However, as more and more information on individuals and companies are placed in the cloud, safety and privacy of the cloud environment become an important issue. Therefore, the integration of cloud computing into RFID systems requires the privacy of the tag owner against the cloud to be taken into account.

1.2 Contributions

The main contributions of the dissertation are given as follows:

1. In Chapter 3, we give our first contributions to the RFID privacy-

preserved authentication protocols based on Physically Unclonable Functions. In this chapter, we study the common assumption of PUFs that their physical structure is destroyed once tampered. This assumption works only in the ideal case because the tamper-resistance depends on the ability of the attacker and the quality of the PUF circuits. We have weaken this assumption by introducing a new definition *k-resistant PUFs*. k-PUFs are tamper-resistant against at most k attacks, i.e., their physical structure remains still functional and correct until at most k^{th} physical attack. Furthermore, we prove that strong privacy can be achieved without public-key cryptography using k-PUF based authentication. We finally prove that our extended proposal achieves both reader authentication and k-strong privacy. The results presented in Chapter 3 have been accepted in [29].

2. In Chapter 4, we first revisit Vaudenay's model [18], extend it by considering offline RFID system and introduce the notion of compromised reader attacks. Then, we propose an efficient RFID mutual authentication protocol for offline RFID system. Our protocol is based on the use of PUFs. We prove that our protocol provides destructive privacy for tag owner even against reader attacks. The results presented in Chapter 4 have been published in [30].

3. In Chapter 5, we formally analyze a recent RFID authentication protocol [31] and proved that it provides destructive privacy according to Vaudenay privacy model [18]. Then, we propose a unilateral authentication protocol and prove that our protocol satisfies higher privacy level such as narrow strong privacy. Moreover, we provide an enhanced version of the protocol, which has the same privacy level as the protocol

of [31], but has also reader authentication against stronger adversaries. Furthermore, the enhanced version of our protocol uses smaller number of cryptographic operations when compared to [31]'s protocol. It is also cost efficient at the server and tag side and requires $\mathcal{O}(1)$ complexity to identify an RFID tag. The results presented in Chapter 5 have been published in [32, 33].

4. In Chapter 6, we introduce the notion of k-previous challenge dependent (k-PCD) distance bounding protocols, in which each response bit depends on the current and the k previous challenges. We then analyze k-PCD distance bounding protocols and show the success probabilities against mafia and distance fraud attacks. We present a simple approach to construct k-PCD protocols with only two registers. The results presented in Chapter 6 have been published in [34] and have been submitted to a Journal [35].

5. In Chapter 7, we first introduce a strong adversary model for PUF based authentication protocol in which the adversary has access to volatile memory of the tag. We show that the security of Sadeghi et al.'s PUF based authentication protocol is not secure according to this model. We provide a new technique to improve the security of their protocol. More specifically, in our scheme, even if an adversary has access to volatile memory, she cannot obtain all long term keys to clone the tag. Next, we propose a novel RFID distance bounding protocol based on PUFs, which satisfies the expected security requirements. Comparing to the previous protocols, the use of PUFs in our protocol enhances the system in terms of security, privacy and tag computational overhead. We also prove that our extended protocol with a final signature

provides ideal security against all those frauds, remarkably the terrorist fraud. Besides, our protocols enjoy the attractive properties of PUFs. The results presented in Chapter 7 was published in [1].

6. In Chapter 8, we first provide a new security and privacy model for RFID systems that utilize the cloud computing. In this context, we first define the capabilities of the adversary and give the privacy definitions. Then, we present two cloud-based RFID authentication protocols in order to illustrate our model. The first one is based on symmetric cryptography and the other one is based on elliptic-curve cryptography. According to our model, we prove that the former protocol achieves destructive privacy and the latter one provides narrow-strong privacy. The cloud is assumed to be honest-but-curious; therefore, tag related data are stored in an encrypted form in the cloud. In order for retrieving tag data without violating privacy of the tag owner, we also propose a private and efficient single keyword search scheme. We prove that our search scheme satisfies data, query and result pattern privacy. The results presented in Chapter 8 have been published in [36] and submitted to a Journal [37].

1.3 Thesis Outline

The organization of the dissertation is outlined as follows. Chapter 2 provides an overview of RFID systems and describes the security and privacy challenges that the RFID technology should address. It also gives the cryptographic background. Chapter 3 introduces privacy models for RFID authentication protocols based on the use of Physically Unclonable Functions (PUFs). Chapter 4 gives our contributions to the offline RFID system. Chap-

ter 5 introduces our proposed RFID authentication protocol and gives its formal security and privacy analysis. Chapter 6 explores k-previous challenge dependent (k-PCD) distance bounding protocols, in which each response bit depends on the current and the k previous challenges. Chapter 7 proposes a new PUF based RFID distance bounding protocol and shows the use of PUF enhancements. Chapter 8 presents our contributions to the RFID systems where cloud services are integrated. Finally, Chapter 9 summarizes our contributions.

Chapter 2

OVERVIEW OF RFID SYSTEMS

In this chapter, we first give a brief explanation of a typical RFID system. Then, we classify RFID systems into two models that differ in terms of connectivity of RFID readers to the back-end server. After that, we explain the security and privacy needs in RFID systems. Finally, the related work on RFID systems is given.

2.1 RFID Systems

Radio Frequency IDentification (RFID) technology is getting pervasively deployed in many daily life applications ranging from inventory management to anti-counterfeiting protection. A typical RFID system consists of three components that actively or passively interact with each other (see Figure 2.1).

The first component of the system is a group known as *tags* or *labels*. Most of the tags contain a tiny integrated microcircuit, of a few millimeters on the side, for storing and calculating information, modulating and demodulating

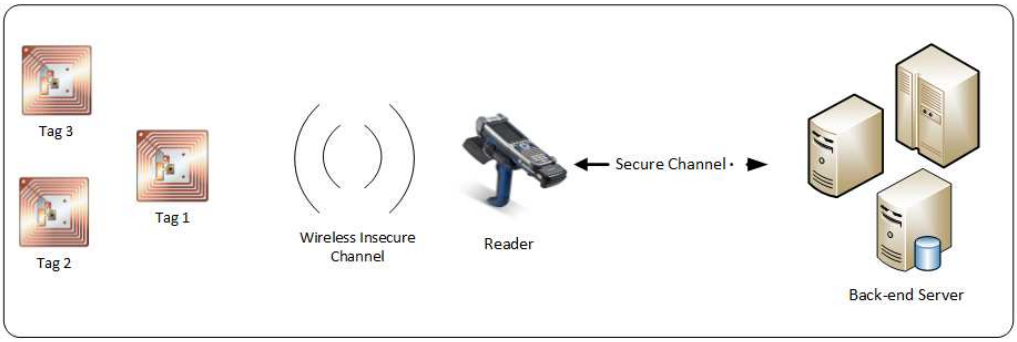

Figure 2.1: A typical RFID system

a radio-frequency (RF) signal and an antenna for receiving and transmitting the signal. There are three types of tags; (i) *passive*, (ii) *active* and (iii) *battery assisted passive* tags. The passive tags have no internal power source and need an external signal to be invoked. They are being energized and activated by radio waves from an outside source. They represent the most commonly used tag class in RFID applications. *Active tags* contain a power source (i.e. a battery) and can actively generate and send signal to a reader for communication. The last tag family (*battery assisted passive tags*) contain a low power source but these kinds of tags still need a wake up signal as passive tags do. They use the battery for only computation inside the chip. The wireless channel between a tag and a reader can use spectrum in the Low Frequency (LF) range (124 to 135 KHz), High Frequency (HF) range (13.56 MHz) or Ultra High Frequency (UHF) range (868, 915, 950 MHz). Thus, direct contact between a reader and a tag is not required. According to the frequency specification, some of them can be queried from several meters. The tagged object does not need to be in the line of sight, but earlier technologies such as the bar-code and smart cards do. This is a significant difference between RFID and the earlier technologies.

Some of the most popular daily life RFID applications are given as follows.

- Tracking persons and animals.
- Access control and management.
- Toll collection.
- Contact-less payment.
- Tracking of books in library.
- Machine readable travel documents.
- Tracking of goods in supply-chain management.

The second component is a group known as *readers* or *interrogators*. RFID readers are commonly composed of an RF module, a control unit, and a coupling element to interact with the tags by means of RF communication [38]. The readers consign the packets collected from the tags to the back-end server in order to perform the identification and/or authentication process. Readers have no physical and computational restriction and they can be mobile or fixed.

The last component is the *back-end system*, which can be centralized or distributed. It stores all tags' information and readers' information in its own database. It is also the synchronization point for all the other components and all initialization routines take place. Moreover, in RFID areas, the back-end system is generally assumed to be secure against all kind of attacks.

2.2 RFID Models

An RFID system can be classified into two models in terms of the communication between a back-end server and readers. First one is referred to as

central database model but throughout the dissertation it is called as *online model*. The latter is referred to *offline model*.

2.2.1 Online Model

In the online model, the back-end system contains all the tag-related information. The readers are assumed to be always connected to the back-end system. Although it is between the tags and the back-end system, the main duty of the reader is to query the tag and to return the response of the tags to the back-end system without knowing the content of the tag reply. It does not contain any tag specific information such as keys, IDs, counters, etc. A good example is a building access system where the users have their own cards to be used as keys in order to enter rooms or to access different facilities. The major shortcoming with the online model is that the readers must have a live secure connection to the database of the central server.

2.2.2 Offline Model

RFID technology is getting more popular in large-scale applications especially in mobile environments, such as ticketing system for mass transportation and sport events. These applications work with offline RFID system which requires three components: RFID tags, readers and server. Tags are inherently mobile but they are not tamper resistant against any physical attack. Considering mobile hand-held devices, the readers are regarded as mobile and they are synchronizations of the database of the readers, and firmware updates. Although the reader in this model is offline during most of its life cycle, it still should be able to identify and authenticate the tags all the time. Such need requires the readers to have a higher resources and computational capacity compared to the online model. For instance, the ticket

intermittently connected to the central server only during verifier of a flying agent in the site of a sport event is connected to the server only when the agent is back to the headquarter. Therefore, the readers should be able to authenticate the customers [39] when the server is offline.

Besides, since the hand-held reader is mobile, the loss or the theft of a hand-held reader is a typical case of a threat for offline system. Since the privacy-preserving authentication protocols for identifying the tags are run by offline reader, there is no practical solution to renovate the privacy as soon as the readers are compromised by a malicious adversary. However, renewing all the tag information, which is impractical, can defeat this threat. The server hosts a centralized back-end system and manages data about the tickets and customers. Since the offline reader is not always connected to server, the detection of fraud (for example, the multiple use of tickets) is very difficult. Moreover, the firmware software or the configuration data of the reader are uploaded to the reader only at an inspection done by a maintenance personnel.

To exemplify the fear of compromise reader attacks in offline infrastructures, we consider a real-life RFID ticketing system deployed by RFIDea during a 3-day automobile race in 2009 [40]. This case study has been analyzed in [41]. In this deployment, several mobile readers and more than 100000 tags for tickets are used in order to reduce queues in the event and curtailing fraud. The system setup procedure works as follows. The mobile readers are first setup by the administrator and then given to the agents in the field until the end of the event. The mobile readers store the tags' secret keys in their database which are used for authentication and identification of all spectators' and employees' badges. The agents are not mobile, whereas spectators and employees are. Thus the offline RFID system can easily manage

the mobility of all the participants during the event. In this event, contrary to the expectations of the event organizer, some of the readers were stolen. With these readers, the participants are traced which violates the privacy. This showed that compromise of a reader attack can really happen.

2.3 Security and Privacy Threats and Cryptographic Background

As deployment of RFID in the world increases, potential security and privacy risks that they bring forward also increase. There are a variety of security threats in RFID systems. Since some of these security and privacy threats are mentioned by popular media, mass civic movements are formed against the use of RFID at different parts of the world. Several companies are taken to court as a result of using RFID tags in their products [42]. If precautions are not taken, mass utilization of RFID tagged items creates an approaching and potentially widespread threat to consumer privacy. To eliminate concerns of the public and to prevent possible future security and privacy problems, it is necessary to increase security and privacy level of RFID systems. Some of the possible security and privacy threats are discussed in this section.

2.3.1 Security Threats

An RFID system is perpetually under the threat of *man-in-the-middle attacks* ensuing from eavesdropping the communication between reader and tag. An adversary may monitor the messages during transmission and use or modify some parts of the messages. Then it can retransmit the messages maliciously to query the tag or the back-end server so as to impersonate the valid tag or

the valid reader. Another important attack is *replay attack* in which a valid message in the previous transmission is fraudulently used in another session. In an RFID scheme, if server has to authenticate a tag as well as to identify it, the scheme must prevent the replay attacks. One way of preventing this is the use of fresh random challenges in the hash calculations [43] or randomizing the responses. Moreover, an adversary can use a faulty/noisy tag or a jammer to cause tag/reader confusion during an authentication session and losing synchronization. Such an attack is called *desynchronization attack*. For instance, suppose a tag updates its shared secret values while the server does not; in such a case, the server is no longer able to authenticate the tags [6].

On the other hand, the tags, which are used in daily life applications, are expected to be low-cost and this restriction yields tag to have limited memory capacity and computational ability. Their memory is also considered as not tamper-resistant.

Furthermore, RFID authentication protocols are vulnerable to relay attacks, in which an attacker defeats the authentication system by only relaying messages from one legitimate party to another legal party (generally a prover and a verifier).

The seminal works of Desmedt et al. [44] and Beth et al. [45] on *mafia* and *terrorist frauds* demonstrated how an adversary can defeat such protocols by simply relaying the messages without dealing with cryptography. The concept of relay attack was originally proposed by Conway using a scenario called "'Chess Grandmaster Problem"' in 1976 [46]. In this scenario, a little girl plays remotely in parallel two correspondence games against two chess grandmasters. By only relaying the moves of the grandmasters she finally either defeats one of the grandmasters or draws against both. Also,

those kinds of attacks have been practically demonstrated in many different contexts and especially in RFID systems [47–51].

According to the capabilities of the adversary, relay attacks are simply classified as mafia, distance and terrorist fraud attacks [52]. Based on the authentication protocols that include challenge-response messages, mafia fraud scenario (see Figure 2.2) can be defined as follows. An adversary pretending to be a legitimate prover (or tag) first gets the challenge from the verifier (or reader) and relays it to the legitimate prover which is out of neighborhood (authentication region) at the beginning of the attack. After that she gets the valid response for this challenge and forwards it to the reader as her answer. Mafia fraud attack demonstrations and constructive considerations are addressed in [47, 49, 53]. The formal definition of the mafia fraud is given as follows.

Definition 1. *Mafia fraud [52]. A mafia fraud is an attack where an adversary defeats a distance bounding protocol using a man-in-the-middle (MITM) between the reader and an honest tag located outside the neighborhood.*

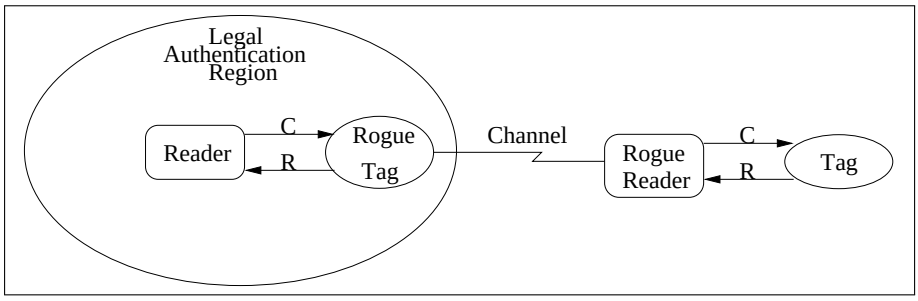

Figure 2.2: Mafia fraud scenario

Another type of attack is terrorist fraud in which the legitimate prover collaborates with an adversary in order to authenticate her when the former is out of the authentication region. In this attack, it is assumed that prover

helps the adversary without revealing any information of the long-term secret key. The formal definition of the fraud is given as follows.

Definition 2. *Terrorist fraud [52]. A terrorist fraud is an attack where an adversary defeats a distance bounding protocol using a man-in-the-middle (MITM) between the reader and a dishonest tag located outside of the neighborhood, such that the latter actively helps the adversary to maximize her attack success probability, without giving to her any advantage for future attacks.*

The example of home confinement can be given as an instance of the terrorist attack [52]. In this example, the arrested offender could get a help from his/her friends who stay close to electronically monitoring system. In such a condition, a terrorist fraud is needed because the ankle bracelet cannot be physically removed except by the authorities.

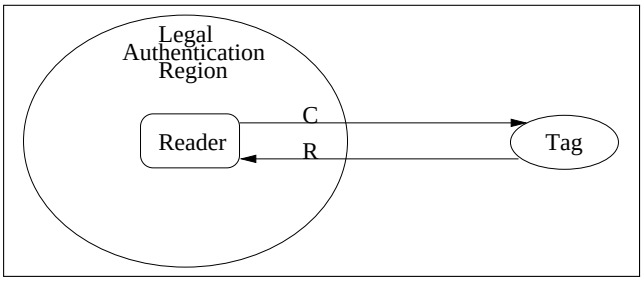

Figure 2.3: Distance fraud scenario

Similar to mafia fraud, there is also another attack called *distance fraud* (Figure 2.3). The distance fraud is an adversary that has an ability to reach secret key (e.g., a dishonest legitimate tag owner) to convince the verifier that she is within the neighborhood whereas she is not. Home confinement based on electronic monitoring with ankle bracelets is a typical example where distance fraud is definitely relevant. This fraud would allow the person under monitoring to temporary leave his residence without being detected.

2.3.2 Privacy Threats

An RFID tag may contain information about a person, item or product. Whenever a legitimate reader interrogates a tag, the tag sends its computed response to the reader. The communication between a tag and a reader could be eavesdropped by an adversary. Since RFID systems use shared unprotected radio medium, this makes such an attack more practical. The data obtained by the adversary can be misused in order to violate the anonymity of tag owners. These collected data might be valuable to some companies for marketing research or even thieves in search of wealthy victims [54]. This threat is classified as *tag information privacy violation*. This threat could be eliminated by controlling RFID systems so that only the authorized readers are able to access the information associated with a tag [6]. A further privacy concern is the possibility of tracking the tag. If the responses of a tag are correlated, then an adversary can record the responses obtained from readers at different locations. With this information, she can track the movement of the tag. In order to avoid this threat, the responses from the tags have to be anonymous.

Apart from these vulnerabilities, a strong adversary could tamper a tag and reach its long term secrets. After the tampering, the privacy for the previous responses of the tag could be questioned. Therefore, the schemes that are used for authentication/identification should satisfy security and privacy not only against passive attacks, replay attacks and cloning attacks, but also against strong adversaries.

2.3.3 Cryptographic Background

For a set S of any cardinality, $s \in_R S$ means s is chosen uniformly random among all elements of S. $y \in \{0,1\}^{\alpha}$ means y is any natural number such that y's bit length is at most α. For the case, $\alpha = *$, there is no restriction on bit length of y, i.e. y can be any natural number. A mapping $X : \{0,1\}^{\alpha} \rightarrow \{0,1\}^{\beta}$ means that X maps elements from $\{0,1\}^{\alpha}$ to $\{0,1\}^{\beta}$. Namely, the domain of X is $\{0,1\}^{\alpha}$ and the range of X is $\{0,1\}^{\beta}$. Let C be any algorithm, then $C(a) = b$ means, on input a, the algorithm C has b as output value. Let E be some event , then $Prob(E)$ denotes the probability that the event E happens. Moreover, $MSB_a\{k\}$ denotes most significant a bits of binary representation of k.

2.3.3.1 Hash Function

The definition of the hash functions used throughout the dissertation is given as follows.

Definition 3. *Hash Function. Let $k \in \mathbb{N}$ be a security parameter such that $\gamma \in \mathbb{N}$ is polynomially bounded by k. Define hash function $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^{2\gamma}$. Then $\mathcal{H}$ has the following properties:*

- *For any given input $m \in \{0,1\}^*$, the time required to calculate $\mathcal{H}(m)$ is polynomially bounded.*
- *Hash functions are pre-image resistant. That means, for any $c \in \{0,1\}^{2\gamma}$, it is infeasible to find $m \in \{0,1\}^*$ such that $\mathcal{H}(m) = c$.*
- *It is infeasible to find two different inputs giving the same output.*
- *Any probabilistic polynomial time adversary can distinguish between output of a $\mathcal{H}$ and random value with at most negligible probability.*

We treat hash functions as random oracles. Namely, the function $\mathcal{H}$ responds to every query with a truly random response chosen uniformly from $\{0,1\}^{\alpha}$. The function always gives the same response for a given input word.

2.3.3.2 Elliptic Curve Cryptography

Points on an elliptic curve are represented by capital letters while scalars are represented by lower-case letters. Let $\mathbb{E}$ be an elliptic curve with prime order p over $\mathbb{F}_p$, then for a point $Q = q_x, q_y$ with $q_x, q_y \in [0, \ldots, p-1]$, $xcoord(Q)$ maps Q to $q_x \bmod \ell$. We define $xcoord(O) = 0$, where O is the point at infinity. Note that the $xcoord(.)$ function is the ECDSA conversion function that comes almost for free when using elliptic curves [22,55]. In this dissertation, we also use the similar hash functions defined in Section 8.3. The security of our some proposals in the thesis depends on the hardness of solving discrete logarithm in elliptic cryptography and the formal definition of this problem is given as follows.

Definition 4. ***ECC Discrete Logarithm Problem.*** *Let P be a generator of a group $\mathbb{G}_\ell$ of order ℓ and let A be a given arbitrary element of $\mathbb{G}_\ell$ The discrete logarithm (DL) problem is to find the unique integer $a \in \mathbb{Z}_\ell$ such that $A = aP$.*

The difficulty of solving discrete logarithm problem in ECC is stated in the following remark.

Remark 1. *It is computationally hard to solve the Discrete Logarithm Problem for Elliptic Curves Cryptography. In fact the expected complexity to solve this problem is $e^{\mathcal{O}(\max(log(q), n\sqrt{log(q)}))}$, where the field that we work is F_{q^n} with $2^{5n} \leq q$ or $q = 2$ and $n^4 \leq q$.*

2.4 Literature on Security and Privacy in RFID Systems

Throughout the dissertation, in some of the authentication proposals, Physically Unclonable Function (PUF) is used to enhance security. In this context, we first provide definition of PUF and the related work on it. Then, we give related work on the solution of relay attacks. After that, we present the literature on the solution of privacy-preserving authentication solutions. Finally, we provide Vaudenay's privacy model, which is used as a basis in some of the chapters.

2.4.1 Physically Unclonable Functions (PUFs)

A Physically Unclonable Function (PUF) is a disordered physical structure implementing a unique function that maps challenges to responses. These responses depend on the nano-scale structural disorder of the PUF that is assumed to be unclonable or not even reproducible by the PUF's manufacturer. Namely, the PUF functions are embodied in a physical structure in a complex way upon several physical properties that the manufacturers cannot control, and they are easy to be computed, but difficult to be predicted, characterize and model the mappings.

The first attempt to exploit the physical properties of the devices for authentication purposes were done in [56–58]. Naccache and Fremanteau [59] later proposed an authentication mechanism for memory cards which uses these physical properties. The concept of PUFs is first introduced by Pappu [60, 61]. Their PUF functions were based on an optical principle of operation. In these PUFs, transparent tokens include randomly distributed scattering particles and are illuminated by a laser light with a specific angle, distance

and wavelength. The resulted speckle patterns from multiple scattering of laser in an incoherent optical medium are used for unique and unpredictable identifier. The challenge of the PUF can be the angle of incidence, the local distance or the wavelength of the laser. The responses can be hash value of digitized image of the speckle pattern. Afterward, several papers considered various hardware structures of PUF [62–65].

Besides, for a given challenge c, a typical PUF P may produce a slightly different response r ($r \leftarrow P(c)$) because the response depends on the physical characteristics that could be affected by environmental noises such as temperature, light and supply voltage variations. This obstacle can be eliminated by a small circuit, called Fuzzy Extractor and with additional helper input w [66, 67]. Moreover, even though two PUFs are implemented on the same device with the same structure, they both give independent responses with overwhelming probability for the same given challenges. Armknecht et al. proposed a formal foundation for such security primitives based on PUFs in [68].

The usage of PUFs in the authentication mechanisms has led to an increase in the security of existing RFID systems. They provide a new way for cost-efficient privacy preserving authentications based on the unclonable physical properties. In [62], it is shown that how PUFs can be used to establish a shared secret with a specific physical device. Namely, PUFs are embedded into a microchip. The first attempts to embody PUF functions into RFID authentication protocols are done in [69, 70]. In these studies, a set of challenge/response is derived from the PUF for each tag. The challenge/response pairs are stored in a secure database. The RFID reader selects a random challenge from the database and broadcasts it to the environment. Then, the received responses of the tags are interpreted by simply looking up

the database. The main obstacle of the scheme is that the challenge cannot be used anymore since it results in replay attacks. Another obstacle is storing huge amount of challenge/response in the database.

Tuyls et al. [71] used PUF functions as secure key derivation mechanism since PUF behaves like a hidden pseudo-random functions. Whenever a key hidden by PUF is needed during an authentication, it is simply derived by evaluating the PUF on the chip. Tuyls et al. assumed that as the adversary tries to evaluate a PUF or an IC, for instance, by using the probes to measure the wire delays, the characteristics of that particular PUF are changed. Thus, the intrinsic structure of the PUFs yields resistance against tampering and this reduces the capability of an adversary to clone an RFID tag. Moreover, they also demonstrated that PUF circuit can be easily implemented on RFID chips with less than 1000 gates [71].

In [72], another way of using PUF within a privacy-preserving RFID authentication scheme was proposed. In this scheme, for each ID of tag, the database of the reader stores the vector $\{ID, P(ID), P^2(ID), \ldots, P^t(ID)\}$ where t is the limit for authenticating a tag. Whenever the reader interrogates a tag, the tag evaluates its PUF with its identifier ID. The response is sent to the reader and the tag updates its ID with this response. The reader simply looks up the database, identifies the tag and removes the used response from the database. The main bottleneck of this protocol is that the system should store a huge amount of data for a large t. It also suffers from Denial of Service(DOS) attacks as the tag must be re-initialized after at most t sessions.

Sadeghi et al. [3] proposed a destructive private RFID authentication protocol based on PUF, which is similar to PUF functions of [71]. Whenever a strong adversary performs a physical attack, such as side channel on PUFs of RFID tags, these PUF functions are destroyed and cannot be evaluated

anymore. Moreover, several new authentication mechanisms based on PUF functions have been recently proposed in order to enhance their security and privacy levels [73–77].

2.4.2 Distance Bounding Protocols

In order to mitigate the frauds defined in Section 2.3, two main countermeasures have been adopted in RFID authentication protocols. The first one is based on measuring the radio signal strength (RSS) so that the verifier can learn whether the prover is close to it. This method has a drawback that a capable adversary can regulate its signal strength to convince the verifier that it is close to the verifier [78]. The second one is distance bounding approach suggested by Desmedt et al. [44,45]. This approach is a breakthrough to mitigate relay attacks by measuring the round trip time of short authenticated messages.

Brands and Chaum introduced the first distance bounding protocol [79]. This protocol aims to bring a solution to mafia and distance frauds. It consists of three phases, a slow phase, followed by a fast phase and a final signature phase. The first slow phase is used to exchange the committed random bits. The proximity verification is achieved by a bit-wise challenge-response during the second phase (i.e., fast phase), namely after series of n rounds where n is a security parameter. For each round of the fast phase, the verifier measures the round-trip time in order to extract the propagation time. Finally, the prover sends a final signature to the verifier and opens the commitments to complete the protocol. The success probability of mafia and distance frauds for this protocol are $(1/2)^n$, but it is not secure against terrorist fraud.

Čapkun et al. modified the Brands and Chaum's protocol to achieve mu-

tual authentication with distance-bounding [80]. However, their protocol is also vulnerable to terrorist fraud and is not resilient to bit errors during the rapid bit exchange.

Hancke and Kuhn proposed the first lightweight distance bounding protocol for RFID systems [78]. The major difference from Brands and Chaum's protocol is that it does not involve a final signature phase. This protocol involves a common secret symmetric-key k between a prover and a verifier. This protocol can be briefly described as follows. The verifier first generates a nonce N_v and sends it to the prover. Similarly, the prover also generates a nonce N_p and sends it to the verifier. Two n-bit registers R^1, R^2 are computed such that $R^1 \| R^2 = f(k, N_v, N_p)$ where f is a public pseudo-random function. After that, n-round fast phase starts. For each i-th round, the verifier picks a random challenge-bit c_i and sends it to the prover. The prover replies with a response-bit r_i such that

$$r_i = \left\{ \begin{array}{ll} R_i^0 & if c_i = 0 \\ R_i^1 & if c_i = 1 \end{array} \right\}.$$

The success probabilities of the mafia fraud and distance fraud are both equal to $(3/4)^n$ [34,78]. These studies triggered other researchers and several distance bounding protocols that use round trip time method have been proposed to increase security conditions against relay attacks [1, 53, 80–97].

One of the main obstacles of the existing distance bounding protocols is achieving the ideal security level (i.e., $(1/2)^n$ where n is a security parameter) against all frauds. However, achieving the ideal security against terrorist fraud is a very challenging task. Some attempts to thwart terrorist fraud [82] yield a more serious security problem; namely, the key recovery attack. This attack occurs due to the misuse of long-term key in the protocols [92].

On the other hand, Avoine et al. [52] introduced a unified framework for improving the analysis and the design of distance bounding protocols. The black-box and the white-box security models are introduced in the distance bounding domain, and the relation between the frauds are described with respect to these models. In the white-box model, the prover can provide more information to the adversary since the prover can access the internal key. We note that the security level of an RFID authentication in white-box model is generally lower than the security level in the black-box model.

2.4.3 Privacy-Preserving RFID Authentication Protocols

Mitigating the problems discussed in Section 2.3 requires the researchers to design identification/authentication protocols that include cryptographic mechanisms. On the other hand, most of RFID tags have limited memory and computational capability; therefore, the existing privacy-preserving mechanisms, which require high computational costs, are not applicable to many restricted RFID systems. Furthermore, most of RFID tags are not tamper resistant against strong adversarial attacks. Namely, physical attacks on tag's chip allow the adversary to learn the secrets stored in the tag. Thus, the design of a privacy preserving and cost-efficient RFID authentication protocol is very challenging task. To fulfill these needs, several authentication mechanisms have been proposed in the literature [1–7, 17].

The design of a privacy-preserving RFID authentication protocol is very difficult without a suitable security and privacy model. A large number of privacy models have been proposed to formalize security and privacy in the context of RFID system [18–27]. Vaudenay's model [18] is one of the most evolved and well defined privacy model. Moreover, Paise et al. [98] extended

Vaudenay's privacy model (PV-model). The model additionally offers reader authentication. Later, Armknecht et al. [99] showed that it is impossible to achieve both reader authentication and any reasonable notion of RFID privacy in the PV-Model, in which the target tags are vulnerable to corruption. On the other hand, Habibi and Araf [100] claimed that the privacy definition and adversary goal presented by Armknecht et al. is completely different from the PV-Model and the highest achievable privacy level in the Armknecht et al.'s privacy model is *narrow weak* privacy. The shortcomings of all recent privacy models are addressed in [28].

2.4.4 Vaudenay's Privacy Model

Throughout the dissertation, we use Vaudenay's privacy model [18] as a baseline during the security analysis of the proposals. Next, we first define the system procedures, adversary oracles and privacy experiments following the standard definitions of [18] for an RFID system. For the sake of simplicity, the reader and the server are assumed to be a single entity which are connected through a secure channel.

2.4.4.1 System Procedure

An RFID scheme is defined by the following procedures.

- SetupReader(1^ℓ) : This algorithm first produces a public-private key pair (K_P, K_S) where ℓ is the security parameter, then initializes its database $\mathcal{DB}$.

- SetupTag$_{K_P}$(ID): This algorithm generates a tag secret K and the initial state S of a tag with identifier ID. If this tag is legitimate, the pair (ID,K) is inserted into the database.

- IDENT: An interaction protocol between a tag and the reader to complete the authentication transcript.

2.4.4.2 Adversary Oracles

An adversary $\mathcal{A}$ can interact with the RFID system by the help of following generic oracles. First of all, $\mathcal{A}$ setups a new tag of identifier $\mathrm{ID}_{\mathcal{T}}$.

- CREATETAG($\mathrm{ID}_{\mathcal{T}}$) : It creates a free tag $\mathcal{T}$ with a unique identifier $\mathrm{ID}_{\mathcal{T}}$ by using $\mathbf{SetupTag}_{K_p}$. It also inserts $\mathcal{T}$ into $\mathcal{DB}$.

- LAUNCH()$\rightarrow \pi$: It makes the reader $\mathcal{R}$ start a new *Ident* protocol transcript π.

- SENDREADER(m, π)$\rightarrow m'$: This sends the message m to the reader $\mathcal{R}$ in the protocol transcript π and outputs the response m'.

- SENDTAG(m, π)$\rightarrow m'$: This sends the message m to $\mathcal{T}$ and outputs the response m'. Also, $\mathcal{A}$ asks for the reader's result of the protocol transcript π.

- DRAWTAG(distr)$\rightarrow(\mathcal{T}_1, b1, \ldots, \mathcal{T}_s, b_s)$: It randomly selects s free tags among all existing ones with distribution probability of distr. The oracle assigns a new pseudonym, $\mathcal{T}_i$ for each tag and changes their status to drawn. This oracle also returns bit b_i of tag i whether it is legitimate or not. The relations ($\mathcal{T}_i$,$\mathrm{ID}_{\mathcal{T}_i}$) are stored in a hidden table Tab. This hidden table is not seen by the adversary until the last step of the privacy game. Finally, the oracle returns all the generated tags in any order.

- FREE($\mathcal{T}$) : This oracle changes status of tag $\mathcal{T}$ from drawn to free. After that, $\mathcal{A}$ does no longer interact with $\mathcal{T}$.

- CORRUPT($vtag$)$\rightarrow S$: It returns volatile and non-volatile memory of the tag.
- RESULT(π)$\rightarrow x$: When π completes, returns $x = 1$ if the tag is identified, $x = 0$ otherwise.

2.4.4.3 Privacy Classes

Vaudenay's privacy model introduces five privacy classes of polynomial-time bounded adversary, determined by $\mathcal{A}$'s access to RESULT or CORRUPT oracles. These classes are defined as follows.

Definition 5. *(Adversary Classes [18]) An adversary $\mathcal{A}$ is a p.p.t. algorithm which has arbitrary number of accesses to the oracles described-above.* ***Weak*** *$\mathcal{A}$ uses all oracles except* CORRUPT *oracle.* ***Forward*** *$\mathcal{A}$ can only use* CORRUPT *oracle after her first call to this oracle.* ***Destructive*** *$\mathcal{A}$ cannot use any oracle against a tag after using* CORRUPT *oracle.* ***Strong*** *$\mathcal{A}$ uses all oracles described-above without any restrictions. Finally,* ***Narrow*** *$\mathcal{A}$ has no access to* RESULT *oracle.*

It is clearly seen that the following relation holds for these classes: WEAK$\subseteq$ FORWARD$\subseteq$ DESTRUCTIVE $\subseteq$STRONG.

Figure 2.4: The adversary classes ($\Rightarrow$: means that it implies.)

Strong	$\Rightarrow$	Destructive	$\Rightarrow$	Weak
$\Downarrow$		$\Downarrow$		$\Downarrow$
Narrow Strong	$\Rightarrow$	Narrow Destructive	$\Rightarrow$	Narrow Weak

2.4.4.4 Notion of Security and Privacy

The security definition given by Vaudenay's privacy model considers attacks in which the adversary aims to impersonate or forge a legitimate tag but not

security against cloning and availability.

Definition 6. *(Tag Authentication [18].) An RFID system achieves tag authentication if for every adversary, $\mathcal{A}^P$, where P is a class of adversary defined in Definition 5, is at most negligible.*

The privacy definition of Vaudenay is flexible and depends on the adversary classes in Definition 5, so it covers different notion of privacy. The privacy is simply based on the existence of a blinder $\mathcal{B}$, which is able to simulate each tag $\mathcal{T}$, and the reader $\mathcal{R}$ without knowing their secrets such that the adversary cannot distinguish whether it interacts with the real or simulated oracles. In the privacy game of Vaudenay's model, the participating entities are a set of tags, a protocol transcript π, and the reader. The adversary can interact with tags and with the reader by calling any oracle polynomial-bounded number of times according to her privacy class. The definition of the blinder is described as follows.

Definition 7. *(Blinder, trivial adversary [18]). A blinder $\mathcal{B}$ is a simulator which simulates* Launch, SendReader, SendTag, *and* Result *oracles without having access to the real secret keys and the database. When a blinded adversary $\mathcal{A}^{\mathcal{B}}$ uses these oracles, she is answered through the blinder $\mathcal{B}$. An adversary $\mathcal{A}$ is trivial if there exists a blinded adversary $\mathcal{A}^{\mathcal{B}}$ such that $Prob[\mathcal{A}\ wins] - Prob[\mathcal{A}^{\mathcal{B}}\ wins]$ is at most negligible.*

Remark 2. *The blinder $\mathcal{B}$ can simulate any tag or reader without knowing the secrets of corresponding tag or reader. Moreover, although there is no interaction between $\mathcal{B}$ and $\mathcal{A}$, the blinder $\mathcal{B}$ can see inputs and corresponding outputs of oracles applied by $\mathcal{A}$. Furthermore, the blinder $\mathcal{B}$ is consistent and acts like a real reader in a way that if a protocol transcript's inputs are derived as a result of usage of oracles to $\mathcal{B}$, the answer given by $\mathcal{B}$ to the* Result

oracle on this protocol transcript is 1. If all inputs of a protocol transcript are not derived as a result of usage of oracles to $\mathcal{B}$, then the answer given by $\mathcal{B}$ to the RESULT *oracle on this protocol transcript depends on the appearance probability of missing inputs on protocol transcript. Besides, $\mathcal{B}$ holds all its answers to the oracles used by $\mathcal{A}$ in its database and answers the new oracles depending on its database.*

We now explicitly describe Vaudenay's privacy game by the following experiment $Exp^{prv-b}_{\mathcal{A}_{prv}}$:

Let ℓ be a given security parameter, $b \in_R \{0,1\}$ and $\mathcal{A}_{prv}$ be an adversary given in Definition 5. There are two phases in the experiment: learning phase and challenge phase. In the learning phase, $\mathcal{R}$ is first set with $(sk_{\mathcal{R}}, pk_{\mathcal{R}}, \mathcal{DB}) \leftarrow$SETUPREADER$(1^{\ell})$. Both A_{prv} and $\mathcal{B}$ also get the public key $pk_{\mathcal{R}}$. Then, $\mathcal{A}_{prv}$ arbitrarily inquires all oracles defined in Section 2.4.4.2 but is limited to use the oracles according to her privacy class (See Definition 5). Whenever $b = 0$, A_{prv} simply calls real oracles. However, when $b = 1$, $\mathcal{B}$ receives and answers all queries to LAUNCH, SENDREADER, SENDTAG, and RESULT oracles. At this moment, $\mathcal{B}$ sees all oracles that are simulated by $\mathcal{B}$, but are made by A_{prv} ($\mathcal{B}$ sees what $\mathcal{A}_{prv}$ sees). These steps are done polynomial number of times. In the challenge phase, $\mathcal{A}_{prv}$ can no longer interact with the oracles but the hidden table Tab of DRAWTAG oracle is revealed to her. Finally, $\mathcal{A}_{prv}$ is expected to return an answer bit b', which is denoted by $Exp^{prv-b}_{\mathcal{A}_{prv}} = b'$. The formal definition of privacy is given as follows.

Definition 8. *(Privacy [18]). Let C be an adversary class defined as in Definition 5. An RFID system is C-private if $\forall \mathcal{A}_{prv} \in C$, there exists a*

p.p.t. algorithm $\mathcal{B}$ *such that the advantage*

$$Adv^{prv}_{\mathcal{A}_{prv}} = |Pr[Exp^{prv-0}_{\mathcal{A}_{prv}} = 1] - Pr[Exp^{prv-1}_{\mathcal{A}_{prv}} = 1]|$$

of $\mathcal{A}_{prv}$ *is at most negligible.* $\mathcal{B}$ *is the blinder, which simulates the* LAUNCH, SENDREADER, SENDTAG, *and* RESULT *oracles without having access to* $sk_{\mathcal{R}}$ *and* $\mathcal{DB}$*. Also, all oracles done by* A_{prv} *are sent to* $\mathcal{B}$

Chapter 3

K-STRONG PRIVACY FOR RFID AUTHENTICATION PROTOCOLS BASED ON PUFS

In the scope of this chapter, we first address the following privacy issue, which is not covered in Vaudenay's privacy model. Assume that a number of physical attacks (say k) are done on a target tag, after k^{th} corruption the tag is no longer usable. During the period of k corruptions, the adversary can interact with the tags and still get its internal state correctly. In Vaudenay's model, privacy in such scenario is not taken into account. This is the starting point of our work, in which we define the security and privacy levels between weak privacy and strong privacy.

The strongest achievable notion of privacy in Vaudenay's model, which is *strong privacy*, entails expensive public-key cryptography. This requirement generally exceeds the computational capabilities of current cost-efficient

RFID tags. In order to achieve the highest privacy level using only low cost cryptography, Physically Unclonable Functions (PUFs) have been studied. In the literature, several PUF-based authentication protocols have been proposed [3, 71, 101]. The security of these protocols relies on tamper-resistant structure of PUF devices which assumes that an attempt to measure physical parameters of PUF will definitely make it unusable. This assumption works only in ideal world whereas in the real case the PUF devices may be usable up to a number of physical attacks. If a PUF device is usable after the first successful physical attack, the security of such devices would be questionable. Therefore, it is not simple to decide whether the security of the system should rely on the protocol or on the tamper resistance of the device. Indeed, ultimate care is required for designing privacy-preserving protocols that the security relies on the tamper resistance of a device. We study these types of PUFs and introduce a new PUF definition, k-resistant PUF, which provides resistance against physical attacks at most k times where the integer value of k depends on the capability of adversary and manufacturing quality of PUFs. We show that the use of k-PUF helps to resolve the above-mentioned privacy issues in Vaudenay's model, the use of k-PUF helps to resolve the privacy issues mentioned above.

Our contributions are multiple. We first revisit Vaudenay's model and introduce two new privacy notions, k-strong privacy and k-forward privacy. Namely, we group all privacy classes of Vaudenay's model into two generic privacy classes. With this methodology, we construct a new privacy class between strong and destructive privacy.

In order to achieve highest security levels with only low-cost primitives, we study Physically Unclonable Functions (PUFs). We note that the security of the system relies on the assumption that physically tampering a PUF will

immediately destroy its physical structure and making it unusable. This is, actually, an assumption commonly used in the literature. However, in the real world, this assumption is not always correct because tamper resistance depends on the ability of the attacker and the quality of the manufacture and the design of the PUF circuit. The circuit may not be destroyed until some number of physical attacks (say k). Moreover, the structure of the PUF might be destroyed when unexpected environmental changes such as voltage, temperature changes occur and this destruction makes the PUF unreliable [102]. Therefore, we introduce a new extended PUF definition what we called k-resistant PUF (k-PUF). These PUFs are resistant against at most k number of physical attacks. After the k-th attack, the structure of the k-PUF is destroyed and can no longer be evaluated correctly. Also, k-PUF functions are more reliable against the k number of unexpected changes.

To illustrate our new privacy model, we analyzed two recent PUF based authentication protocols and show their security and privacy levels in our model [1, 3]. We show that these protocols do not achieve k-strong privacy for $k > 1$.

Next, we propose an efficient unilateral RFID authentication protocol based on k-PUFs. We prove that our protocol achieves k-strong privacy with low-cost cryptographic primitives such as hash functions and PUFs. When we choose k to be zero, 0-strong privacy implies weak privacy in Vaudenay's model, and when k is infinite, ∞-strong privacy implies strong privacy in Vaudenay's model. Therefore, to the best our knowledge, this is the first attempt to achieve strong privacy of Vaudenay's model only using symmetric cryptographic primitives.

Finally, we adapt and extend our generic authentication protocol to a mutual authentication. We prove that this extended protocol achieves both

k-strong privacy and reader authentication.

The organization of the chapter is as follows. Section 3.1 gives the motivation behind this study and formulate the problem statement. In Section 3.2, we first briefly describe PUF functions and its characteristics. Then we discuss the problem on the common PUF assumption and give our new PUF definition. Section 3.3 introduces our extended privacy model. Section 3.4 introduces two recent PUF based RFID protocols and analyze their security and privacy levels. In Section 3.4, we propose a simple generic PUF based RFID authentication protocol and analyze it with the help of our model. In Section 3.6, we prove that it is possible to provide both k-strong privacy and reader authentication in an RFID scheme. Section 3.7 concludes the chapter.

The results presented in Chapter 3 have been accepted in [29].

3.1 Motivation and Problem Statement

Vaudenay defines several adversary classes which cover almost all of the privacy levels in his seminal work [18]. Nevertheless, the following privacy issues are not considered in the model. Suppose that an adversary corrupts a target tag k times where k is an integer. During (and after) these attacks, the tag is still functional and the adversary can still interact with it and the privacy of the tag is satisfied. However, after the $k+1$-th corruption, the privacy of the tag is not satisfied. The security and privacy of this scenario is not addressed in Vaudenay's model. Note that when k goes to infinity, if the privacy of the tag is ensured against such an attack, then the strong privacy of Vaudenay's model is achieved. If k is equal to 1 and the privacy is still ensured, then the destructive privacy of Vaudenay's model is achieved. Similarly, if k is equal to 0, the weak privacy of Vaudenay's model is achieved. However, the

privacy levels are not defined in Vaudenays model in case of $k \geq 2$. This is the starting point of our work, in which we define the security and privacy levels between weak privacy and strong privacy notions for the first time in the literature.

We would like to highlight that the strong privacy of Vaudenay's model requires expensive public key cryptography. The driving motive behind this chapter is achieving security levels of $k \geq 1$ using only low cost primitives. In this context, we have studied PUF functions and the common assumption on the PUFs. Then, we defined a new generic PUF function, which we call k-PUF. With this new k-PUF function, we show that the security levels described above can be achieved.

Now, let us look at the assumption. A large body of literature dedicated to PUFs assumes that any attempt to tampering the PUF circuit in order to observe its internal states will *most likely* alter these variables or even destroy the structure of the circuit [64, 71, 101, 103–105]. Here, *most likely* means that in practice *some* circuits may stay working as usual after *a number of* physical attacks. In fact, it depends on the manufacturing structure of the circuit and the ability of the attacker. Therefore, it is a strong assumption to postulate that any PUF circuit will destroy after a single attack. In what follows, we examine this problem and give a more general statement for realistic circumstances by weakening this assumption.

Let p be the destruction probability of a given PUF after a single physical attack. The value of p depends on the attacker's capability and chip's level of strength against the physical attacks. The PUF circuit is assumed to be destructed if $p \geq P_{dest}$ where P_{dest} denotes a threshold value. If $p \geq P_{dest}$ after the first corruption then the circuit fulfills the best tamper-resistance property which corresponds to the ideal PUF case. More generally, let $P(X =$

i) denote the event of tag's evaluating not correctly after i-th corruption, then the probability of tag's not evaluating correctly at most k physical attack is

$$\sum_{i=1}^{k} P(X=i) = p\sum_{i=0}^{k-1}(1-p)^i = 1-(1-p)^k$$

where $k \geq 1$, $k \in \mathbb{Z}$. Thus, the tag cannot evaluate correctly if the condition below is satisfied

$$1-(1-p)^k \geq P_{dest} \Rightarrow k \geq \frac{\ln(1-P_{dest})}{\ln(1-p)}.$$

Note that the basic case of $k = 1$ corresponds to the ideal PUF. In the next section, we generalize the definition of ideal PUF by extending it to a more realistic sense by allowing limited number of attempts to tamper without destruction (up to a level of k).

3.2 Our New PUF Definition: k-PUF

In this chapter, we introduce a new PUF function definition (k-PUFs) that are resistant to at most k number of physical attacks. Contrary to the PUF of [1, 3, 71], after the k^{th} physical attack on the chip, the PUF inside the tag cannot be evaluated anymore because the structure of the PUF is destroyed with overwhelming probability. Similar to [1], we also assume that an adversary can reach to volatile and non-volatile memory of the tag in the case of physical attacks. The formal definition of our PUF is given as follows.

Let us denote $s \in_R S$ for choosing a value s uniformly at random from the set S. $y \in \{0,1\}^{\alpha}$ means y is any natural number such that y's bit length is at most α. For the case $\alpha = *$, there is no restriction on bit length of y,

i.e., y can be any natural number.

A mapping $f : \{0,1\}^{\alpha} \rightarrow \{0,1\}^{\beta}$ means that f maps elements from $\{0,1\}^{\alpha}$ to $\{0,1\}^{\beta}$. $Pr(E)$ denotes the probability of event E occurring. $MSB_a\{k\}$ denotes the most significant a bits of binary representation of k.

We are now ready to present our new definition of PUF as follows:

Definition 9. *(k-resistant PUF (k-PUF)) Let $\kappa \in \mathbb{N}$ be a security parameter such that $\beta, \theta \in \mathbb{N}$ are polynomially bounded in κ. Define an evaluation function of k-resistant PUF (k-PUF) $P_k : \{0,1\}^{\beta} \rightarrow \{0,1\}^{\theta}$. Then, P_k has the following properties:*

- *Same inputs always give same output result, i.e, let $P_k(a_1) = b_1$ and $P_k(a_2) = b_2$, if $a_1 = a_2$ then $Pr(b_1 = b_2) = 1$.*
- *Any probabilistic polynomial time adversary has at most negligible success probability to distinguish between output of P_k and a random value.*
- *k-PUF is resistant against any physical attack at most k times (e.g., invasive attack). Namely, P_k cannot be evaluated correctly anymore after k physical attacks.*

3.2.1 Practicality of k-PUF

In this section, we are going to provide some intuition about how to create a k-PUF structure. The coating PUF modeled by Tuyls et al. in [104] has a self destructing capability control where an invasive attack would probably cause to destroy PUF structure. This control detects the attack whenever the level of noise caused by the attack in the output of the PUF exceeds some threshold; so, if not detected, the PUF will not be destructed. This

makes coating PUF non-ideal in real life. If the PUF is destroyed after the first attack, this PUF could be considered as a natural example of k-PUF where $k = 1$. Our construction of k-PUF is inspired by the above-mentioned observation on [104] is described as follows.

The coating PUF can be built as top layer of an Integrated Circuit (IC) by applying circuit paths and laid out in a comb shape. These paths will be encased by a material that is randomly doped with dielectric particles of different size and dielectric strength. Each pair of circuit paths forms a capacitor with random capacitance, which again is unlikely to be controllable by the manufacturer. Random capacitor allows PUF to give a response with noise for a given challenge. In order to clean the noise from the response (i.e., error correction), helper data algorithm/fuzzy extractor is used for the reconstruction of secret keys [66,67]. Tuyls et al. [102,104] show that coating PUFs are resistant to an adversary who has the following optical and invasive methods.

- Optical inspection equipment to look into memory cells.
- Etching methods (e.g. chemical) to remove protective layers.
- Focused Ion Beam (FIB) to make holes in protective layers and allow for probing (of e.g. memory).

Since the coating is opaque, it is not so possible to look into the digital memory optically without damaging the coating [104]. Tuyls et al. [104] presented an advanced attack on the coating PUF where an adversary uses FIB to make an hole in the coating. The adversary uses her micro-probe(s) to retrieve the key bits during the reconstruction phase of the key. The use of FIB and micro-probes might give damage on the PUF. This damage causes the extracted key bits with more noise. It is stated that during reconstruction

phase, the extracted keys are checked with a signature. If the level of the noise is very high, then the computed signature would not be valid and the PUF would be destroyed by the controller. However, the adversary gets key bits with some noise during the attack. For example, if the PUF produces key length of 128-bits then the attacker can recover the complete bits with 2^{51} trials (we refer to [104] for further details.). We highlight that the level of noise in the PUF response is not only affected by the physical attacks but also affected by the unexpected significant environmental changes such as temperature, voltage changes. Thus, this environmental situation makes PUF unreliable.

The proposed k-PUF design is described as follows. We employ an additional counter, which is initialized to zero in the PUF control. The counter enables the PUF to limit the number of invasive attacks applied to the circuit. For example, a similar attack described above is performed, the PUF's control would detect the attack and it increments the counter by one because the attack causes the circuit to produce key bits with higher noise and Fuzzy Extractor is not able to produce a valid key and the signature would not be correct. When the counter is greater than or equal to $k-2$, the control in the PUF immediately destroys the circuit. In the worst case, in each attack, the adversary is assumed to recover a different key. In total she can gain at most $k-1$ different keys but in the k^{th} attack the structure of the PUF is destroyed. Hence the security of the our PUF is still protected. Moreover, our PUF functions are also vulnerable to environmental changes but they are reliable against $k-1$ number of unexpected changes.

3.3 Our Extended Security and Privacy Model

In this section, we borrow many of the privacy concepts of Vaudenay's privacy model, which is explained in Section 2.4.4 in detail. We extend this model by introducing two new classes of adversary, namely, k-strong and k-forward adversaries. After that, we introduce our privacy definitions.

3.3.1 Our Extended Privacy Experiment

Contrary to Vaudenay's model, we first introduce two new notion of adversary classes: k-strong adversary and k-forward adversary. The k is defined as an integer for privacy level. k-strong adversary covers three privacy classes of Vaudenay's model. These are Weak, Destructive and Strong adversaries. We finally give the formal definitions of k-strong and k-forward privacy according to these two new adversary classes.

Definition 10. *(k-Strong adversary). Let an RFID system $\mathcal{S}$ and a target tag $\mathcal{T}$ be given. Let also k be defined as a privacy level, which is an integer in $\mathcal{Z}^+ \cup \{0\}$. k-strong adversary $\mathcal{A}$ has the following capabilities:*

- *$\mathcal{A}$ can use* Corrupt *oracle on $\mathcal{T}$ at most k times.*
- *$\mathcal{A}$ cannot use any other oracles after $\mathcal{A}$ made its k^{th} corruption on the target tag.*
- *$\mathcal{A}$ can use all oracles if less than k* Corrupt *oracles are used.*

Definition 11. *(k-Forward Adversary). Let an RFID system $\mathcal{S}$ and a target tag $\mathcal{T}$ be given. Let also k be defined as a privacy level which is an integer in $\mathcal{Z}^+ \cup \{0\}$. k-forward adversary $\mathcal{A}$ has the following capabilities:*

- *$\mathcal{A}$ can use any other oracles until k^{th}* Corrupt *oracle on $\mathcal{T}$.*

- $\mathcal{A}$ *can use only* CORRUPT *oracle after* k^{th} CORRUPT *oracle on* $\mathcal{T}$.

Remark 3. *For the case* $k = 0$, $\mathcal{A}$ *can not use* CORRUPT *oracle on any tag, but* $\mathcal{A}$ *can use all oracles except* CORRUPT *oracle without any limitation.*

Next, we are now ready to define our privacy definition according to our new adversary classes. Note that this definition is almost similar to Vaudenay's privacy game except its adversary classes.

Definition 12. *(k-Strong Privacy). Let* $\mathcal{A}_{prv}$ *be a k-strong adversary defined as in Definition 10. An RFID system is k-Strong private if* $\forall \mathcal{A}_{prv}$, $\exists$ *a p.p.t. algorithm* $\mathcal{B}$ *such that the advantage*

$$Adv^{prv}_{\mathcal{A}_{prv}} = |Pr[Exp^{prv-0}_{\mathcal{A}_{prv}} = 1] - Pr[Exp^{prv-1}_{\mathcal{A}_{prv}} = 1]|$$

of $\mathcal{A}_{prv}$ *is at most negligible.* $\mathcal{B}$ *is the blinder, which simulates the* LAUNCH, SENDREADER, SENDTAG, *and* RESULT *oracles without having access to* $sk_{\mathcal{R}}$ *and* $\mathcal{DB}$. *Also, all oracles done by* A_{prv} *are sent to* $\mathcal{B}$.

Theorem 1. *When* $k = 0$, 0*-strong privacy implies* WEAK *privacy. When* $k = 1$, 1*-strong privacy implies* DESTRUCTIVE *privacy. When* $\lim_{k\to\infty}$, *k-strong privacy implies* STRONG *privacy.*

Proof. Let us start with the trivial cases. By remark 3, when $k = 0$, by definition, 0-strong privacy is equivalent to WEAK privacy. Moreover, when $k = 1$, by definition 3, 1-strong adversary cannot use any other oracles after the first CORRUPT oracle usage and the adversary can apply any oracle before the first CORRUPT oracle usage. Hence, this definition is equivalent to destructive adversary in Vaudenay's model.

For the $\lim_{k\to\infty}$, k-strong privacy case, we are going to prove the following claim.

Claim 1. $\lim_{k\to\infty}$ *k-strong privacy implies that the tag privacy protected against any number of* CORRUPT *oracle usage.*

Assume to the contrary the claim is wrong, then there exists integer k_0 such that after k_0 number of CORRUPT oracles are applied, the privacy of the tag is violated. However, by definition, $(k_0+1)-$strong privacy implies that the tag privacy is protected until $(k_0+1)^{th}$ CORRUPT oracle usage. Thus $\lim_{k\to\infty}$ k-strong privacy$\subset$ $(k_0+1)-$strong privacy.

Claim 2. $(k_0+2)-$*strong privacy*$\subset$ $\lim_{k\to\infty}$ *k-strong privacy.*

In fact, the problem is equivalent to the classical calculus problem, which is whether or not $(k_0+2) < \lim_{k\to\infty} k$. By undergraduate calculus, we know that $\lim_{k\to\infty} k{=}\infty$, so the claim holds.

Therefore, we have $\lim_{k\to\infty}$ k-strong privacy $\subset$ $(k_0+1)-th$ strong privacy $\subset$ $(k_0{+}2)-$strong privacy $\subset$ $\lim_{k\to\infty}$ k-strong privacy. This is a contradiction. Hence, the proposed claim holds.

Note that the tag's standing against any number of CORRUPT oracle usage corresponds to strong privacy in Vaudenay's model. Hence, $\lim_{k\to\infty}$, k-strong privacy in our model corresponds to strong privacy in Vaudenay's model. □

Remark 4. *Theoretically, one can claim that a tag can live forever regardless of how many times it has corrupted. However, in practice, it is impossible to create a tag standing against infinitely many number of corruptions physically. Hence,* $\lim_{k\to\infty}$ *k-strong privacy is more plausible to define for real world. For example, if a tag lives until* t^{th} *corruption, and until its destruction it gives no clue about privacy, then for this tag,* $t-$*strong privacy is equivalent to the strong privacy. However, this* t *value changes tag to tag so it is impossible to say that* t*-strong privacy is equivalent to strong privacy in*

Vaudenay's model for any $t \in \{Z\} - \infty$. *This theoretical approach covers this need.*

Moreover, one can claim that, if a tag lives until t corruption and until its destruction, it gives no clue about privacy, this tag also has $p-$*strong privacy where* $p >= t$. *Therefore, according to this perspective, for all the tags in the system, the system satisfies* $\lim_{k\to\infty}$ *k-strong privacy.*

There can be an adversary $\mathcal{A}$ such that $\mathcal{A}$ can corrupt a target tag k-times and $\mathcal{A}$ can interact with any oracle until its k^{th} corruption. In such case, the system should be private. Such a privacy is not handled in Vaudenay's model; however, k-strong privacy captures this concern.

On the other hand, k-forward privacy is similarly defined if an adversary $\mathcal{A}_{prv}$ is defined according to the Definition 11.

Hence, the new relations between our privacy classes holds as follows: 0-Forward⊆0-Strong ⊆ ... ⊆k-Forward⊆k-Strong.

3.4 Analysis of Two Recent Authentication Protocols

In this section, we analyze the security and privacy level of two recent PUF based authentication protocols according to our model.

3.4.1 Sadeghi et al.'s Authentication Protocol

Sadeghi et al. [3] use an ideal PUF (which corresponds to 1-PUF according to our model) in their proposed protocol. They assumed that whenever a strong adversary corrupts a tag, the adversary cannot reach to its temporary state and the structure of PUF would be destroyed. However, we assume that a

PUF cannot be destroyed immediately after the first corruption. Tags may have a limited number of resistance against any strong attacks. We briefly describe their protocol, then analyze the protocol according to our model.

Let $\ell \in \mathbb{N}$ be a security parameter, $\alpha, \beta, \gamma, \kappa$ be polynomial bounded in ℓ. Let $F : \{0,1\}^{\kappa} \times \{0,1\}^{2\alpha} \rightarrow \{0,1\}^{\beta}$ be a pseudo-random function. Each tag $\mathcal{T}$ is equipped with an ideal unique PUF function $P : \{0,1\}^{\gamma} \rightarrow \{0,1\}^{\kappa}$ and stores a random state $S \in_R \{0,1\}^{\gamma}$. On the other hand, the reader's $\mathcal{R}$ database $\mathcal{DB}$ stores a set of records (ID,K) for each tag in the system, where $K = P(S)$. The authentication protocol steps are summarized in Figure 3.1.

In the protocol, $\mathcal{R}$ first sends a random challenge $a \in_R \{0,1\}^{\alpha}$ to a tag $\mathcal{T}$. Once $\mathcal{T}$ receives the challenge, $\mathcal{T}$ picks another random challenge $b \in_R \{0,1\}^{\alpha}$. $\mathcal{T}$ reconstructs the secret key K and computes response $c = F_K(a, b)$ sends b and c to $\mathcal{R}$. Then, $\mathcal{T}$ erases a, b, c, K from its volatile memory. Upon $\mathcal{R}$ receives b, c from $\mathcal{T}$, $\mathcal{R}$ recomputes $c' = F_K(a, b)$ for each record (K,S) in $\mathcal{DB}$ until $\mathcal{R}$ finds a match ($c' = c$). If a match is found, $\mathcal{R}$ sends the ID, otherwise sends $\perp$.

Tag $\mathcal{T}_{ID}$ S		Reader $\mathcal{R}$ **DB** = $\{(\mathbf{ID}_1, \mathbf{K}_1), \ldots, (\mathbf{K}_n, \mathbf{ID}_n)\}$
$b \in_R \{0,1\}^{\alpha}$	$\xleftarrow{a}$	$a \in_R \{0,1\}^{\alpha}$
$K \leftarrow P(S)$		
$c \leftarrow F_K(a, b)$		
	$\xrightarrow{b,c}$	**If** $\exists$(ID,K) $\in \mathcal{DB}$
delete K, a, b, c		**s.t.** $c \stackrel{?}{=} F_K(a, b)$ **then**
		return ID
		else return $\perp$
		endif

Figure 3.1: Sadeghi et al.'s authentication protocol

Remark 5. *Note that output of a true random number generator and output of hash function in the random oracle model are indistinguishable. Therefore in practicality, outputs of pseudo-random functions and hash functions work similarly.*

Theorem 2. *The RFID protocol demonstrated in Figure 3.1 achieves* 0*-strong privacy.*

Proof. Without loss of generality assume that there are one reader $\mathcal{R}$ and one tag in the system (note that it is shown in [30] that a system with many tags and one reader has at most negligible advantage). First of all, we show that, if adversary is not allowed to use CORRUPT oracle, then the adversary cannot distinguish $\mathcal{R}$ from the blinder $\mathcal{B}$. Then, we show that if the adversary is allowed to use CORRUPT oracle at least once, then the adversary can distinguish $\mathcal{R}$ from $\mathcal{B}$.

In the first case, the system runs m times by $\mathcal{R}$ or $\mathcal{B}$. During the runs, the adversary guesses number of t values for K and checks the corresponding guessed key values at any of previous runs. Note that both m and t are polynomially bounded in ℓ. In order to calculate the maximum success probability, we have to consider two cases: (i) the probability that the adversary guesses the correct value of the key is $\frac{t}{2^\kappa}$. (ii) the probability that the adversary determines whether c is correct or not is $1-(1-(\frac{1}{2^\beta}))^m$. Since the values m and t are polynomially bounded the corresponding RFID scheme satisfies 0-strong privacy.

Let the adversary apply CORRUPT oracle at least once. Then, the adversary learns the value of K. For the consecutive protocol run, after getting values of a, b and c, the adversary computes the real value of c by using a, b and K and compares it with the given c value. The probability of distinguishing the real oracle from the blinder for only one protocol run is $1-\frac{1}{2^\beta}$. If

the adversary observes more protocol runs, her success probability increases. Since the advantage is non negligible, in fact close to 1, the system does not achieve k-strong privacy for $k \geq 1$. □

3.4.2 Kardas et al.'s Authentication Protocol

Kardas et al. [1] also proposed another PUF based authentication protocol and applied it into a distance bounding protocol and showed its security enhancements. Similar to Sadeghi et al.'s model, they also assume that whenever a strong adversary corrupts a tag, the PUF in the tag is destroyed; however, the adversary can reach its volatile memory only once. Their assumptions are weaker than Sadeghi et al.'s adversary model. In the following, we show that their protocol achieves 1-strong privacy according to our adversarial model. In this section, we first simplify Kardas et al.'s protocol without changing the core of the protocol. Then, we analyze its privacy level in our model. The authentication protocol steps are summarized in Figure 3.2.

Let $F : \{0,1\}^{\ell} \times \{0,1\}^{2\ell} \rightarrow \{0,1\}^{2\ell}$ be a one-way pseudo random function and $P_i : \{0,1\}^{k} \rightarrow \{0,1\}^{\ell}$ be an ideal PUF (1-PUF) function for tag $\mathcal{T}_i$. Each tag stores two random states $G'1_i, G_i^2 \in_R \{0,1\}^k$. On the other hand, the reader's database $\mathcal{DB}$ stores a set of records $(ID_i,\ K_i,\ L_i)$ for each tag $\mathcal{T}_i$ in the system, where $K_i = P_i(G_i^1)$ and $L_i = P_i(G_i^2)$. The authentication protocol is summarized in Figure 3.2.

The protocol starts with $\mathcal{R}$ sends a random challenge $a \in_R \{0,1\}^{\alpha}$ to a tag $\mathcal{T}_i$. Whenever $\mathcal{T}_i$ receives this challenge, it chooses another random challenge $b \in_R \{0,1\}^{\alpha}$. $\mathcal{T}_i$ reconstructs the secret key K_i and computes $T = F_K(a,b)$. Then, it deletes the K_i from its volatile memory. After that, $\mathcal{T}_i$ reconstructs the secret L_i by re-evaluating the PUF with G_i^2 $(L_i = P_i(G_i^2))$, calculates the response $c = F_{L_i}(T)$, and erases L_i from its volatile memory. $\mathcal{T}_i$ sends c along

with b to $\mathcal{R}$. Once $\mathcal{R}$ receives b, c from $\mathcal{T}_i$, it recomputes $c' = F_{L_i}(F_{L_i}(a, b))$ for each record (ID_i, K_i, L_i) in $\mathcal{DB}$ until $\mathcal{R}$ a match ($c' = c$) is found. If a match is found, $\mathcal{R}$ sends the ID, otherwise sends $\perp$.

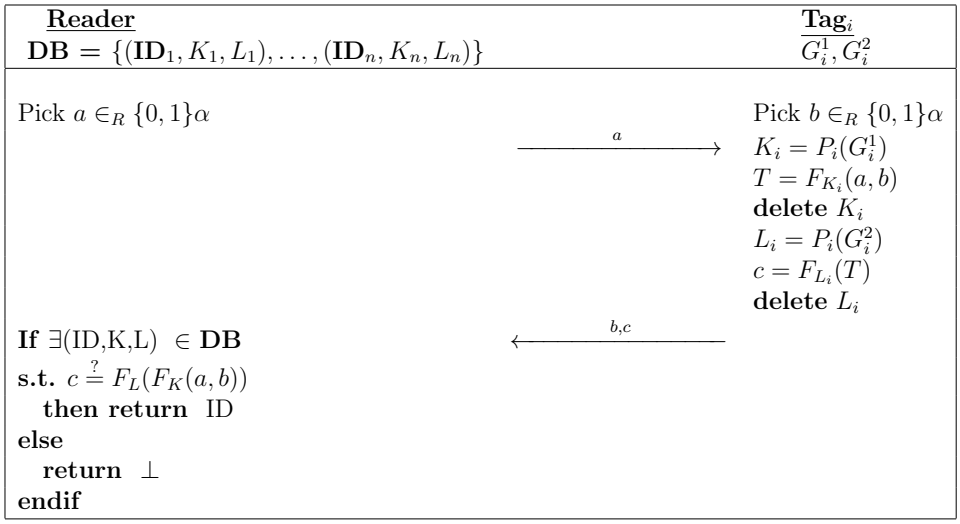

Figure 3.2: Kardas et al.'s authentication protocol

Theorem 3. *The RFID protocol demonstrated in Figure 3.1 achieves* 1*-strong privacy.*

Proof. Let there be one tag and one reader in the system [30]. We consider two cases. In the first case, the adversary is allowed to apply CORRUPT oracle at most once in order to maximize her success probability. As a second case, we investigate privacy issue when the adversary is allowed to use CORRUPT oracle more than once.

After the adversary applies the CORRUPT oracle, either the value of K or L is learned, but not both at the same time since the PUF P_i is 1-PUF, which means its function is destroyed after the $1st$ CORRUPT oracle usage. Similar to the calculations done in the proof of Theorem 2, if the system is run m times by blinder or the reader and the adversary guesses number of t values for the unrevealed key value (K or L). Then the maximum advantage

that the adversary gets in distinguishing the reader from the blinder is $\frac{t}{2^\kappa} + 1 - (1 - (\frac{1}{2^\beta}))^m$. Since m and t values are polynomially bounded, then the system achieves 1-strong privacy.

If the adversary applies corrupt oracle more than once, then both K and L are revealed in the worst case scenario. Similar to the calculations done in the proof of Theorem 2, the advantage that adversary has in order to distinguish the reader from the blinder is $1 - \frac{1}{2^\beta}$, which is non-negligible. Thus, the system does not achieve k-strong privacy for $k \geq 2$. □

3.5 k-Strong Private Authentication Protocol

Let κ be the security parameter of the system. Let $P_i : \{0,1\}^\beta \rightarrow \{0,1\}^\theta$ be a k-PUF of the i^{th} legitimate prover $\mathcal{P}_i$ where θ is polynomially bounded in κ. Let $\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^\gamma$ be one-way collision resistant hash function where where γ is polynomially bounded in κ. The credentials database $\mathcal{DB}$ of the reader $\mathcal{R}$ stores the following tag related information $((K_1^1, \ldots, K_1^{k+1}, ID_1), \ldots, (K_n^1, \ldots, K_n^{k+1}, ID_n)$ for $j = 1, \ldots, k+1$, $K^j = P_i(G_i \oplus j)$ for random states $G_i \in_R \{0,1\}^\beta$ where β is polynomially bounded in κ. Our unilateral authentication protocol depicted in Figure 3.3 works as follows.

- First of all, $\mathcal{R}$ generates a nonce $a \in_R \{0,1\}^\alpha$ and sends it to $\mathcal{T}_i$.
- Upon receiving a, $\mathcal{T}_i$ generates a nonce $b \in_R \{0,1\}^\alpha$ and computes $H = \mathcal{H}(a,b)$. $\mathcal{T}_i$ reconstructs $K^j = P_i(G_i \oplus j)$ and computes $H = \mathcal{H}(K^j, H)$, then immediately deletes K^j from the memory where $j = 1, \ldots, k+1$. The final value of H is assigned to c and $\mathcal{T}_i$ sends c along with b to the verifier.

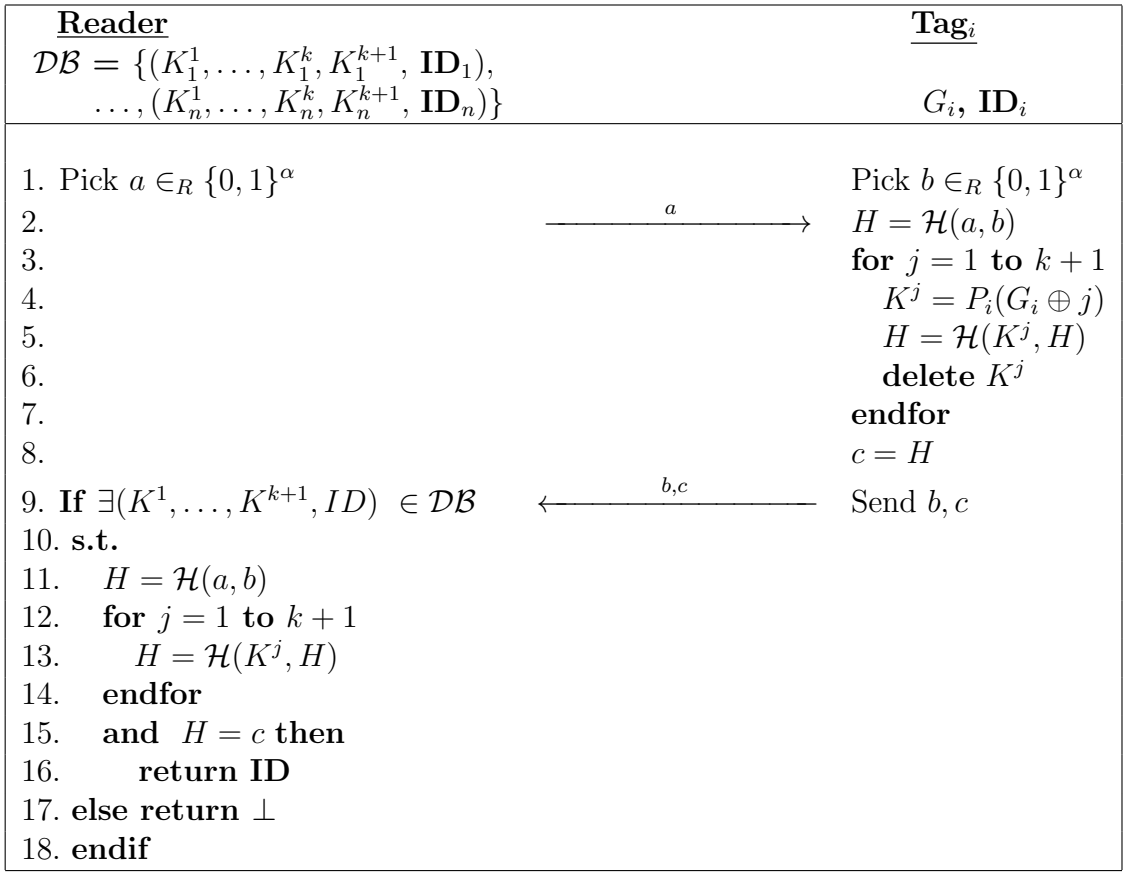

Figure 3.3: A generic PUF based authentication protocol

- Upon receiving b and c, for each record $(K^1, \ldots, K^{k+1}, ID)$ in $\mathcal{DB}$, $\mathcal{R}$ does following steps. $\mathcal{R}$ first computes $H = \mathcal{H}(a, b)$, then updates $H = \mathcal{H}(K^j, H)\ \forall j = 1, \ldots, k+1$. The last H value is assigned to b'. If a match ($c' = c$) is found, the authentication succeeded. Otherwise, $\mathcal{R}$ does these steps with another record in $\mathcal{DB}$. If no match is found, the authentication aborts.

3.5.1 Security Analysis

Throughout the chapter, we utilize the following rule. Let $B = \{1, \ldots, k+1\}$ be a set and $B_i = B/\{i\}$, where $i \in \{1, \ldots, k+1\}$. When it is said that CORRUPT oracle applied by B_i, we mean that the adversary captures all key values except the value of i^{th} key K^i. Moreover, throughout all proofs in this section, we assume that a tag is destructed at k^{th} CORRUPT oracle usage. This assumption does not restrict role of the adversary whereas this assumption gives the adversary the opportunity to take advantage of performing maximum number of oracles to any tag.

Lemma 1. *Let $\mathcal{A}_d$ be a k-strong adversary, $\mathcal{T}$ be a target tag and $B = \{1, \ldots, k+1\}$ be a set. Let B_i be $B/\{i\}$, where $i \in \{1, \ldots, k+1\}$. Then, the advantage that $\mathcal{A}_d$ obtains by applying* CORRUPT *oracle on tag $\mathcal{T}$ by the rules of B_i (not getting K_i) and the advantage that the adversary gets by applying* CORRUPT *oracle on tag $\mathcal{T}$ by the rules of B_j with $i \neq j$ are equal.*

Proof. Note that a set with $k+1$ elements has $k+1$ subsets having k elements. Thus, we can choose such two subsets (B_i, B_j) in $\frac{k(k-1)}{2}$ ways. Let us fix two integers i_0 and j_0 with $i_0 \neq j_0$ and $i_0, j_0 \in \{1, \ldots, k+1\}$.

Let m and n be polynomially bounded positive integers in κ. If $\mathcal{A}_d$ applies CORRUPT oracle on tag $\mathcal{T}$ by rules of B_{i_0}, then after k^{th} CORRUPT oracle us-

age, in the worst case, $\mathcal{A}_d$ has the knowledge of $K^1, \ldots, K^{i_0-1}, K^{i_0+1}, \ldots, K^{k+1}$. If $\mathcal{A}_d$ observes m number of protocol runs until k^{th} CORRUPT oracle usage, $\mathcal{A}_d$ also has knowledge of $(a_1, b_1, c_1), \ldots, (a_m, b_m, c_m)$. Then, $\mathcal{A}_d$ can compute c_{m+1} value in three cases:

- If a_{m+1} is equal to any of a_l values for $l \in \{1, \ldots, m\}$, then with 1 probability, the adversary figures out the value of c_{m+1} by choosing $b_{m+1} = b_l$.
- If this is not the case, $\mathcal{A}_d$ guesses number of n values of K^{i_0} and checks her guesses in any of the previous runs.
- In the case of failure, eventually the adversary has to guess the value of K^{k+1} or K^{i_0} for the corresponding protocol run.

Thus, the success probability of $\mathcal{A}_d$ is $\frac{m}{2^\alpha} + \frac{2^\alpha - m}{2^\alpha}\left[\frac{n}{2^\theta} + \frac{2^\theta - n}{2^\theta}\left(\frac{1}{2^\gamma} + \frac{1}{2^\theta - n}\right)\right]$. Similarly, if the CORRUPT oracle usage applied by the rules of the set B_{j_0}, one deduces that $\mathcal{A}_d$ gets the same success probability. The result follows by the fact that i_0 and j_0 are chosen arbitrarily. □

From now on, when it is said that a tag is corrupted, it should be understood that it is corrupted by rules of $B_{k+1} = B/\{k+1\} = \{1, \ldots, k\}$.

Lemma 2. *Let $\mathcal{A}_d$ be a k-strong adversary and $\mathcal{T}_t$ be the target tag. Then $\mathcal{A}_d$'s analyzing the system with many tags including $\mathcal{T}_t$ gives him at most negligible advantage over her analyzing the system with only $\mathcal{T}_t$.*

Proof. Assume that there are one reader and n tags in the system, where n is polynomially bounded in κ. For every $i \in \{1, \ldots, n\}$, the reader and tag $\mathcal{T}_i$ realize m_i number of the protocol runs before k^{th} corruption. Note that our aim is to observe the adversarial advantage difference between analyzing the

systems with multiple tags and single tag. Thus, we have to figure out how much $\mathcal{A}_d$ gets advantage by guessing the value of c_{m_t+1} after corrupting $\mathcal{T}_i$ and observing the protocol runs realized by $\mathcal{T}_i$, $i \in \{1, \ldots, t-1, t+1, \ldots, n\}$. Since the value of G_i and the PUF function P_i differ from tag to tag, the only advantage of $\mathcal{A}_d$ is to find relations among the keys or the resulting c values. By letting $m = \max\{m_1, \ldots, m_{t-1}, m_{t+1}, \ldots, m_n\}$, the total advantage is at most $km(n-1)\frac{1}{2^\theta} + m(n-1)\frac{1}{2^{2\theta}} + m(n-1)\frac{1}{2^\gamma}$. Since n, k and m are polynomially bounded in κ and θ is sufficiently large, the advantage is at most negligible. □

From now on, in the theorems stated below, we assume there are only one reader R and one tag $\mathcal{T}$, target tag, in the system.

Theorem 4. *The RFID protocol demonstrated in Figure 3.3 achieves tag authentication for a k-strong adversary $\mathcal{A}_k$.*

Proof. Let κ be the security parameter in the RFID system. According to Lemma 2, there are only one tag, $\mathcal{T}$ and one reader, $\mathcal{R}$ in the system. Note that the adversary does not need to apply CreateTag, DrawTag and Free oracles. $\mathcal{A}_k$ can use SendReader(π) oracle to start a protocol run either between $\mathcal{R}$ and $\mathcal{T}$ or between $\mathcal{R}$ and himself. Furthermore, $\mathcal{A}_k$ can use Result oracle polynomially bounded in κ number of times by sending b and c values to the reader for corresponding a values, which are sent by $\mathcal{R}$ as a result of the usage of SendReader(π) oracle. Moreover, $\mathcal{A}_k$ can use SendTag oracle polynomially bounded in κ number of times to send a challenge value a to $\mathcal{T}$. Besides, $\mathcal{A}_k$ can use Corrupt oracle at most k times and we assume that the adversary exactly applies Corrupt oracle k times to increase her chance to destroy tag authentication.

By Lemma 1, we assume that $\mathcal{A}_k$ applies Corrupt oracle by rules of

the set B_{k+1}. Moreover, let us assume that $\mathcal{A}_k$ observed m_1 number of protocol runs between $\mathcal{R}$ and $\mathcal{T}$ and queried SENDREADER(π) oracle m_2 times to start protocol run between $\mathcal{R}$ and $\mathcal{T}$. Furthermore, $\mathcal{A}_k$ uses SENDTAG oracle m_3 times. Note that m_1, m_2, m_3 are polynomially bounded integers in κ and in order to increase the success probability of $\mathcal{A}_k$'s destroying tag authentication, we assume that in all protocol runs, occurred as a result of above oracle usages and observation, different a values are used. Moreover, assume that SENDREADER(π) oracle is used m_4 times to start protocol run between the reader and the adversary. After k^{th} corruption, $\mathcal{A}_k$ uses m_5 number of SENDREADER(π) oracles to start protocol run between the reader and herself. In each of these runs $\mathcal{A}_k$ receives different a values, then she generates a pair (b, c) and $\mathcal{A}_k$ sends this pair to the reader and finally $\mathcal{A}_s$ uses RESULT oracle for triple (a, b, c). Assume the adversary has y chances to impersonate the corresponding tag without using any oracle where y is polynomial bounded in κ. Moreover, $\mathcal{A}_k$ is allowed to prepare p_i triples (K^{k+1}, b_i, c_i) for corresponding impersonation trial i. Note that these triples are prepared according to guesses of $\mathcal{A}_k$ on the value of the missing key. $\mathcal{A}_k$ checks if any of the triples is true or false based on the protocol transcripts reached so far at each impersonation round. If $\mathcal{A}_k$ has no success at p_i triples, then the adversary just guesses the values of b and c. Let us denote $M = m_1 + m_2 + m_3 + m_4 + m_5$ and $P = \max\{p_1, \ldots, p_y\}$. Note that M and P are polynomially bounded in κ. Let us figure out the success probability of the adversary at i^{th} impersonation trial. The reader sends a_i as a challenge to the adversary. If a_i is equal to any of the a values that were used at previous successful protocol transactions observed or created by oracle usage, then with 1 probability, the adversary succeeds. However, the probability of realization of this scenario is at most $\frac{M}{2^\alpha}$. In case of failure, then $\mathcal{A}_k$ checks

correctness of each p_i triple. However, the success probability of $\mathcal{A}_k$ in this case is at most $\sum_{l=(i-1)P-1}^{iP-2}[(\prod_{j=0}^{l}(1-\frac{1}{2^\theta-j}))\frac{1}{2^\theta-l-1}]$. If the adversary fails after two cases discussed above, then she guesses the values of b and c. At each trial, the success probability is $\frac{1}{2^\gamma-P}$.

Thus, maximum success probability of $\mathcal{A}_k$ at the end of y^{th} impersonation trial is smaller than $\frac{yM}{2^\alpha}+(1-\frac{M}{2^\alpha})[\frac{1}{2^\theta}+\sum_{i=0}^{yP-2}[(\prod_{j=0}^{i}(1-\frac{1}{2^\theta-j}))\frac{1}{2^\theta-i-1}]]+(\frac{y}{2^\gamma-P})$. Let us denote above probability by B. Then,

$$\begin{aligned} B &\leq \frac{yM}{2^\alpha}+\sum_{i=0}^{yP-2}\frac{1}{2^\theta-i-1}+\frac{y}{2^\gamma-P} \\ &\leq y\left[\frac{M}{2^\alpha}+\frac{P}{2^{\theta-1}}+\frac{1}{2^\gamma-P}\right] \end{aligned} \tag{3.1}$$

The resulting probability is negligible since y, M and P are polynomially bounded and α, θ and γ are big enough. Thus the system satisfies tag authentication. □

Theorem 5. *The RFID protocol demonstrated in Figure 3.3 achieves k-strong privacy.*

Proof. Assume to the contrary, the system does not satisfy k-strong privacy. Then, there exists an adversary $\mathcal{A}_k$, who can distinguish between the real RFID system and the system simulated by a blinder $\mathcal{B}$ with non-negligible probability. By definition, $\mathcal{B}$ simulates LAUNCH, SENDTAG, SENDREADER and RESULT oracles without knowing the tag and the reader secrets.

Let us start with how $\mathcal{B}$ evaluates the oracles:

- LAUNCH(): $\mathcal{B}$ evaluates this oracle in a trivial way.
- SENDREADER(π): The output is $a \in_R \{0,1\}^\alpha$.

- SendTag(a): The output is $b \in_R \{0,1\}^{\alpha}$, $c \in_R \{0,1\}^{\gamma}$.
- SendReader((b,c),π): returns no output.
- Result(π): If π is generated by Launch oracle and the protocol transcript is generated by SendTag and SendReader oracles, the output is 1. If one of the conditions does not hold, then the output is 0.

By Lemma 2, we assume that there are only one tag and one reader in the system. Moreover, for simplicity and to increase the success probability of $\mathcal{A}_k$ to destroy the privacy, we assume the database of the reader is not updated throughout the proof. Let the system run for n times only by real RFID system or the blinder $\mathcal{B}$, where n is polynomially bounded integer in κ. In other words, all usable oracles defined at Section 2.4.4.2 is used at most n times. Moreover, by Lemma 1, assume that Corrupt oracle is applied by the rules of the set B_{k+1}.

There are three cases to consider: The first case is guessing of the value of K^{k+1}. The probability of this happening is $\frac{1}{2^{\theta}}$. The second case is $\mathcal{A}_k$ to determine the correct value of c in at least one of the protocol runs. The probability of this case is $1-(1-\frac{1}{2^{\gamma}})^n$. The last case is $\mathcal{A}_k$ to guess the value that is produced by the Result oracle is correct or wrong successfully.

By contradiction assumption, since $\mathcal{A}_k$ destroys the privacy, either one of two probabilities given above is non-negligible or the probability of realization of the last case is non-negligible. However, with sufficiently large θ and γ values, first two probabilities are negligible. Thus, the success probability of $\mathcal{A}_k$ to guess the value that is produced by the Result oracle is correct or wrong is non-negligible. However, this contradicts with Theorem 4, namely, contradicts to the tag authentication. □

3.6 Adapting Our Protocol to Reader Authentication

The privacy definition given by Paise and Vaudenay (P-V) is based on the anonymity of the tags and unlink-ability of the interactions. The privacy of an RFID scheme is broken when an adversary identifies a victim tag or links its interactions [98]. Nevertheless, Armknecht et al. define privacy as the ability of an adversary to distinguish real oracles from the blinder $\mathcal{B}$ [99]. The concept of privacy in the P-V model is based on distinguishing between different tags, whereas in the Armknecht et al.'s model the privacy is defined based on the notion of (left-or-right) or (0-or-1) indistinguishability game. Therefore, their results on the privacy with reader authentication are different.

By using [99] approach, Habibi et al. claim that the highest achievable privacy level is narrow-weak privacy with reader authentication [100]. However, in this section, we prove that it is possible to achieve k-strong privacy and reader authentication by introducing a PUF based RFID mutual authentication protocol. This is the first attempt to provide both these security and privacy properties in the literature. For our proposed mutual authentication protocol, we first give definitions of two functions, $\mathcal{F}_{tag}$, $\mathcal{F}_{reader}$ which combine some steps of computation at tag and reader side respectively. These functions make our next protocol more readable. The function $\mathcal{F}_{tag}$ requires two random challenges (a, b), the initial nonce G and k number of the internal steps. $\mathcal{F}_{tag}$ does the computation from step 2 to step 6 at the tag side (see Figure 3.3). The process depicted in Figure 3.4.

$\mathcal{F}_{reader}$ takes two challenges (a,b) and the secret keys of a tag $(K^1, \ldots, K^{k+1})$ and produces the output H. It simply does the computation from step 11 to

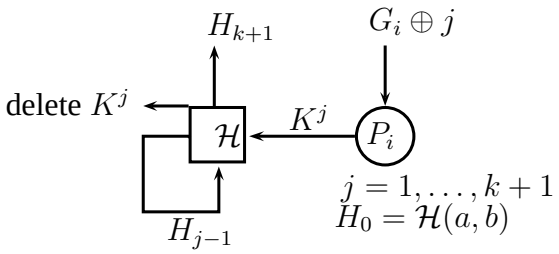

Figure 3.4: A generic function $\mathcal{F}_{tag}(a, b, G_i, k+1) = H_{k+1}$

step 14 at the reader side(see Figure3.3). The process depicted in Figure 3.5.

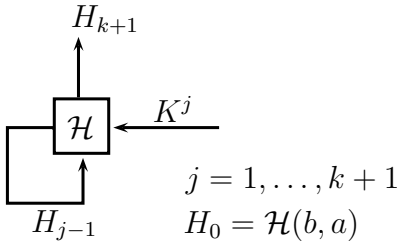

Figure 3.5: $\mathcal{F}_{reader}(b, a, K^1, \ldots, K^{k+1}) = H_{k+1}$

Note that the notations used in the protocol are already described in Section 3.5. The extended mutual authentication protocol works as follows. First of all, $\mathcal{R}$ generates a random nonce a and sends it to $\mathcal{T}_i$. As receiving a, $\mathcal{T}_i$ generates a random nonce b and computes $c = \mathcal{F}_{tag}(a, b, G_i, k+1)$ and sends c along with b to the reader. Then, for each record $(K_j^1, \ldots, K_j^{k+1}, ID_j)$ in $\mathcal{DB}$ where $j \in \{1, \ldots, n\}$, $\mathcal{R}$ computes $c = \mathcal{F}_{reader}(a, b, K_j^1, \ldots, K_j^{k+1})$. If a match $(c' = c)$ is found, then the tag authentication succeeds and $\mathcal{R}$ computes $d = \mathcal{F}_{reader}(b, a, K_j^1, \ldots, K_j^{k+1})$ and sends d to $\mathcal{T}_i$. If no match is found in $\mathcal{DB}$, $\mathcal{R}$ sends random bits to $\mathcal{T}_i$. Finally, upon receiving d, $\mathcal{T}_i$ computes $d' = \mathcal{F}_{tag}(b, a, G_i, k+1)$ and if d is equal to d', then the reader authentication succeeds.

Reader		**Tag**$_i$
$\mathcal{DB} = \{(K_1^1, \ldots, K_k^1, K_{k+1}^1, \mathbf{ID}^1),$ $\ldots, (K_1^n, \ldots, K_k^n, K_{k+1}^n, \mathbf{ID}^n)\}$		G_i, $\mathbf{ID}_i$
1. Pick $a \in_R \{0,1\}^\alpha$		Pick $b \in_R \{0,1\}^\alpha$
2.	$\xrightarrow{a}$	$c = \mathcal{F}_{tag}(a, b, G_i, k+1)$
3.		Send b, c
4. **If** $\exists(K_j^1, \ldots, K_j^{k+1}, ID_j) \in \mathcal{DB}$	$\xleftarrow{b,c}$	
5. **s.t.** $c \stackrel{?}{=} \mathcal{F}_{reader}(a, b, K_j^1, \ldots, K_j^{k+1})$		
6. **then**		
7. **return** $d = F_{reader}(b, a, K_j^1, \ldots, K_j^{k+1})$		
8. **else return** $d \in_R \{0,1\}^\gamma$		
9. **endif**		
10.	$\xrightarrow{d}$	$d \stackrel{?}{=} \mathcal{F}_{tag}(b, a, G_i, k+1)$

Figure 3.6: A generic PUF based mutual authentication protocol

3.6.1 Security and Privacy Analysis

In this section, we first prove that our protocol achieves reader authentication. Then we utilize this proof in order to prove the protocol also provides k-strong privacy. Note that, throughout all proofs in this section, we assume that a tag is destructed at k^{th} CORRUPT oracle usage. This assumption gives the adversary the opportunity to take advantage of performing maximum number of oracles to any tag.

Theorem 6. *The RFID protocol demonstrated in Figure 3.6 achieves reader authentication for k-strong adversary $\mathcal{A}_k$.*

Proof. By Lemma 2, let there be one reader, $\mathcal{R}$ and one tag, $\mathcal{T}$ in the system. Also, the adversary $\mathcal{A}$ has applied CORRUPT oracle to $\mathcal{T}$ k times with rules of B_{k+1}. Besides, A_k observes m_1 number of the protocol runs between $\mathcal{R}$ and $\mathcal{T}$. Also assume that A_k applies following oracles with given number of times before authentication game as described below:

1. m_1 times: no oracle usage, the adversary just watches protocol run between $\mathcal{R}$ and $\mathcal{T}$

2. m_2 times: SENDREADER(π) oracle to start protocol run between $\mathcal{R}$ and $\mathcal{T}$

3. m_3 times: SENDTAG(a) oracle and SENDREADER(b, c) , where SENDTAG(a) $\rightarrow (b, c)$

4. m_4 times: A_k derives (b, c) and uses SENDREADER(b, c) and RESULT(d) oracles, where SENDREADER(b, c) $\rightarrow d$.

In order to increase the success probability of A_k, let us assume that the value of a that is sent to tag by the adversary or derived as a result of SendReader(π) oracle is fixed. Moreover, let us assume that different b, c values are used by the adversary or the tag as a result of SendTag(a) oracle usage.

Let the adversary have y number of chances in order to impersonate the corresponding reader without using any oracle. Moreover, $\mathcal{A}_k$ is allowed to prepare p_i pairs (K_j^{k+1}, d_j^i), $j = 1, \ldots, p_i$, for corresponding impersonation trial i. Note that these pairs are prepared according to guesses of $\mathcal{A}_k$ on value of missing key. $\mathcal{A}_k$ checks if any pair created is true or false based on the protocol transcripts reached so far at each impersonation round. If $\mathcal{A}_k$ has no success at p_i pairs, then the adversary just guesses the values of d^i.

Let us denote $M = m_1+m_2+m_3+m_4$ and $P = \max\{p_1, \ldots, p_k\}$ where M and P are polynomially bounded positive integers in κ. Let us figure out the success probability of the adversary at i^{th} impersonation trial. Assume that the adversary sends a to the tag. If the tag responds with (b, c) pair value that was used previously while using the oracles defined above, then the adversary succeeds with probability 1. If this is not the case, then A_k checks the correctness of each (K_j^{k+1}, d_j^i), $j = 1, \ldots, p_i$. However, the success probability of $\mathcal{A}_k$ in this case is at most $\sum_{l=(i-1)P-1}^{iP-2} \left[\left(\prod_{j=0}^{l} \left(1 - \frac{1}{2^\theta - j}\right)\right) \frac{1}{2^\theta - l - 1}\right]$. If the adversary fails after two cases discussed above, then she guesses the values of d^i. At this trial the success probability is $\frac{1}{2^\gamma - P}$.

Thus, maximum success probability of $\mathcal{A}_k$ at the end of y^{th} impersonation

trial is smaller than

$$\frac{1}{2^\theta}\left(1-\frac{M}{2^\alpha}\right)+\sum_{i=0}^{yP-2}\left[\left(\prod_{j=0}^{i}\left(1-\frac{1}{2^\theta-j}\right)\right)\frac{1}{2^\theta-i-1}\right] + \frac{yM}{2^\alpha}+\left(\frac{y}{2^\gamma-P}\right)$$

Let us denote above probability by B. Then,

$$\begin{aligned} B &\leq \frac{yM}{2^\alpha}+\sum_{i=0}^{yP-2}\frac{1}{2^\theta-i-1}+\frac{y}{2^\gamma-P} \\ &\leq y\left[\frac{M}{2^\alpha}+\frac{P}{2^{\theta-1}}+\frac{1}{2^\gamma-P}\right] \end{aligned} \tag{3.2}$$

The resulting probability is negligible by the same argument since y, M and P are polynomially bounded in κ and α, θ and γ are big enough. Thus the system achieves reader authentication. □

Theorem 7. *The RFID protocol demonstrated in Figure 3.6 achieves both k-strong privacy and reader authentication.*

Proof. Note that by Theorem 6 the system achieves reader authentication. Thus, we only need to prove k-strong privacy.

Assume to the contrary, there exists an adversary $\mathcal{A}_k$ who can distinguish the real RFID system and the system simulated by the blinder $\mathcal{B}$. The blinder simulates the oracles as it is defined at proof of Theorem 5 except SendReader$((b,c),\pi)$ oracle. In this case, $\mathcal{B}$ evaluates this oracle and it outputs $d \in_R \{0,1\}^\gamma$. Moreover, there is one more oracle $SendTag(d,\pi,end)$ simulated by $\mathcal{B}$. The blinder returns no output to this oracle.

By Lemma 2, let there be one tag and one real reader in the system. Moreover, let us assume that the reader is not updated throughout the proof. Let $\mathcal{A}_k$ apply the Corrupt oracle k times by the rules of the set B_{k+1} by

Lemma 1 and the system runs y times before distinguish-ability phase.

There are four cases to consider. The first case, as indicated at proof of Theorem 6, is the value of K^{k+1} or the value of c is determined correctly by the adversary $\mathcal{A}_k$ at least one protocol run by obtained information. However, the probabilities are $\frac{1}{2^\theta}$ and $1-(1-\frac{1}{2^\gamma})^y$ respectively.

The second case is to make $\mathcal{A}_k$ to determine the answers given from usage of RESULT oracle true or false after receiving $d \leftarrow$SENDREADER(b,c). Nonetheless, this is possible only if $\mathcal{A}_k$ knows the value of K^{k+1} but this can only happen with probability of $\frac{1}{2^\theta}$. The third case is that the correct value of d is determined by $\mathcal{A}_k$'s at least in one of the protocol runs. This probability is $1-(1-\frac{1}{2^\gamma})^y$. The last case is the value of c or d is guessed correctly by $\mathcal{A}_k$. However, the success probability is $\frac{1}{2^{\gamma-1}}$.

As all calculated probabilities are negligible and finite sum of negligible numbers are negligible. Thus we have a contradiction. Namely, $\mathcal{A}_k$ has at most negligible advantage at distinguishing the real system from the blinder. Thus, the system satisfies k-strong privacy. □

3.7 The Summary of the Chapter

In this chapter, we revisited Vaudenay's privacy model, which is one of the well-known models in RFID frameworks. We went one step further and introduced two new notions of adversary classes, k-strong adversary and k-forward adversary. These two adversary classes cover all the classes defined by Vaudenay's model and yield two new privacy classes, k-strong privacy and k-forward privacy. Contrary to Vaudenay's model, our model covers the security level between destructive privacy and strong privacy.

We also proposed a new extended PUF definition k-PUFs. Ideal PUFs

are assumed to be destroyed once tampered. However, our proposal extends this assumption to the real case, i.e., these types of PUFs are tamper proof up to k corruptions. This new type of PUFs seems to be more plausible than prior proposals. This approach can also be considered as a more realistic scenario to analyze RFID authentication protocols.

Next, we give two robust PUF based authentication protocols to illustrate different privacy levels in our new extended model. In our first protocol, we prove that the strong privacy (∞-strong privacy in our model) in Vaudenay's model can be achieved by only using symmetric encryption and PUF functions. In our second protocol, we prove that both strong privacy and reader authentication can be achieved in our model (as it was not possible in the Paise Model previously).

Chapter 4

PUF-ENHANCED OFFLINE RFID SECURITY AND PRIVACY

In this chapter, we first revisit Vaudenay's adversary model and extend it to the offline RFID system. We introduce the notion of reader compromise attacks. Then, we define the notion of **privacy+** where compromise attacks on readers are considered. After that we propose a new RFID mutual authentication protocol. In our protocol, we use physically unclonable functions (**PUF**) as unique identity provider mechanisms for the tags. PUF outputs are analogous to the biometric traits in terms of uniqueness. This property provides a secure key derivation for low-cost RFID tags [1]. In our protocol, we use this PUF mechanism to make RFID tags strong against side-channel attacks. Finally, we prove that our protocol provides the narrow destructive privacy for tag owner. Also, we prove that our protocol satisfies narrow destructive **privacy+** in case of compromise reader attacks. To the best our knowledge, our work is the first protocol which uses symmetric operations

and PUF functions and satisfies these privacy properties.

The rest of the chapter is organized as follows: Section 4.1 describes the notations and adversary capabilities of the extended model are described. Section 4.2 describes the proposed authentication protocol. Section 4.3, we present the adversary capabilities and formal security analysis of the protocol. Lastly, in Section 4.4, we give a brief discussion and conclude the chapter.

The results presented in Chapter 4 have been published in [30].

4.1 Extended RFID Security and Privacy Model

In this section, we present an improvement to formal specification of the RFID security and privacy model [18] proposed by Vaudenay in ASIACRYPT 2007. We extend it by introducing notion of compromise of reader attacks and capability of the adversaries.

In our model, an offline RFID system consists of a single operator $\mathcal{I}$, a secure back-end system $\mathcal{DB}$, a set of readers $\mathcal{R}_i$, and a polynomial number of tags $\mathcal{T}$. Each tag $\mathcal{T}$ is assumed to be capable of performing basic cryptographic primitives such as hashing, symmetric encryption, PUF evaluations, and random number generation. On the other hand, each reader $\mathcal{R}_i$ can perform public-key cryptography and can also handle polynomial number of authentication protocols with different tags in parallel. We borrow the definitions of oracles and adversary classes from Vaudenay's model which is explained in Chapter 2.4.4 in detail. We introduce definitions of security and privacy notions for analysis of privacy-preserved offline RFID authentication protocols.

4.1.1 Security, Privacy, and Privacy+

In this chapter, we focus on only security and privacy, so the correctness property is not discussed further. Vaudenay's correctness definition can be combined with the new privacy definition, without compatibility issues. Also, we utilize the tag authentication and privacy definitions of Vaudenay model. For detailed explanations on tag authentication and privacy definition of Vaudenay's model, see Chapter 2.4.4.

For our new privacy definition, contrary to Vaudenay model, similar to RFID tags, the readers can also be corrupted by a malicious adversary because the readers in this context are mobile embedded devices, which have secure discontinuous access to the central database. In our model, we provide a new oracle for strong and destructive adversaries so as to enhance their capabilities.

Corrupt($\mathcal{R}_i$): This oracle enables $\mathcal{A}$ to corrupt reader $\mathcal{R}_i$ and gets all internal states of that reader.

Remark 6. *Once an adversary $\mathcal{A}$ uses (Corrupt($\mathcal{R}_i$)) oracle, $\mathcal{A}$ can interact tag $\mathcal{T}$ after the server's $\mathcal{DB}$ updates other reader's database and one of the updated readers run at least one successful protocol transaction with each tag $\mathcal{T}_i$ used in challenging phase of privacy game.*

Considering compromise of readers, we define a new privacy notion, *privacy+*, for tag owner as follows.

Definition 13. *(Privacy+) An RFID system $\mathcal{S}$ provides **privacy+** notion of $\mathcal{P}$ if $\mathcal{S}$ is still private against an adversary $\mathcal{A}_\mathcal{P}$ even in the case of following conditions:*

- *Some of the readers are corrupted by $\mathcal{A}_P$.*

- *All readers except the corrupted ones are updated by the server.*
- *All tags have at least one successful interaction with one of the updated reader.*

From Definition 13, it is clearly seen that once an adversary corrupts a reader in the system, she captures all the tag related information in the reader's database. Therefore, if the system does not update the remaining readers and the tags do not have successful interactions with one of the updated reader, then the adversary easily impersonates the victim reader and is able to trace any victim tag.

4.2 The PUF Based RFID Authentication Protocol

In this section, we first give the definition of PUF function used in our proposal. Then, we describe the authentication protocol which is composed of three phases; registration, reader update, and authentication phases.

4.2.1 Physically Unclonable Function (PUF)

In this chapter, we use the ideal PUF mechanism, which is described in [1], in our proposed offline-RFID authentication protocol. To the best of our knowledge, such a usage of PUF is the first in the literature.

Definition 14. *Physically Unclonable Function(PUF). Let $k \in \mathbb{N}$ be a security parameter such that $\beta, \theta \in \mathbb{N}$ are polynomially bounded in k. An ideal PUF function is defined as $P : \{0,1\}^{\beta} \rightarrow \{0,1\}^{\theta}$ that holds the following properties:*

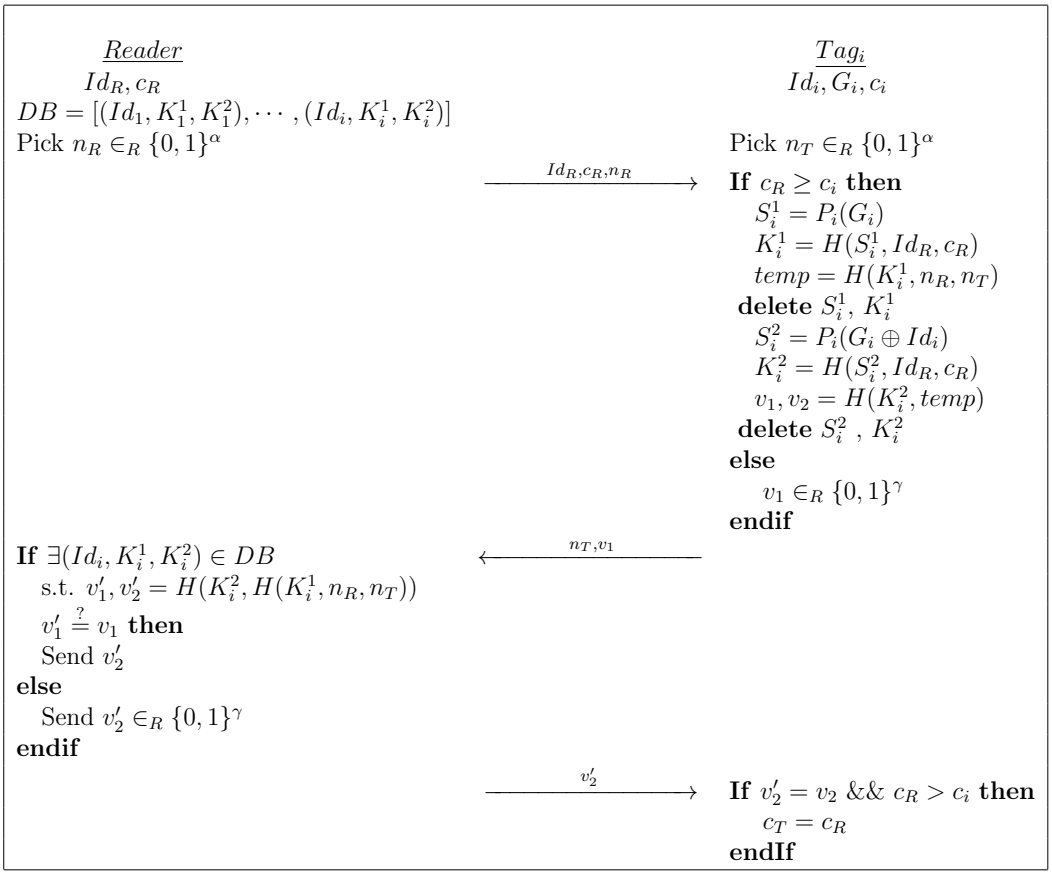

Figure 4.1: The proposed authentication protocol

- *Any physical search trial to investigate the structure of P results in destruction of corresponding P. Namely, after the attack, the tag having this P cannot be evaluated anymore.*

- *Same inputs give same output result. Namely, let $P(a_1) = b_1$ and $P(a_2) = b_2$, if $a_1 = a_2$, then $Prob[b_1 = b_2] = 1$.*

- *Any probabilistic polynomial time adversary can distinguish between output of a P and random value with at most negligible probability.*

As it can be understood from the definition, instead of studying in real PUFs, where for the same inputs they might produce slightly different outputs, we study with an idealized version of PUFs [1, 3, 71] which gives same output results for same inputs.

4.2.2 The Proposed Protocol

In this section, for a complete RFID system, we provide three phases; registration, update reader's database, and authentication.

4.2.2.1 Registration Phase

Initially, in a stable RFID system, counter c_R and c_T are equal to each other. For each tag $\mathcal{T}_i$, Issuer $\mathcal{I}$ first setups $\mathcal{T}_i$ with a random $G_i \in \{0,1\}^\beta$, a unique ID of the tag $\mathcal{T}_i$, Id_i and the counter c_T. Then, $\mathcal{I}$ gets the secrets $S_i^1 \in \{0,1\}^\theta$, $S_i^2 \in \{0,1\}^\theta$ from $\mathcal{T}_i$ PUF evaluations. The record $\{Id_i, S_i^1, S_i^2\}$ is inserted into the central server's database $\mathcal{DB}$. After that, $\mathcal{I}$ setups each reader $\mathcal{R}_j$ in the systems with a unique ID of the reader $\mathcal{R}_j$, Id_j and the counter c_R. Lastly, the server starts secure communication with each reader to update their database. The update mechanism works as explained in the next subsection.

4.2.2.2 Update Reader's Database

The update protocol of reader's database is carried out during the registration phase and whenever a compromised reader is detected. The protocol works as follows. When the server starts a secure communication with the reader R_j, the server first gets Id_R, c_R from the target reader. The c_R is incremented by one. Then, for each tag $\mathcal{T}_i$ in $\mathcal{DB}$, the server computes a new record $\{Id_i, K_i^1, K_i^2\}$ where $K_i^1 = H(S_i^1, Id_R, c_R)$ and $K_i^2 = H(S_i^2, Id_R, c_R)$. Finally, the generated records and the new counter c_R are sent to $\mathcal{R}_j$ in order to update the reader's database and its counter.

4.2.2.3 Authentication

The protocol steps are summarized in Figure 4.1. The detailed protocol steps are described as follows.

As soon as a tag $\mathcal{T}_i$ is in the authentication region, the reader chooses $n_R \in \{0,1\}^\alpha$ and sends it along with its Id_R and c_R to $\mathcal{T}_i$. Then, $\mathcal{T}_i$ first checks whether c_R is greater than or equal to c_i. If condition is not satisfied, $\mathcal{T}_i$ sends random bits to the reader. Otherwise, $\mathcal{T}_i$ generates a random $n_T \in \{0,1\}^\alpha$ and computes the secret value $S_i^1 = P_i(G_i)$. This is where the PUF function is used. Since P_i is specific to $\mathcal{T}_i$ and cannot be cloned, S_i^1 value can only be calculated by that tag. Session key (K_i^1) corresponding to that counter epoch (c_R) is calculated by concatenating S_i^1, Id_R and c_R and then by hashing the result. Then, a temporary hash is computed ($temp = H(K_i^1, n_R, n_T)$) and both secrets S_i^1, K_i^1 are deleted from the volatile memory. After that, the tag computes another pair of secrets by evaluating the function P_i with G_i ($S_i^2 = P_i(G_i \oplus Id_i)$) and a hash ($K_i^2 = H(S_i^2, Id_R, c_R)$). Finally, another hash is calculated over the concatenation of K_i^2 and $temp$ to get the session vectors v_i and v_2. S_i^2 and K_i^2 are both deleted from the memory. The tag

sends n_T and v_1 to the reader.

For each record $\{Id_i, K_i^1, K_i^2\}$ in the reader's database, the reader calculates $v_1', v_2' = H(K_i^2, H(K_i^1, n_R, n_T))$ and compares v_1' to v_1. If a match is found, then she identifies the tag and sends v_2' to $\mathcal{T}_i$. If no match is found in the database, then the reader sends random bits with bit-length of γ to $\mathcal{T}_i$. Finally, $\mathcal{T}_i$ compares v_2' that it has received from the reader to v_2. If they are equal, then the reader is genuine and the tag updates c_i if it is less than c_R. Otherwise, the tag figures out that reader is compromised.

4.3 Security Analysis of the Proposed Scheme

Our proposed protocol uses the PUF mechanism presented in [1]. This mechanism provides a secure key derivation for low-cost RFID tags so that it makes the RFID tags tamper-proof against malicious strong adversaries. We divide this section into two parts. In the first part, we state and prove some lemmas, which describe the capabilities of a strong adversary on PUF circuitry, are used in the proofs of security analysis results. In the second part, we provide security analysis of the protocol.

4.3.1 Security Analysis Tools

The following theorem and the proof are derived from [1].

Theorem 8. *Let S_i^1, S_i^2 be secrets of a tag $\mathcal{T}_i$ for some i in the above-mentioned protocol (see Figure 4.1). Assume that there is an adversary $\mathcal{A}$ with a full side-channel capability on the tag $\mathcal{T}_i$. If P_i is an ideal PUF, then $\mathcal{A}$ can only access either the secret S_i^1 or the secret S_i^2, but not both in $\mathcal{T}_i$.*

Proof. (sketch) The secret G_i and Id_i are fed into the P_i function to compute the real keys S_i^1 and S_i^2. The real keys only appear during the execution of

the protocol. Notice that S_i^1 and S_i^2 never appear in the memory of $\mathcal{T}_i$ at the same time because S_i^1 is first used as an input of a one-way hash function, and then completely erased from the memory. Next, in a similar way, S_i^2 is computed by evaluating $P_i(G_i \oplus Id_i)$ and used in the hash function. Whenever $\mathcal{A}$ applies a side channel attack to $\mathcal{T}_i$, the physical characteristics of P_i will be broken and will no longer be evaluated correctly. If $\mathcal{A}$ applies side-channel attack to extract S_i^1 then the structure of P_i will be destroyed and S_i^2 cannot be computed. Similarly, if $\mathcal{A}$ applies side-channel attack to generated S_i^2 she cannot obtain S_i^1 since it is already erased. Hence, $\mathcal{A}$ can access either S_i^1 or S_i^2 but not both. □

Lemma 3. *Let $\mathcal{A}_d$ be destructive adversary and $\mathcal{T}_i$ be a target tag. During a protocol transcript, the advantage of $\mathcal{A}_d$'s of corrupting $\mathcal{T}_i$ before second deletion (delete S_i^2 , K_i^2) over corrupting $\mathcal{T}_i$ before first deletion (delete S_i^1 , K_i^1) is negligible.*

Proof. Let $\mathcal{A}_d$ corrupt tag $\mathcal{T}_i$ just before the first deletion , then the adversary gets the values of S^1, K^1, n_T, n_R and *temp* of the corresponding protocol run. Then, in order to beat the system in any aspect like security, privacy, the adversary has to find the values of S^2 or K^2. Thus, $\mathcal{A}_d$ has to solve a PUF function output or hash function output. Similarly, let us assume $\mathcal{A}_d$ corrupts the tag just before the second deletion. Then the adversary knows the values of S^2, K^2, n_T, n_R, *temp*, v_1 and v_2 values of the corresponding protocol. Then, in order to beat the system, the adversary has to find the values of S^1 or K^1. Hence, similar to the above deduction, $\mathcal{A}_d$ has to solve a PUF function output or hash function output. Therefore, there is no real advantage difference for the adversary of corrupting a tag before first deletion and the second deletion. □

Lemma 4. *Let $\mathcal{A}_d$ be destructive adversary. Then $\mathcal{A}_d$'s investigating the system with many readers and tags gives him negligible advantage when it is compared with the situation that her investigating the system with just one reader and one tag.*

Proof. Before starting the proof, let us introduce some notations. Let ${}_iv_{1_j}^k$, ${}_iK_k^d$, ${}_in_{e_j}^k$ and ${}_iS^d$ be notations used at protocol description where i is tag index, k is reader index, j is protocol run index, $d \in \{1,2\}$ and $e \in \{R,T\}$. Assume that there are l readers and n tags in the system where l and n are polynomially bounded. Moreover, the number of protocol runs between reader k' and tag i' is ${m_{i'}}^{k'}$ for $k' \in \{1,2,\ldots,l\}$ and $i' \in \{1,2,\ldots,n\}$ before corruption of tag i'. Besides, let the adversary starts a protocol run between reader k' and tag i' ${p_{i'}}^{k'}$ times and starts a protocol run between the tag i' and himself as a replacement of reader k' ${r_{i'}}^{k'}$ times for $k' \in \{1,2,\ldots,l\}$ and $i' \in \{1,2,\ldots,n\}$ before corruption of tag i'. Furthermore, let the adversary starts a protocol run between himself as a replacement of tag i' ${t_{i'}}^{k'}$ times for $k' \in \{1,2,\ldots,l\}$ and $i' \in \{1,2,\ldots,n\}$ before corruption of tag i'. Moreover, let $m = \max_{i',k'}\{{m_{i'}}^{k'}\}$, $p = \max_{i',k'}\{{p_{i'}}^{k'}\}$, $r = \max_{i',k'}\{{r_{i'}}^{k'}\}$, $t = \max_{i',k'}\{{t_{i'}}^{k'}\}$ and let $M = m + p + r + t$. Note that M is polynomially bounded as m, p, r and t values are polynomially bounded. After $\mathcal{A}_d$' observing or corrupting the tags, $\mathcal{A}_d$ has at most $k.M.l$ ${}_iv_{1_j}^k$ values such that ${}_iv_{1_j}^k = MSB_\gamma\{H({}_iK_k^2, H({}_iK_k^1, {}_in_{R_j}^k, {}_in_{T_j}^k))\}$.

By Lemma 3, let us assume that all tags are corrupted before the second deletion. Let us fix tag T_y and reader R_z. In order to prove the lemma, we have the figure out how much advantage $\mathcal{A}_d$ gets to guess the value of ${}_yv_{1_{m_y{}^z+1}}^z$ by observing, creating or corrupting all protocol runs except all protocol runs between (T_y, R_z) pair and T_y and himself as a replacement of R_z and reader R_z and himself as a replacement of T_y. Now, let us take a

pair $(u, w) \neq (y, z)$. There are two cases to consider. First of all, let $u = y$ and $w \neq z$. Then if the adversary finds the value of ${S_u}^1$, then the adversary can calculate the value of ${}_uK_w^1$. Otherwise, the adversary has to find relation the among keys or S values or resulting v_1 values. The maximum success probability is $M(l-1)(\frac{1}{2^{\theta-1}} + \frac{1}{2^{4\gamma}} + \frac{1}{2^{2\gamma}} + \frac{1}{2^{\gamma}})$. Let C denote this probability. As a second case, if $u \neq y$, then $\mathcal{A}_d$ again has to find relation the among keys or S values or resulting v_1 values. However, in this case, the maximum success probability is $Ml(n-1)((\frac{1}{2^{\theta}} + (\frac{1}{2^{2\theta}} + \frac{1}{2^{2\gamma}} + \frac{1}{2^{\gamma}} + \frac{1}{2^{2\gamma}} \max\{\frac{1}{2^{2\gamma}}, \frac{1}{2^{\theta}}\})$ Let D denote this probability and let $\beta = \max\{\theta, \gamma\}$. Then $C + D \leq M(ln-1)\frac{1}{2^{\beta-2}}$. Since n, l and M are polynomially bounded and the value of β is sufficiently large, the maximum total advantage is negligible. □

In the next section, these theorems and lemmas will be used in the proof of security and privacy analysis of the proposed protocol.

4.3.2 Security and Privacy Analysis

In this section, we first prove that our protocol achieves tag authentication and destructive privacy. Then, we also prove that our protocol satisfies reader authentication and destructive *privacy+*.

Theorem 9. *The RFID protocol demonstrated in Figure 4.1 achieves tag authentication if H is a hash function (Definition 3).*

Proof. Assume to the contrary, the protocol described in Figure 4.1 does not achieve tag authentication. That means, the adversary $\mathcal{A}_s$ behaves like a legitimate tag to a legitimate reader with non-negligible probability. By Lemma 4, let us assume that there are only one legitimate reader R and one tag T in the system and for simplicity, R is not updated throughout the proof. By the argument above, the strong adversary $\mathcal{A}_s$ does not need

to apply CreateTag,DrawTag andFree oracles. Let $\mathcal{A}_s$ observe protocol runs between the reader and the tag m times. Moreover, let $\mathcal{A}_s$ uses SendReader(π) oracle p times to start protocol run between the reader and the tag and uses SendTag oracle r times to start protocol run between himself and the tag. Here, the values of m, p and r are polynomially bounded. Note that, $\mathcal{A}_s$ can use Corrupt oracle at most one time as the tag T has PUF function inside. However, we assume that $\mathcal{A}_s$ applies this oracle exactly one time as this assumption increases his chances to win the game.

Let the adversary have chance to impersonate the corresponding tag k times, where k is polynomially bounded. In order to achieve the impersonation, at each round $\mathcal{A}_s$ creates u_i triple $(S^2, K^2, v_{1_i})_j$, where $i \in \{1, \ldots, k\}$, $j \in \{1, \ldots, u_i\}$ and each u_i is polynomially bounded. Note that, if the space of PUF is smaller than the space of hash function, these triples are created on guesses of $\mathcal{A}_s$ on the values of S^2s. Otherwise, they are created on guesses of $\mathcal{A}_s$ on the values of K^2s. Since the hash function is pre-image resistant, guesses are not made on the third component. The adversary checks whether they are true or not at each triple at each impersonation trial based on the protocol transcripts that have been reached so far. If the adversary could not find any match at the end of calculations, then the adversary just guesses the value of v_{1_i}.

Let $M = m + p + r$ and $U = \max\{u_1, u_2, \ldots, u_k\}$ and so M and U are polynomially bounded. Moreover, let $\beta = \max\{\theta, 2\gamma\}$. By Lemma 4, let us assume that corruption made before the first deletion. Note that, if the value of n_R sent by the reader at each impersonation trial is one of those n_R values which is used at previous protocol runs, then the success probability of destroying tag authentication is 1 by choosing corresponding n_T value. However, the probability of realization of this scenario is at most $1-(1-\frac{M}{2^\alpha})^k$.

Otherwise, the probability of $\mathcal{A}_s$'s generating correct value of v_1 in at least one impersonation trial is at most $2 - \prod_{j=0}^{kU-1}(1 - \frac{1}{2^{\beta}-j}) + (1 - \frac{1}{2^{\gamma}})^k$. In order to see the total probability is minimum, let us use $ln(1-x) \approx -x$ for small x values. Then the success probability is at most $3 - e^{-\frac{Mk}{2^{\alpha}}} - e^{-\frac{kU}{2^{\beta}-kU}} - e^{-\frac{k}{2^{\gamma}}}$. By contradiction assumption, this probability is non-negligible, so at least one of the values of M, U and k is non-negligible. However, this contradicts with the fact that M, U and k are polynomially bounded. □

Theorem 10. *The RFID protocol demonstrated in Figure 4.1 achieves destructive privacy if the protocol achieves tag authentication, P is a PUF (Definition 14) and H is hash function(Definition 3).*

Proof. Assume to the contrary, the system does not achieve destructive privacy property. That means, there is a destructive adversary $\mathcal{A}_d$, who can distinguish between the real RFID system and the system which is simulated by a blinder $\mathcal{B}$ with non-negligible probability. Note that, $\mathcal{B}$ simulates LAUNCH, SENDTAG, SENDREADER and *Result* oracles without knowing the tag and the reader secrets.

More formally, let there exists an oracle $\mathcal{O}^{dest}$ such that $\mathcal{A}_d$ plays the following game with this oracle. $\mathcal{O}^{dest}$ chooses a number $b \in_R \{0,1\}$, if $b = 1$, real RFID system is used, otherwise $\mathcal{B}$ simulates the system. $\mathcal{A}_d$ watches the system for polynomially bounded number of times and the adversary is allowed to use corrupt oracle as well. At the end, $\mathcal{A}_d$ guesses a number b'. If $|Prob(b = b')| = \frac{1}{2} + a$ where a is non-negligible, $\mathcal{A}_d$ wins the game, else the adversary loses. Note that, by contradiction assumption, $\mathcal{A}_d$ wins the game.

Let us start with how $\mathcal{B}$ evaluates oracles:

- LAUNCH(): Evaluated in a trivial way.

- SendTag($Id_R, c_R, n_R, vtag$): The output is $n_T \in_R \{0,1\}^\alpha$, $v_1 \in_R \{0,1\}^\gamma$.
- SendReader(π): The output is $n_R \in_R \{0,1\}^\alpha$ and the the real values of Id_R and c_R.
- SendReader((n_T, v_1),π): The output is $v_2 \in_R \{0,1\}^\gamma$.
- SendTag(v_2): returns no output.
- Result(π): If π is generated by Launch oracle and the protocol transcript is generated by SendTag and SendReader oracles, the output is 1. If one of the conditions does not hold, then the output is 0.

By Lemma 4, let us assume that there are only one legitimate reader R and one tag T in the system and for simplicity, R is not updated throughout the proof. Let the system be run n_1 times only by real RFID system or the blinder according to b value the oracle $\mathcal{O}^{dest}$ chooses and let at n_1-th run, $\mathcal{A}_d$ applies Corrupt oracle to the tag T. By Lemma 3, let us assume that corruption is applied before second deletion. Thus, $\mathcal{A}_d$ have the knowledge of $\{({n_R}^1, {n_T}^1, {v_1}^1, {v_2}^1), ({n_R}^2, {n_T}^2, {v_1}^2, {v_2}^2), \ldots, ({n_R}^{n_1}, {n_T}^{n_1}, {v_1}^{n_1}, {v_2}^{n_1})\}$ and S^2, K^2, $temp^{n_1}$.

There are five cases to consider. First two cases are $\mathcal{A}_d$'s determining the value of S^1 or K^1. The probability of these happening are $\frac{1}{2^\theta}$ and $\frac{1}{2^{2\gamma}}$, respectively. The third case is $\mathcal{A}_d$'s determining value of temp at least one protocol run. The probability of this case is $1-(1-\frac{1}{2^{2\gamma}})^{n_1}$. The fourth possibility is $\mathcal{A}_d$'s determining value of v_1 at least one protocol run. The probability of this case is $1-(1-\frac{1}{2^{\gamma}})^{n_1}$. The last case is $\mathcal{A}_d$'s determining value of v_2 being random.

By contradiction assumption, as $\mathcal{A}_d$ wins the game against the oracle, then one of the four probabilities above is non-negligible or realization of the last case is non-negligible. However, with sufficiently large θ and γ values, the four possibilities listed above are negligible. Thus, by assumption, the probability of $\mathcal{A}_d$'s determining v_2 value being random is non-negligible. However, this statement contradicts with Theorem 9, i.e. contradiction to tag authentication. Thus, proposed protocol satisfies destructive privacy property. □

Theorem 11. *The RFID protocol demonstrated in Figure 4.1 achieves reader authentication if H is a hash function (Definition 3).*

Proof. By Lemma 4, without loss of generality, there are one reader R and one tag T in the system. Let $\mathcal{A}_d$ observe previous p run of tag T and R before $\mathcal{A}_d$ starts a protocol run with T. As a result of the observations, $\mathcal{A}_d$ gets the following protocol transcripts $(n_{R_1}, Id_R, c_R, n_{T_1}, v_{1_1}, v_{2_1}), \ldots, (n_{R_p}, Id_R, c_R, n_{T_p}, v_{1_p}, v_{2_p})$. Note that, $\mathcal{A}_d$'s aim is to impersonate the reader R by convincing T. The most logical move for $\mathcal{A}_d$ is choosing one of the values of $n_{R_1}, n_{R_2}, \ldots, n_{R_p}$ as n_R value. W.l.o.g., let $\mathcal{A}_d$ sends n_{R_1}, Id_R, c_R to tag T. There are two cases to consider. First of all, if T responds with n_{T_1}, v_{1_1}, then the probability that the adversary returns the correct value of v_2 is 1. If this is not the case, then there are two cases, which are $\mathcal{A}_d$'s calculating the value of v_2 or guess the value of v_2. For the first case, $\mathcal{A}_d$ has to now at least one of the values of (S^1, S^2), (S^1, K^2), (K^1, S^2) and (K^1, K^2). The corresponding probabilities are $\frac{1}{2^{2\theta}}$, $\frac{1}{2^{\theta+2\gamma}}$, $\frac{1}{2^{\theta+2\gamma}}$, $\frac{1}{2^{4\gamma}}$. Let $q = \max\{\frac{1}{2^{2\theta}}, \frac{1}{2^{\theta+2\gamma}}, \frac{1}{2^{4\gamma}}\}$. For the second case, $\mathcal{A}_d$ guess the value of v_2 with possibility $\frac{1}{2^{\gamma}}$. Thus, the probability that $\mathcal{A}_d$'s convincing the tag T is $\frac{1}{2^{\alpha}} + \frac{2^{\alpha}-1}{2^{\alpha}} \max\{m, \frac{1}{2^{\gamma}}\}$. Note that the probability given above negligible provided that α, γ and θ are large enough. □

Theorem 12. *The RFID protocol illustrated in Figure 4.1 provides destructive* ***privacy+*** *if the protocol achieves tag authentication, P is a PUF (Definition 14) and H is hash function(Definition 3).*

Proof. Assume that a reader R_C is compromised. Then the adversary $\mathcal{A}_{R_C}$ gets the information $(Id_1, {K_1}^1, {K_2}^1, \ldots, Id_n, {K_n}^1, {K_n}^1)$ of tags T_i for $i = 1, 2, \ldots, n$, where n is polynomially bounded. Due to the assumption at Remark 6, after DB updates all other reader, the value of c_R changes. Moreover, as the value of c_R changed, then the values of ${K_i}^1 = H({S_i}^1, Id_R, c_R)$ and ${K_i}^2 = H({S_i}^2, Id_R, c_R)$ for $i = 1, \ldots, n$ are changed. Note that, the adversary $\mathcal{A}_{R_C}$ does not have the values of ${S_i}^1, {S_i}^2$ for $i = 1, \ldots, n$ due to the pre-image resistance property of hash function. Thus, from previous knowledge of $(Id_1, {K_1}^1, {K_2}^1, \ldots, Id_n, {K_n}^1, {K_n}^1)$, $\mathcal{A}_{R_C}$ cannot calculate new ${K_i}^1, {K_i}^2$ values for $i = 1, \ldots, n$. Therefore, the only legitimate information that $\mathcal{A}_{R_C}$ has after system re-setup is Id's of all tags. Therefore, by Theorem 9 and Theorem 10, the system is private against the adversary $\mathcal{A}_{R_C}$. Hence, the RFID system provides destructive **privacy+**. □

4.3.3 Security & Privacy and Performance Comparisons

Considering memory storage for tag identifiers or keys and other information, our protocol requires 3β-bit (Id, G, and c) memory in tag side where β is at most the length of a hash output. Contrary to tags, server has no limited resource, so we do not concern on the server-side memory usage. In terms of computational cost, our protocol requires at most four hash computation and two PUF evaluations overhead at the tag side. On the other hand, the computational complexity at the server side is at most $\mathcal{O}(n)$, where n is the

number of tags in the system.

Table 4.1 summarizes the comparison of our protocol with other protocols, where n is the number of tags in the system. Our protocol and [39] have reader authentication whereas only our protocol provides destructive privacy, and destructive *privacy+*. While considering computational complexity at the server side, the complexity required for each scheme is roughly proportional to the number of tags in the system.

	[106]	[39]	Our Protocol
Reader Authentication	+	+	+
Privacy	-	Weak	Destructive
Privacy+	-	Weak+	Destructive+
Crypto Primitive	Hash	Hash	Hash & PUF
Reader Complexity	$\mathcal{O}(n)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$

Table 4.1: The security, privacy and performance comparisons

4.4 The Summary of the Chapter

In this chapter, we first extend Vaudenay's adversarial model [18] for offline RFID system and introduce the notion of compromise reader attacks. We define the notion of *privacy+* and the game behind this privacy notion. Then, we propose an RFID mutual authentication protocol based on PUF functions. We prove that our protocol achieves destructive privacy for tag owner. To the best our knowledge, it is the first protocol which uses only symmetric cryptographic primitives and PUF functions and provides destructive *privacy+* even in case of compromising reader attacks. Our protocol can be efficiently implemented in low-cost RFID tags because the tags need only low cost cryptographic primitives such as hash and PUF functions.

Chapter 5

A QUADRATIC RESIDUE BASED RFID AUTHENTICATION PROTOCOL

Designing authentication protocols without lowering security and privacy levels negatively affects the efficiency of the entire system. In addition, achieving the security and privacy properties, the complexity in tag and server side can vary dramatically from one protocol to another. Hence, while handling security and privacy issues, it is also important to realize them with less computational complexity in the server and tag side.

In order to resolve these security and privacy issues, numerous RFID authentication protocols have been recently proposed in the literature [10–16]. Many of them failed to provide security and privacy and the computation on the server side is also very high. Recently, Yeh et al.proposed an improvement of the RFID authentication protocol [107] which utilizes quadratic residue

for security and privacy [31]. It requires constant time at the server side for identification; however, this proposal has lack of a formal security and privacy analysis.

In this chapter, we first present a security analysis of Yeh et al. authentication protocol according to Vaudenay's model. We prove that this protocol satisfies at most destructive privacy; but, the tag and reader authentication are secure against at most weak adversary. Then, we propose a unilateral authentication protocol which achieves narrow strong privacy. After that, we propose an enhanced version of proposed protocol, which satisfies mutual authentication with reader authentication against stronger adversaries. It achieves destructive privacy according to Vaudenay's model. Note that, our proposed protocol and enhanced version of it has constant-time complexity to identify and authenticate a tag.

The outline of the chapter is as follows. In Section 5.1, we give a brief discussion on formal model on the security. Section 5.2 describes Yeh et al.'s proposed protocol and gives its security and privacy analysis. In Section 5.3, the first proposed protocol with security and privacy analysis is given in a detail. In Section 5.4, analysis of our second mutual protocol is given in a detail. In Section 5.5, we conclude the chapter.

The results presented in Chapter 5 have been published in [32, 33].

5.1 Formal Tools for Security and Privacy Analysis

We divide this section into three parts. In the first part, preliminaries and notations are described. After that, we summarize Vaudenay's privacy model. Finally, we give brief information about ProVerif which is a tool used in

security analysis.

5.1.1 Vaudenay's privacy model

In order to analyze the protocols in this chapter, we use Vaudenay's privacy model [18]. Vaudenay's model is discussed in detail in Chapter 2.4.4. Note that, in this chapter, in all protocol descriptions, tags only include T_{ID} as a tag related information. Hence, when RESULT oracle is applied, for the current protocol run, the notion of privacy is meaningless. Thus,we look for privacy for protocol runs where CORRUPT oracle does not take place. As a reference, following remark can be given.

Remark 7. *In this chapter, the adversary is not allowed to distinguish between the real system and the blinder at protocol runs where* CORRUPT *oracle takes place.*

5.1.2 Security Analysis

Securing a system is a complex problem since it requires a careful analysis of the underlying assumptions about cryptographic functions and trusted parties, and an accurate implementation of hardware and software. Satisfying all these requirements is virtually impossible without the use of formal analytical techniques [108] which are invaluable tools for identifying weaknesses in security protocols.

In order to verify formally whether an authentication protocol achieves a certain security property, we first create a model which specifies the capability of an adversary. Then, we describe the interactions of the adversary in this model and the definition of the security property within the model. Finally, by using this model, a formal tool checks whether the goals in the security

protocol are achieved or not. Recently, several different symbolic formal models have been proposed in the literature [109–111]. In our analysis, we use ProVerif [109] which is automatic tool to verify a wide range of security of cryptographic protocols.

In order to describe an authentication protocol and its interactions, we used the applied pi-calculus [112]. The grammar used in the applied pi-calculus is described below, where M and N are terms, n is a name, x is a variable and u stands either for a name or for a variable.

$P, Q, R, ::=$	
0	*null process*
$P \mid Q$	*parallel composition*
$!P$	*replication*
$vn.P$	*name restriction*
$let\ x = M\ in\ P\ else\ Q$	*term evaluation*
$if\ M = N\ then\ P\ else\ Q$	*conditional*
$in(u, x).P$	*message input*
$out(u, N).P$	*message output*

Properties of the processes described in the applied pi-calculus can be proved by automated tools ProVerif [113]. ProVerif first translates the applied pi-calculus process into a set of Horn clauses. These clauses account for the initial knowledge of the attacker and the inference rules she can apply to broaden her knowledge pool for the messages. ProVerif can prove

reach-ability properties that are typical of model checking tools such as correspondence assertions, and observational equivalence. ProVerif can also reconstruct an execution trace that falsifies the desired property: when a desired property cannot be proved. Furthermore, in ProVerif analysis, protocol analysis is considered in accordance with an infinite number of sessions, an unbounded message space and parallel sessions.

5.2 Yeh et al.'s Proposed Protocol and Its Privacy Analysis

In this section, we first present Yeh et al.'s authentication protocol [31] by considering the server and the reader as a single entity, just reader, since the channel between these two entities is assumed to be secure. Then, we analyze the protocol according to Vaudenay's privacy model. We prove that this protocol satisfies destructive privacy. The protocol steps are described as follows.

Let $h : \{0,1\}^* \rightarrow \{0,1\}^\alpha$ be a hash function and $PRNG : \{0,1\}^\alpha \rightarrow \{0,1\}^\alpha$ be a pseudo-random number generator. Let $r, s, t, n \in \{0,1\}^\alpha$. Each tag $\mathcal{T}$ is equipped with a unique $\mathcal{T}_{ID}$ and stores the value n and r. These values are given by reader in the initialization phase. Reader stores the values $h(\mathcal{T}_{ID}), \mathcal{T}_{ID}, r, r_{old}$ where $r_{old} = r$ at the beginning.

In the protocol, the reader $\mathcal{R}$ first sends a random challenge $s \in_R \{0,1\}^\alpha$ to a tag $\mathcal{T}$. Once $\mathcal{T}$ receives the challenge, $\mathcal{T}$ picks another random challenge $t \in_R \{0,1\}^\alpha$. $\mathcal{T}$ constructs x, y, X, R and T as follows: $x = h(TID) \oplus r \oplus s \oplus t$, $y = r \oplus t$, $X = x^2 \mod n$, $R = (r^2 \mod n) \oplus t$, $T = t^2 \mod n$. After these calculations, the tag sends $X, R, T, h(x), h(y)$ and $h(t)$ to $\mathcal{R}$. Then, $\mathcal{R}$ gets (x_1, x_2, x_3, x_4) and (t_1, t_2, t_3, t_4) by solving $X = x^2 \mod n$ and $T = t^2$

mod n by using the factors of n, which are p and q. After that $\mathcal{R}$, determines correct values of x and t by comparing $h(x_i) \stackrel{?}{=} h(x)$ and $h(t_i) \stackrel{?}{=} h(t)$. Then, $\mathcal{R}$ determines the correct value of r in a similar way. $\mathcal{R}$ computes $h(\mathcal{T}_{ID})$ and seeks $\mathcal{T}_{ID}$ from database and compares received r with r or r_{old}. If received r is valid, then computes acknowledgment message $x_{ack} = \mathcal{T}_{ID} \oplus t \oplus r$ or r_{old}, sends $h(x_{ack})$ to $\mathcal{T}$ and updates r_{old} as r as $PRNG(r)$. Then $\mathcal{T}$ checks whether $h(x_{ack}) \stackrel{?}{=} h(\mathcal{T}_{ID}) \oplus r \oplus t$. If it is valid, $\mathcal{T}$ updates r as $PRNG(r)$, otherwise the protocol aborts.

Before starting the security and privacy analysis of the protocol, we can assume, without loss of generality, there are one reader and one tag in the system since the variables which change tag to tag at calculation steps are $h(T_{ID})$ and r which have same bit length as s. Thus, by deriving more s values, i.e. more protocol runs, we can recover the advantage loss due to working with one tag instead of many tags.

Theorem 13. *Yeh et al.'s Proposed Protocol achieves tag authentication and reader authentication if the adversary $\mathcal{A}_w$ belongs to weak class.*

Proof. Let the adversary $\mathcal{A}_w$ observe ℓ protocol runs between the reader and the tag. Let us assume that $\mathcal{A}_w$ tries to impersonate the tag at $\ell + 1^{th}$ run. If the value of s sent by the reader is equal to the one of the s values sent at one of the previous protocol runs, $\mathcal{A}_w$ impersonates the tag with success probability 1. Otherwise, $\mathcal{A}_w$ has to guess the values of $h(T_{ID})$ and r for corresponding run correctly. Thus, the success probability for $\mathcal{A}_w$ to impersonate the tag is $\frac{\ell}{2^\alpha} + (1 - \frac{\ell}{2^\alpha})\frac{1}{2^{2m}}$, which is negligible. Hence, the system achieves tag authentication if the adversary is weak.

Similarly, if $\mathcal{A}_w$ tries to impersonate the reader, then $\mathcal{A}_w$ sends a challenge s to the tag. Upon receiving the challenge, the tag responses with $X, R, T, h(x), h(y), h(t)$ according to which t value the tag chooses. However,

Tag $\mathcal{T}_{ID}$ $n, r, \mathcal{T}_{ID}$		**Reader** $\mathcal{R}$ $p, q, n, \mathcal{T}_{ID}, r, r_{old}$
$t \in_R \{0,1\}^\alpha$ $x = h(\mathcal{T}_{ID}) \oplus r \oplus s \oplus t$ $y = r \oplus t$ $X = x^2 \mod n$ $R = (r^2 \mod n) \oplus t$ $T = t^2 \mod n$	$\xleftarrow{(1)\ hello,s}$ $\xrightarrow{(2) X,R,T,h(x),h(y),h(t)}$	$s \in_R \{0,1\}^\alpha$ 1. Solves $X = x^2 \mod n$ and $T = t^2 \mod n$ to get (x_1, x_2, x_3, x_4) and (t_1, t_2, t_3, t_4) 2. Compares $h(x_i) \stackrel{?}{=} h(x)$ and $h(t_i) \stackrel{?}{=} h(t)$ to determine x and t 3. Solves $R = (r^2 \mod n) \oplus t$ to get (r_1, r_2, r_3, r_4) 4. Compares $h(r_i \oplus t) \stackrel{?}{=} h(y)$ to determine r 5. Computes $h(\mathcal{T}_{ID}) = x \oplus r \oplus s \oplus t$ 6. Seeks $\mathcal{T}_{ID}$ record using $h(\mathcal{T}_{ID})$ then compares received $r \stackrel{?}{=} r$ or r_{old} else abort 7. Prepares ACK message, $x_{ack} = \mathcal{T}_{ID} \oplus t \oplus r$ or r_{old}
1. Check $h(x_{ack}) \stackrel{?}{=} h(\mathcal{T}_{ID}) \oplus r \oplus t$ if not abort 2. Updates r as $PRNG(r)$	$\xleftarrow{(3)\ h(x_{ack})}$	8. Updates r_{old} as r as $PRNG(r)$

Figure 5.1: T.-C. Yeh et al.'s improved scheme

as $\mathcal{A}_w$ does not know the value of r, $\mathcal{A}_w$ can not figure out the value of t. Moreover, since $\mathcal{A}_w$ does not know the factors of n, which are p and q, $\mathcal{A}_w$ can not the roots of X and R and T. Besides, $\mathcal{A}_w$ has to guess correct value of T_{ID}. Thus, the probability that $\mathcal{A}_w$ sends correct $h(x_{ack})$ to the tag is $\frac{1}{2^{2m}}$, which is negligible. Therefore, the system achieves the reader authentication if the adversary is in class of weak. □

Theorem 14. *Yeh et al.'s proposed protocol achieves destructive privacy but does not achieve narrow strong privacy.*

Proof. Let there are one reader and one tag in the system and let $\mathcal{A}_d$ be a destructive adversary. Assume to the contrary, the protocol does not achieve destructive privacy. That is, the adversary $\mathcal{A}_d$ can distinguish between the real RFID system and the system simulated by the $\mathcal{B}$ with non-negligible probability.

Let us start with how $\mathcal{B}$ evaluates oracles:

- LAUNCH(): Evaluated in a trivial way.
- SENDREADER(π): The output is $s \in_R \{0,1\}^{\alpha}$.
- SENDTAG(s, π): The output is $X, R, T, h(x), h(y), h(t)$.
- SENDREADER($(X, R, T, h(x), h(y), h(t))$,$\pi$): The output is $h(x_{ack})$.
- RESULT(π): This oracle works as defined in Remark 2

Let the system is run ℓ times only by the real RFID system or $\mathcal{B}$ and let $\mathcal{A}_d$ applies CORRUPT oracle at $\ell + 1^{th}$ protocol run. $\mathcal{A}_d$ gets the values of T_{ID}, ℓ and $r_{\ell+1}$, $t_{\ell+1}$, $x_{\ell+1}$, $y_{\ell+1}$ as a result of CORRUPT oracle usage.

There are three ways for $\mathcal{A}_d$ to distinguish between the real reader from the blinder. The first way is $\mathcal{A}_d$'s guessing the correct value of r at any

protocol run. If this is the case, then by using the relation $R = (r^2 mod n) \oplus t$ formula, $\mathcal{A}_d$ gets the value of t for the corresponding round. Moreover, $\mathcal{A}_d$ gets the values of x, y, X, T values of the corresponding round. Furthermore, as $\mathcal{A}_d$ can calculate next rounds' r value, in a similar way $\mathcal{A}_d$ gets the values of t, x,y, X, T values for each advancing protocol run. Therefore, if $\mathcal{A}_d$ correctly guesses r value at least 1 protocol run, then $\mathcal{A}_d$ can check correctness of the protocol at next protocol runs. Therefore, in this case, the adversary distinguishes the real system from the blinder. However, realization of this case has probability at most $1 - (1 - \frac{1}{2^\alpha})$, which is negligible. The next way for $\mathcal{A}_d$ is to guess the correct value of $h(ack)$ at any protocol run. Similarly, the realization of this case has probability at most $1 - (1 - \frac{1}{2^\alpha})$, which is negligible.

The last way is $\mathcal{A}_d$'s determining the value that is produced by *Result* oracle is right or wrong. By contradiction assumption, $\mathcal{A}_d$'s success probability at this case is non-negligible as the success probability of previous two ways are negligible. However, this contradicts with the Theorem 13 as in our case, for past protocol runs, destructive adversary acts like weak adversary as r values of previous protocol runs can not be deduced from the knowledge of $r_{\ell+1}$. Thus, the protocol achieves destructive privacy.

Let $\mathcal{A}_s$ be a narrow strong adversary. In this case, let $\mathcal{A}_s$ corrupts the tag before starting any protocol run. As indicated above, $\mathcal{A}_s$ gets the value of r, and due to the nature of PRNG functions, $\mathcal{A}_s$ can calculate the value of r in any advancing run. Therefore, she can calculate the value of t, x, y, X and T at each protocol run. Hence, $\mathcal{A}_s$ can distinguish the real system from the blinder. Thus, the protocol does not achieve narrow strong privacy. □

Tag $\mathcal{T}_{ID}$		**Reader** $\mathcal{R}$
$n, \mathcal{T}_{ID}$		$p, q, n, \mathcal{T}_{ID}$
$t \in_R \{0,1\}^\alpha$ $x = h(\mathcal{T}_{ID}) \oplus s \oplus h(t) \oplus t$ $X = x^2 \mod n$, $T = t^2 \mod n$	$\xleftarrow{(1)\ s}$	$s \in_R \{0,1\}^\alpha$
$M = h(x\|\|t)$	$\xrightarrow{(2)\ X,T,M}$	1. Solves $X = x^2 \mod n$ and $T = t^2 \mod n$ to get (x_1, x_2, x_3, x_4) and (t_1, t_2, t_3, t_4) 2. Compares $h(x_i\|\|t_j) \stackrel{?}{=} M$ to determine x and t 3. Computes $h(\mathcal{T}_{ID}) = x \oplus s \oplus h(t) \oplus t$ 4. Check $h(\mathcal{T}_{ID})$ exists in database

Figure 5.2: Our proposed narrow strong private scheme

5.3 The Proposed Protocol

In this section, we first present a novel scalable RFID authentication protocol which is based on quadratic residue. Then, we give security and privacy analysis of it according to Vaudenay's model.

Let $h : \{0,1\}^* \rightarrow \{0,1\}^\kappa$ be a hash function. Let $s, n, t \in \{0,1\}^\alpha$. Each tag $\mathcal{T}$ is equipped with a unique secret $\mathcal{T}_{ID}$ and stores the value n. These values are given by reader $\mathcal{R}$ in the initialization phase. $\mathcal{R}$ stores the values $h(\mathcal{T}_{ID})$ and $\mathcal{T}_{ID}$. The authentication protocol is summarized in Figure 5.2.

In the protocol, $\mathcal{R}$ first sends a random challenge $s \in_R \{0,1\}^\alpha$ to a tag $\mathcal{T}$. Once $\mathcal{T}$ receives the challenge, $\mathcal{T}$ picks another random challenge $t \in_R \{0,1\}^\alpha$. $\mathcal{T}$ constructs x, X, T and M respectively as shown in Fig. 2, then sends X, T and M to $\mathcal{R}$. Once $\mathcal{R}$ receives X, T and M, it gets (x_1, x_2, x_3, x_4) and (t_1, t_2, t_3, t_4) by solving $X = x^2 \mod n$ and $T = t^2 \mod n$ by the help of factors on n. After that $\mathcal{R}$, determines correct values of x and t by comparing $h(x_i||t_j) \stackrel{?}{=} M$. Now, $\mathcal{R}$ can compute $h(\mathcal{T}_{ID})$ and then check existence of $\mathcal{T}_{ID}$ in the database.

5.3.1 Security and Privacy Analysis

Before starting the security analysis of the proposed protocol, Note that, we can assume there is one reader and one tag in the system. Since the variables which change tag to tag at calculation steps are $h(TID)$ which has same bit length as s. Thus, by deriving more s values, i.e. more protocol runs, we can recover the advantage loss due to working with one tag instead of many tags.

Theorem 15. *The proposed RFID protocol achieves tag authentication if the adversary $\mathcal{A}_w$ belongs to the weak class.*

Proof. Let the adversary $\mathcal{A}_w$ observe ℓ protocol runs between the reader and the tag. First of all, let us assume that $\mathcal{A}_w$ tries to impersonate the tag at $\ell + 1^{th}$ run. There are two cases to consider. If the challenge value s sent by the reader is equal to the one of the s values sent at previous protocol run, then with 1 success probability, $\mathcal{A}_w$ impersonates the tag. However, the probability of realization of this scenario is $\frac{\ell}{2^\alpha}$. If this is not the case, then the only way for $\mathcal{A}_w$ to impersonate the tag is to guess the value of $h(T_{ID})$ correctly. The success probability in this case $\frac{1}{2^\alpha}$. Hence, $\mathcal{A}_w$ impersonates the tag with probability $\frac{\ell}{2^\alpha} + (1 - \frac{\ell}{2^\alpha})\frac{1}{2^\alpha}$, which is negligible. Therefore, the system achieves tag authentication if the adversary is weak. □

Theorem 16. *The proposed RFID protocol achieves narrow strong privacy.*

Proof. Before starting the proof steps, note that, for proposed protocol, in terms of privacy analysis, there is no real difference between the adversary's applying CORRUPT oracle only one time and more than one time. Since, at each CORRUPT oracle usage, the adversary gets the values of T_{ID} and n, which do not changes among protocol runs and session specific t and x values and there is no real connection between any of two protocol runs' corresponding values. Therefore, in the proof, the adversary applies the CORRUPT oracle only once.

Let there are one reader and one tag in the system and let $\mathcal{A}_s$ be a narrow strong adversary. Assume to the contrary, the protocol does not achieve narrow strong privacy. That is, the adversary $\mathcal{A}_s$ can distinguish between the real RFID system and the system simulated by the $\mathcal{B}$ with non-negligible probability.

Let us start with how $\mathcal{B}$ evaluates oracles:

- LAUNCH(): Evaluated in a trivial way.

- SendReader(π): The output is $s \in_R \{0,1\}^m$.
- SendTag(s, π): The output is X, T, M.

Let the system is run ℓ times only by the real RFID system or $\mathcal{B}$. Let $\mathcal{A}_s$ applies Corrupt oracle at $\ell+1$st protocol runs and after that oracle usage, the system run k more times. Note that, $\mathcal{A}_s$ gets the values of T_{ID}, n, $t_{\ell+1}$ and $x_{\ell+1}$ as a result of Corrupt oracle usage.

Note that, there are two ways for $\mathcal{A}_w$ to distinguish the real system from the blinder. The first one is to guess t value correctly at any of previous n protocol runs or next k runs. The other way is to guess one of the X, T and M value correctly. Hence, the total success probability of the adversary is $\frac{\ell+k}{2^\alpha} + (1 - \frac{\ell+k}{2^\alpha})\frac{3}{2^\alpha}$, which is negligible. Of course, one can run this process defined above polynomially bounded time and increase the adversary's chance but the resulting success probability will be at most negligible. □

5.4 An Enhanced Version of the Proposed Protocol

In this section, we propose an enhanced version of the proposed protocol which provides mutual authentication. We prove that our protocol depicted in Figure 5.3 satisfies reader authentication against strong adversary and has destructive privacy level.

The protocol steps of this protocol consists of the unilateral authentication protocol and the last message sent by reader to the tag. The reader prepares $x_{ack} = TID||t||s$ and sends $h(x_{ack})$ to the tag. The tag checks validity of $h(x_{ack})$ by comparing its value with $h(TID||t||s)$. All the steps of the second protocol are summarized in Figure 5.3.

5.4.1 Security and Privacy Analysis

Theorem 17. *The protocol depicted in Figure 5.3 satisfies tag authentication against weak adversary and satisfies reader authentication against narrow strong adversary.*

Proof. First of all, note that by Theorem 15, the protocol satisfies tag authentication against weak adversary. Let us prove the reader authentication part. Let the adversary $\mathcal{A}_s$ be a narrow strong adversary and $\mathcal{A}_s$ observes n protocol run between the reader and the tag. Let us assume that $\mathcal{A}_s$ corrupts the tag at $\ell + 1^{th}$ round and tries to impersonate the reader at $\ell + 2^{th}$ run. Note that, $\mathcal{A}_s$ gets the value of TID and $t_{\ell+1}$ as a result of CORRUPT oracle usage. Let us do our analysis in the worst case such that in the first $\ell + 1$ protocol runs, the reader sends the same s value to the tag as a challenge. $\mathcal{A}_s$ sends the same s value to the tag as a challenge so as to increase his chance to impersonate the reader. There are two cases to consider. The first case is tag's choosing t among previous chosen t values. In this case, the adversary impersonates the reader with 1 possibility. If this is not the case, adversary has to guess the correct value of t chosen by the tag to create $h(x_{ack})$. Therefore, $\mathcal{A}_s$ impersonates the reader with probability at most $\frac{\ell+1}{2^\alpha} + (1 - \frac{\ell+1}{2^\alpha})\frac{1}{2^\alpha - \ell - 1}$, which is negligible.

Note that, one can give more impersonation chance to the adversary and increases his chance to impersonate the reader. However, at the end, the success probability remains negligible. □

Theorem 18. *The protocol demonstrated at Figure 5.3 achieves destructive privacy.*

Tag $\mathcal{T}_{ID}$ $n, \mathcal{T}_{ID}$		**Reader** $\mathcal{R}$ $p, q, n, \mathcal{T}_{ID}$
$t \in_R \{0,1\}^\alpha$ $x = h(\mathcal{T}_{ID}) \oplus s \oplus h(t) \oplus t$ $X = x^2 \mod n$ $T = t^2 \mod n$	$\xleftarrow{(1)\ hello,s}$	$s \in_R \{0,1\}^\alpha$
$M = h(x\|\|t)$	$\xrightarrow{(2)\ X,T,M}$	1. Solves $X = x^2 \mod n$ and $T = t^2 \mod n$ to get (x_1, x_2, x_3, x_4) and (t_1, t_2, t_3, t_4) 2. Compares $h(x_i\|\|t_j) \stackrel{?}{=} M$ to determine x and t 3. Computes $h(\mathcal{T}_{ID}) = x \oplus s \oplus h(t) \oplus t$ 4. Check $h(\mathcal{T}_{ID})$ exists in database -if exists prepares ACK message, $x_{ack} = \mathcal{T}_{ID}\|\|s\|\|t$ -otherwise
Check $h(x_{ack}) \stackrel{?}{=} h(\mathcal{T}_{ID}\|\|s\|\|t)$	$\xleftarrow{(3)\ h(x_{ack})}$	prepares ACK message, $x_{ack} \in \{0,1\}^*$

Figure 5.3: Enhanced version of proposed protocol

Proof. Let there are one reader and one tag in the system and let $\mathcal{A}_d$ be a destructive adversary. Assume to the contrary, the protocol does not achieve destructive privacy. That is, the adversary $\mathcal{A}_d$ can distinguish between the real RFID system and the system simulated by the $\mathcal{B}$ with non-negligible probability.

$\mathcal{B}$ evaluates oracles in the same way as indicated at the proof of Theorem 16 with addition:

- **SendReader**$((X, T, M), \pi)$: The output is $h(x_{ack})$.
- **Result**(π): This oracle works as defined in Remark 2

Let the system is run ℓ times only by the real RFID system or $\mathcal{B}$ and let $\mathcal{A}_d$ applies CORRUPT oracle at $\ell + 1$st protocol run. $\mathcal{A}_d$ gets the values of TID, n and $t_{\ell+1}$, $x_{\ell+1}$ as a result of CORRUPT oracle usage.

There are three cases to consider. The first case is $\mathcal{A}_d$'s guessing the value of t in any of previous ℓ protocol runs. However, as there is no connection between $t_{\ell+1}$ and previously chosen t values, the realization of first case is negligible. The second case is $\mathcal{A}_d$'s guessing the correct value of $h(x_{ack})$. Similarly, the probability of realization of this case is negligible.

The last way is $\mathcal{A}_d$'s determining the value that is produced by *Result* oracle is right or wrong. By contradiction assumption, $\mathcal{A}_d$'s success probability at this case is non-negligible as the success probability of previous two ways are negligible. However, this contradicts with the Theorem 17 as in our case, for past protocol runs, destructive adversary acts like weak adversary. Thus, the protocol achieves destructive privacy. □

5.4.2 Formal Analysis

In this section, we use ProVerif tool in order to formally prove the security property of our enhanced protocols such as reader authentication and tag authentication.

To encode the protocol into the pi-calculus, we first determine the required cryptographic primitives with function symbols, and rewrite rules and equations over terms. Let $hash()$ be a universal hash function. Let xor be the function which satisfies $\forall x, y \in \{0,1\}^{\alpha}$, $xor(x, y) = x \oplus y$. Note that, ProVerif cannot evaluate XOR functions properly and so we provide all possible reduction functions $(xor_1, \ldots, xor_8)$ which help ProVerif to simulate XOR function. Let two large primes, (P,Q) be a factors of a common modulus N. Then, let *smodulus* denote a type of pair of (P,Q) and *pmodulus* denote a type of public modulus (N=PQ). The reader stores factors of a public modulus N *P_and_Q* and tag stores the modulus, *publicmod(P_and_Q)*.

We also simulate quadratic residue functions, one for taking modulo square, one for taking modulo square root. $\forall x, X \in \{0,1\}^{\alpha}$ and *pmodulus* $N \in \{0,1\}^{\alpha}$, $square(x, N)$ is equal to $x^2 \mod N$ and $ssquare(X, N)$ gives all possible solutions to $X^{-2} \mod N$.

The public channel between reader and tags are described as $free\ c$: *channel*. The adversary is also allowed to use this channel for her attack.

Our mutual authentication protocol is expected to satisfy (informally) the following properties:

- Authentication of tag to reader: if the reader identifies tag, it responds so that at the end of the protocol, tag has approval to engage with reader in a session, only if reader permits it.

- Authentication of reader to tag: similar to the above.

- Secrecy of session keys (combination of s and t).

In our model, we assume *secret* is a private key shared between tag and reader which is unknown by the adversary. Our interest in this model is to verify the secrecy of the bitstring (t) generated by tag. Therefore, as soon as tag authenticates reader, tag broadcasts *secret* XORed with the generated t ($out(c, secret \oplus t)$). If there is no way that an adversary can derive *secret* by applying the rules, then the protocol is safe. Namely, the authentication procedure has not been compromised. In order to challenge the adversary, we write the query syntax, as the following: **query attacker(secret)**.

The behavior of the reader is encoded into following process, *Reader*. In this process, the reader waits any message from tag on channel $in(c : channel, data)$. It sends any message to tag through the same channel ($out(c : channel, data)$).

```
1.  let Reader(TID:bitstring ) = new s:bitstring;
2.  (* Message 1 *) out(c, s);
3.  (* Message 2 *)
4.  in(c, (X:bitstring, T:bitstring, M:bitstring));
5.  let x = ssquare(X,P_and_Q) in
6.  let t = ssquare(T,P_and_Q) in
7.  let (=M) = hash((x,t)) in
8.  let HTID = hash(TID) in let HT = hash(t) in
9.  let (=HTID) = xor1(xor1(xor1(x,HT),t),s)
10. in event readerAuthTag(s,t);(* Message 3 *)
11. out(c, hash((TID,s,t))); 0.
```

The behavior of the tag is encoded into following process:

12. let **Tag**(TID:bitstring, N : pmodulus) =
13. (* Message 1 *)
14. in(c, s:bitstring); new t:bitstring ;
15. let HT = hash(t) in let HTID = hash(TID) in
16. let x = ssquare(X,P_and_Q) in
17. let X = square(x,N) in let T = square(t,N) in
18. let M = hash((x,t)) in
19. (* Message 2 *) out(c,(X,T,M)); (* Message 3 *)
20. in(c, ack:bitstring);
21. let (=ack) = hash((TID,s,t)) in
22. event tagAuthReader(s,t);
23. out(c, xor(secret,t)) ;0.

These two processes are executed multiple times in parallel using the following syntax:

24. **process**
25. let N = publicmod(P_and_Q) in out (c,N);
26. new TID:bitstring;
27. (!Reader(TID) | !Tag(TID,N) | phase 1; out(c,TID))

In this process, we first created a public modulus N, which is sent through channel c. Then we create a new TID for a tag identifier. This TID and the private products of N (P_and_Q) are given to reader. ProVerif first converts these processes and adversary actions into a set of Horn clauses [114] so as to automatically prove queries. Then, it runs the processes and searches for a valid security gap based on requested queries. The output of ProVerif confirms that the attacker cannot derive the term (*secret*) so the authentication procedure can be performed successfully without being compromised. Also, the attacker is not be able to cheat both reader and tag even if we provide TID of the victim tag to adversary in phase 1.

5.5 The Summary of the Chapter

Nowadays, several RFID applications have been deployed in our daily lives such as contact-less credit cards, e-passports, ticketing systems, and etc. The importance security and privacy concerns has been gradually increasing for RFID systems.

In this chapter, we first give a formal security and privacy analysis of Yeh et al.'s authentication protocol. We proved that this protocol provides at most destructive privacy according to Vaudenay' model whereas the tag and reader authentication is secure against at most weak adversary. Then, we introduced an unilateral authentication protocol and we formally proved that this protocol achieves narrow strong adversary. We also proposed the enhanced version of the protocol that provides reader authentication. We proved that the second protocol satisfies destructive privacy and the reader authentication is secure against narrow strong adversary.

Chapter 6

OPTIMAL SECURITY LIMITS OF RFID K-PCD PROTOCOLS

In this chapter, we focus on the low-cost distance-bounding protocols that have bit-wise fast phase and no final signature. As for the classification, we introduce the notion of k-previous challenge dependent (k-PCD) protocols where each response bit depends on the current and the k previous challenges. We call the 0-PCD protocols as current challenge dependent protocols (CCD). Then, we provide trade-off curves between the optimal security limits of mafia and distance frauds for CCD protocols and k-PCD protocols. After that, we give the security analysis of k-PCD distance bounding protocols and show the success probabilities against mafia and distance fraud attacks. Our results show that when we increase the number 'k', the security level of distance bounding protocols enhanced as expected. We also demonstrate the results calculated via developed computer program and observe k-PCD trade-off curve. We show that the curve for k-PCD protocols is below the

trade-off cure for $k-1$-PCD protocols for all $k \geq 1$. Finally, we give a simple and generic method of extending CCD protocols to k-PCD protocols.

The composition of the chapter is following. Section 6.1 gives general notations and definitions. In Section 6.2 we briefly explain the current challenge dependent (CCD) protocols and give its security limits. In Section 6.3, conjectures and open questions of k-PCD protocols and some relevant definitions are given. Section 6.4 introduce the way of constructing k-PCD protocols and gives their security levels.

The results presented in Chapter 6 was published in [34] and have been submitted to a Journal [35].

6.1 General Notions, Definitions

Distance bounding protocols are generally composed of two types of phases: slow phase and fast phase. In some protocols there is only one slow phase at the beginning of the protocol [78, 95], on the other hand some other protocols [79, 92] composed of three phases; slow phase-I, fast phase, and slow phase-II. The slow phases consist of the time-consuming operations such as random nonce generations, commitment and signature calculations. The fast phase includes non-time consuming response generations and rapid bit exchanges. Particularly during the slow phase-II the prover has to calculate a final signature. In the slow phase, both parties constitute the *session secrets* (for example, the session secret in the HK protocol presented in Figure 6.1 consists of two registers) that are used to produce response bits during the fast phase. Throughout the fast phase, both parties use the same *response generating function* which produces a response by using the session secrets and given a challenge value. In this chapter, we mainly focus on the distance

bounding protocols in which there is no final signature and having bit-wise fast phase.

In what follows we study on how to achieve the optimum security against mafia fraud and distance fraud. For that, we first define a class of protocols without a final signature and, in which each response bit depends on the current challenge, Current Challenge-Dependent (CCD) protocols. It is defined as follows.

Definition 15 (CCD Protocol). *Let $f : \mathbb{F}_2^{m+1} \rightarrow \mathbb{F}_2$ be a Boolean function. A CCD protocol $\mathcal{P}$ is a distance bounding protocol that satisfies the following properties:*

- *During the fast phase, each response bit r_i is computed as $r_i := f(c_i, y_0^i, \ldots, y_{m-1}^i)$, where c_i is the i-th challenge bit and $(y_0^i, \ldots, y_{m-1}^i)$ is the i-th string of the session secret shared by both prover and verifier for $i = 1, \ldots, n$, where n is the number of rapid bit exchanges.*
- *There is no final slow phase.*

The protocol $\mathcal{P}$ is denoted as $f(c_i, y_0^i, \ldots, y_{m-1}^i) \rightarrow r_i$ CCD protocol. The function f is called the response function of the protocol $\mathcal{P}$.

One popular example of CCD protocols is Hancke and Kuhn (HK) protocol [78]. The protocol consists of two phases: *Slow phase* and *fast phase* (or rapid bit exchange phase). As depicted in Figure 6.1 the protocol steps are as follows.

- **Slow phase -** The prover and the verifier exchange their randomly generated nonce. From these random numbers and a shared secret x both party compute two $n - bit$ registers y_0 and y_1, using a pseudo-random function h. These registers are used as session secrets during the fast phase.

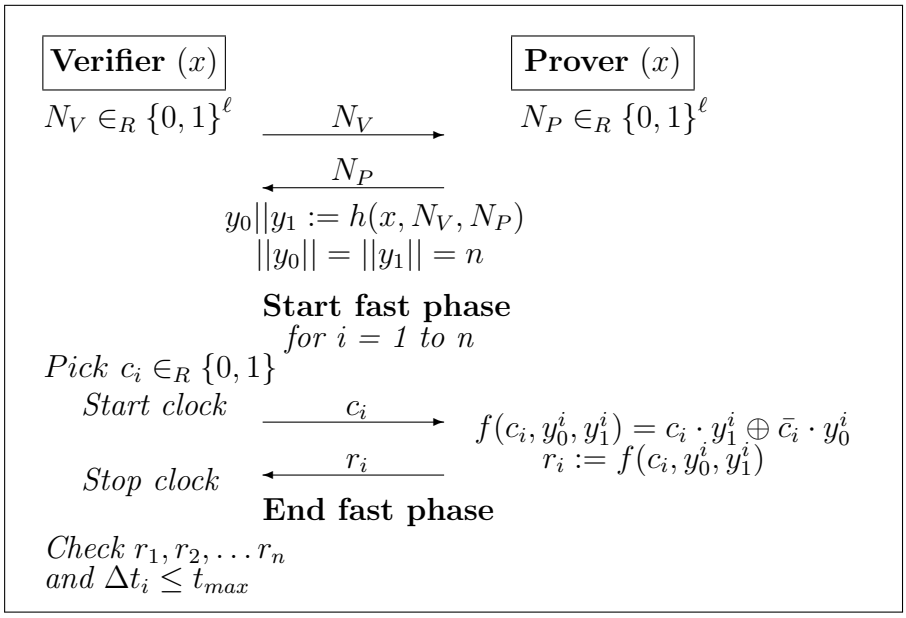

Figure 6.1: Hancke and Kuhn's distance bounding protocol

- **Fast phase -** The verifier sends a random challenge c_i to the prover, then the later replies with r_i, by using the challenge and shared session secrets such that $f(c_i, y_0^i, y_1^i) = y_{c_i}^i$, where $i = 1, 2 \ldots n$. For each rapid bit exchange the verifier measures the round trip time Δt_i. After n rapid bit exchanges the verifier checks the correctness of r_i's and $\Delta t_i \leq t_{max}$ where n is the security parameter and t_{max} is the maximum allowed time delay for each rapid bit exchange.

The response function of the protocol can be described as the following Boolean function:

$$f(c_i, y_0^i, y_1^i) = c_i \cdot y_1^i \oplus (\bar{c}_i) \cdot y_0^i = y_{c_i}^i \tag{6.1}$$

where $\oplus$ and $\cdot$ are the addition and the multiplication operations of the binary Galois Field respectively.

Let us denote P_{maf}^E the success probability of correctly guessing one bit

response for mafia fraud of an attack E, and similarly P_{dis}^{E} for distance fraud of an attack E. The security levels of a given protocol $\mathcal{P}$ are defined as follows.

Definition 16. *$P_{maf}(\mathcal{P}) = \max_E P_{maf}^{E}$ and $P_{dis}(\mathcal{P}) = \max_E P_{dis}^{E}$. That is, $P_{maf}(\mathcal{P})$ is the maximum of P_{maf}^{E} over all the mafia fraud attacks E mounted on $\mathcal{P}$, and similarly $P_{dis}(\mathcal{P})$ is the maximum of P_{dis}^{E} over all the distance fraud attacks E mounted on $\mathcal{P}$.*

The success probability of mafia and distance fraud against HK protocol is $(3/4)^n$ for the attacks given in [34, 78]. Therefore, $P_{maf}(HK) \geq 3/4$ and $P_{dis}(HK) \geq 3/4$. It has been an open question that these security levels are optimum for CCD protocols. Also, it is not known whether it is possible to improve the security level against mafia fraud without sacrificing the security level against the distance fraud and vice-versa. In general, we have the following open questions for CCD protocols:

- What is the best security levels for both mafia fraud and distance fraud among all CCD protocols?
- What is the optimum achievable security level for mafia fraud of a CCD protocol?
- For a CCD protocol, what is the minimum value of P_{maf} if P_{dis} is ideal (i.e. $\frac{1}{2}$)?

The above-mentioned questions are answered in [34]. We show that there is a trade-off between mafia fraud and distance fraud, namely $P_{maf}(\mathcal{P}) + P_{dis}(\mathcal{P}) \geq 3/2$. It is also proven that for any CCD protocol there is a security limit concerning the mafia fraud such that $P_{maf}(\mathcal{P}) \geq 3/4$ for any CCD

protocol $\mathcal{P}$. As a consequence of this result it is shown that if $P_{dis}(\mathcal{P}) = 1/2$ then the protocol is completely vulnerable to mafia fraud (i.e., $P_{maf}(\mathcal{P}) = 1$).

In order to improve the security levels against these frauds without using a final signature, the notion of *k-Previous Challenge Dependent* (*k-PCD*) protocols is introduced, in which each response bit depends on the current and the k previous challenges during fast phase. The definition of the *k-PCD* protocol as follows.

Definition 17 (**k-PCD Protocol**). *Let $g : \mathbb{F}_2^{m+k+1} \to \mathbb{F}_2$ be a Boolean function. A k-PCD protocol $\mathcal{P}$ is a distance bounding protocol that satisfies following properties*

- *During the fast phase, each response bit r_i is computed as $r_i := g(c_i, \ldots c_{i-k}, y_0^i, \ldots, y_{m-1}^i)$ where c_j is the j-th challenge bit and $(y_0^i, \ldots, y_{m-1}^i)$ is the i-th string of the session secret shared by both prover and verifier for $i = 1, \ldots, n$, where n is the number of rapid bit exchanges.*
- *There is no final slow phase.*

The protocol $\mathcal{P}$ is denoted as $g(c_i, \ldots, c_{i-k}, y_0^i, \ldots, y_{m-1}^i) \to r_i$ k-PCD protocol. The function g is called the response function of the protocol $\mathcal{P}$.

Remark 8. *From Definitions (1) and (2), a CCD protocol is a class of k-PCD protocol where $k = 0$.*

6.2 Optimal Security Limits for CCD Protocols

In this section, we demonstrate the security trade-off between mafia and distance frauds for CCD distance bounding protocols. We use the characteristics of the response function f used in a CCD protocol, during the security

analysis against mafia and distance frauds. We suppose that all the challenges and the shared session secrets, which are used to produce response bits, are uniformly random. Let m be the security parameter. For a given response function f, let us define the following sets:

$$\begin{aligned} \mathcal{A} &= \{y = (y_0, y_1, \ldots, y_{m-1}) \in \mathbb{F}_2^m : \\ &\quad f(0, y_0, \ldots, y_{m-1}) \neq f(1, y_0, \ldots, y_{m-1})\} \\ \mathcal{B} &= \{y = (y_0, y_1, \ldots, y_{m-1}) \in \mathbb{F}_2^m : \\ &\quad f(0, y_0, \ldots, y_{m-1}) = f(1, y_0, \ldots, y_{m-1})\} \end{aligned} \tag{6.2}$$

Let a and b be the cardinality of the sets $\mathcal{A}$ and $\mathcal{B}$, respectively. Then, it clearly holds $a + b = 2^m$. We also define a generic distance fraud attack that can be mounted on all CCD protocols and this attacks is depicted in Algorithm 6.1.

Algorithm 6.1 A generic distance fraud attack for CCD Protocol (n)

Require: n: Number of rounds
  for $i \leftarrow 1$ **to** n **do**
    $t \leftarrow f(0, y_0^i, \ldots, y_{m-1}^i) + f(1, y_0^i, \ldots, y_{m-1}^i)$
    if $t = 0$ **then**
      Send 0
    else if t=2 **then**
      Send 1
    else
      Send a random bit
    end if
  end for

We also describe a generic mafia fraud attack that can be mounted on all CCD protocols. In this attack, the adversary first relays the messages (e.g nonce or commitments etc.) between the verifier and the prover, during the

slow phase. Then, during the fast phase she executes Algorithm 6.2. We assume that, the protocol is public. Therefore, a and b can be computed during the off-line phase.

The following statement gives a trade-off between mafia fraud and distance fraud for CCD protocols.

Theorem 19. *Let $\mathcal{P}$ be a $f(c_i, y_0^i, \ldots, y_{m-1}^i) \rightarrow r_i$ CCD protocol. Assume that c_i and y_j^is used during the fast phase of $\mathcal{P}$ are uniformly random. Then, (i) $P_{maf}(\mathcal{P}) \geq 3/4$, and (ii) $P_{maf}(\mathcal{P}) + P_{dis}(\mathcal{P}) \geq 3/2$.*

Algorithm 6.2 A generic mafia fraud attack for CCD protocol (n,a,b)

Require: n: Number of rounds
Require: flip: Deciding on flipping the response
  if $b \leq a$ **then**
    flip$\leftarrow 1$
  else
    flip$\leftarrow 0$
  end if
  for $i \leftarrow 1$ **to** n **do**
    Send a random challenge $c_i' \in \{0,1\}$
    Record the prover's response r_i'
  end for
  */*Then, Mafia continues the protocol with the verifier*/*
  for $i \leftarrow 1$ **to** n **do**
    record i-*th* challenge of the verifier in c_i
    if $c_i' = c_i$ **then**
      Send r_i'
    else
      Send $r_i' \oplus flip$
    end if
  end for

Proof. Let us first consider the distance fraud attack described in Algorithm 6.1. For any challenge c_i, the adversary always produces a correct response if $y_0^i, y_1^i, \ldots, y_{m-1}^i$ are in the set $\mathcal{B}$. Otherwise, i.e., when they are in the set $\mathcal{A}$,

she successfully predicts the response with a probability of 1/2 because c_i, and y_j^i s are uniformly random. Thus, the success probability of P_{dis} for the attack given in Algorithm 6.1 is equal to $\frac{b}{2^m} \cdot 1 + \frac{a}{2^m} \cdot \frac{1}{2} = \frac{a+2b}{2^{m+1}} = \frac{1}{2} + \frac{b}{2^{m+1}}$.

Concerning the mafia fraud attack given in Algorithm 6.2, let the adversary receive the r'_i responses from the prover for her predicted challenges c'_i. Then, she executes the attack against the verifier. Since c_is are randomly produced by the verifier, there are two equally likely cases. (a) If $c_i = c'_i$ the adversary knows the answer then sends r'_i. (b) If $c_i \neq c'_i$ she has to predict the response bit r_i. The probability that r'_i and r_i are equal is $\frac{b}{2^m}$, and that are not equal is $\frac{a}{2^m}$. The adversary chooses the larger probability in order to decide whether she flips the response bit (i.e., $r'_i \oplus 1$). Then, we have $P_{maf} = \frac{1}{2} \cdot 1 + \frac{1}{2} \cdot \max\{\frac{a}{2^m}, \frac{b}{2^m}\}$. Since $a + b = 2^m$, $\max\{\frac{a}{2^m}, \frac{b}{2^m}\} \geq \frac{1}{2}$ and this implies that $P_{maf} \geq \frac{3}{4}$.

If $b \leq 2^{m-1}$ ($b \leq a$), then, $P_{maf} = \frac{1}{2} + \frac{a}{2^{m+1}}$ for the attack. So, we have $P_{dis} + P_{maf} = \frac{3}{2}$. On the other hand, when $b \geq 2^{m-1}$ ($b \geq a$), $P_{maf} = \frac{1}{2} + \frac{b}{2^{m+1}} \geq \frac{3}{4}$. Thus, $P_{dis}(\mathcal{P}) + P_{maf}(\mathcal{P}) \geq \frac{3}{2}$. □

The first part of Theorem 19 indicates that there is a security limit for CCD protocols concerning the mafia fraud, and the second part attests the security trade-off between mafia and distance frauds. Figure 6.2 depicts the *trade-off curve* between the success probabilities of these frauds for any CCD protocol.

One interesting result of Theorem 19 is that CCD protocols cannot attain the ideal security level against the distance fraud without being vulnerable against mafia fraud. This is also stated in Corollary 1.

Corollary 1. *For a CCD protocol $\mathcal{P}$, if the security level for the distance fraud is ideal (i.e. $P_{dis}(\mathcal{P}) = 1/2$) then, $P_{maf}(\mathcal{P})$ is 1.*

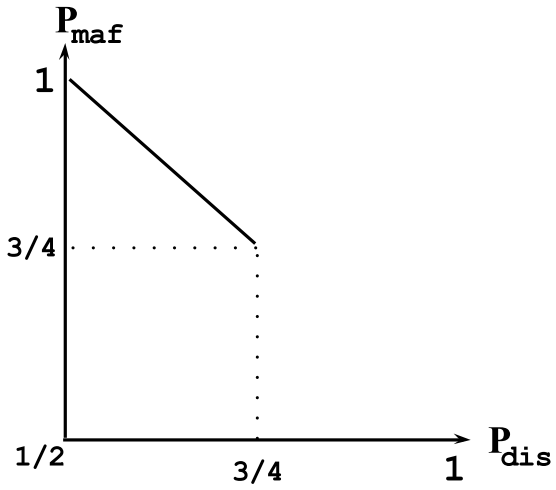

Figure 6.2: The trade-off between distance and mafia for CCD protocols

Proof. The probability $P_{dis}(\mathcal{P})$ satisfies the condition in Theorem 1, so $P_{maf}(\mathcal{P}) = 3/2 - 1/2 \geq 1$. □

Remark 9. *Recall that the security levels of the HK protocol against the mafia and distance frauds are both* $3/4$*. Security levels of HK protocol lie on the trade-off curve.*

6.3 Optimal Security Limits for k-PCD Protocols

In this section, we give the security analysis of k-PCD distance bounding protocols. In this respect, we first describe the concept of neighborhood, which is helpful for security analysis of the distance fraud. Then, we introduce two generic attacks for mafia and distance frauds that can be mounted on all k-PCD protocols.

While designing k-PCD distance bounding protocol, there are n-round one-bit challenge/response during fast phase. There is an exceptional case

for the first round of this phase. In the first round, the verifier sends k initial challenges before sending a challenge c_1. For example, in the first round of a k-PCD protocol, the verifier first sends $c_{-k+1}, \ldots, c_{-1}, c_0$ and c_1 then waits for r_1.

6.3.1 Security Regions for Distance Fraud

Let us consider an adversary who tries to cheat on the distance against a verifier. While producing a response bit r_i, the adversary may use some of the received previous challenges in her attack. This can increase the success probability of the adversary. However, reception of the challenges at earlier time depends on how far the adversary is away from the verifier. Therefore, in order to make the attack analysis simpler, we describe $k+2$ spherical regions $(Z_0, \ldots, Z_{k+1})$, in which the adversary can communicate with the verifier (see Figure 6.3.1). Let d_0 be the maximum radius of Z_0 that is the legal authentication region, and t_0 be the elapsed time for a signal to travel the distance d_0. Z_i is the annulus region between two concentric spheres with radius of d_{i-1} and $d_{i-1} + d_i$ where $d_i = (i+1) \cdot d_0$ and $0 \leq i \leq k$. Z_i is the outside of Z_{i-1}. We assume that the speed of the signal is constant.

When the adversary is in the region Z_0, she always accesses to all the challenges and produces valid responses on time. However, when the distance between the adversary and the verifier is $d_0 + \delta_d$ ($\delta_d > 0$), any signal traveling this distance takes $t_0' > t_0$, i.e., $t_0' = t_0 + \delta_t$. In order to run her attack successfully, the adversary should send each current response (r_i), at least $2\delta_t$ before receiving the current challenge (c_i). When $\delta_t > k \cdot t_0$, she is in region Z_{k+1}, she should send the response r_i before receiving $c_{i-k}, c_{i-k+1}, \ldots, c_i$. However, when the adversary is in Z_u, where $u < k+1$, she accesses some of the previous challenges to send r_i. This increases the attacker's success

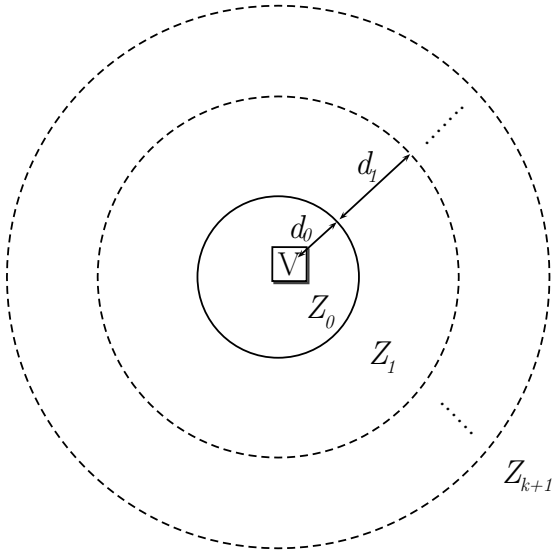

Figure 6.3: Regions for distance fraud

probability. As a result, while analyzing the security of a k-PCD protocol against distance fraud, the region of the adversary should be considered.

In the next subsection, we focus on the security of *k-PCD* protocols against mafia and distance frauds for arbitrary number of k values. For the sake of simplicity for distance fraud analysis, we assume that the adversary is in Z_{k+1}.

6.3.2 Security Trade-off for k-PCD Protocols

Let f be the function that outputs the response bit r_i from the challenges $c_{i-k}, c_{i-k+1}, \ldots, c_i$ and the precomputed session secrets $y_i^0, y_i^1, ..., y_i^{m-1}$. The function f is executed n times to form the whole set of responses. For $y = (y_0, y_1, ..., y_{m-1}) \in \mathbb{F}_2^m$, let α_y be

$$\alpha_y = \sum_{c_{i-k}, c_{i-k+1}, ..., c_i \in \{0,1\}} g(c_{i-k}, c_{i-k+1}, \ldots, c_i, y) - 2^k.$$

Also, we define the following sets:

$$\begin{aligned}\mathcal{A} &= \{y \in \mathbb{F}_2^m : |\alpha_y| = 2^k\},\\ \mathcal{B} &= \{y \in \mathbb{F}_2^m : 0 < |\alpha_y| < 2^k\},\\ \mathcal{C} &= \{y \in \mathbb{F}_2^m : \ \alpha_y = 0\},\end{aligned}$$

where $|\cdot|$ denotes the absolute value.

The set $\mathcal{A}$ includes the session secrets that produce the same response bit for any $c_{i-k}, c_{i-k+1}, \ldots, c_i$. The set $\mathcal{B}$ consists of the session secrets that produce the responses, majority of them are equal, for any $c_{i-k}, c_{i-k+1}, \ldots, c_i$. The set $\mathcal{C}$ contains the session secrets that produce the responses, half of them are equal, for any $c_{i-k}, c_{i-k+1}, \ldots, c_i$.

Let a, b and c be the cardinality of the sets $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$, respectively. Then $a+b+c = 2^m$. We assume that all the challenges and the precomputed session secret bits, which are used to compute response bits, are uniformly random.

Theorem 20. *Let $\mathcal{P}$ be a $f(c_{i-k}, c_{i-k+1}, \ldots, c_i, y_0^i, \ldots, y_{m-1}^i) \rightarrow r_i$ k-PCD protocol. Assume that c_i's and y_j^i's used in the fast phase of the protocol $\mathcal{P}$ are uniformly random. Then $P_{maf}(P) \geq 1/2 + 1/2^{k+2}$ and $P_{maf}(P) + P_{dis}(P) \geq 1 + 1/2^{k+2}$.*

Proof. Considering distance fraud attack depicted in Algorithm 6.3, for any challenge value, the adversary always guesses a correct response if y^i is in the set A. If it is in the set B, she predicts the response with probability between $1/2 + \frac{2^{k+1}-1}{2^{k+1}}$ and $1/2 + \frac{2^k+1}{2^{k+1}}$ by choosing the frequent one. However, if it is in the set C, she can predict the response with probability $1/2$. Therefore,

Algorithm 6.3 A generic distance fraud attack for k-PCD protocol (n)

Require: n: Number of rounds

$c_p \leftarrow \{0,1\}$
for $i \leftarrow 1$ **to** n **do**
  if $\alpha_{y^i} \geq 2^{k-1}$ **then**
    Send 1
    if $f(c_{i-k},\ldots,c_{i-1},0,y_0^i,\ldots,y_{m-1}^i) = 1$ **then**
      $c_p \leftarrow 0$
    else
      $c_p \leftarrow 1$
    end if
  else
    Send 0
    if $f(c_{i-k},\ldots,c_{i-1},0,y_0^i,\ldots,y_{m-1}^i) = 0$ **then**
      $c_p \leftarrow 0$
    else
      $c_p \leftarrow 1$
    end if
  end if
  $c_{i-t-1} \leftarrow c_{i-t}$ for $t = 0,\ldots,k-1$
  $c_i \leftarrow c_p$
end for

the success probability P_{dis} for this attack is computed as follows:

$$\begin{aligned} P_{dis} &\geq \frac{a}{2^m} \cdot 1 + \frac{b}{2^m} \cdot \frac{2^k + 1}{2^{k+1}} + \frac{c}{2^m} \cdot \frac{1}{2} \\ &\geq \frac{1}{2} \cdot \left(\frac{a}{2^m} + \frac{b}{2^m} + \frac{c}{2^m} \right) + \frac{1}{2} \cdot \frac{a}{2^m} + \frac{1}{2^{k+1}} \cdot \frac{b}{2^m} \\ &\geq \frac{1}{2}. \end{aligned}$$

Considering the mafia fraud attack described in Algorithm 6.4. After the first *k-1* queries, the adversary carries out the attack against the verifier. The adversary knows the correct response (i.e., $r'_i = r_i$) if $c'_{i-k} = c_{i-k}$, $c'_{i-k+1} = c_{i-k+1}$, 1... and $c'_i = c_i$. The probability of this event is $1/2^{k+1}$ since all the challenge bits are produced uniformly random. For the remaining cases, the adversary has to predict the corresponding response bit r_i.

The attacker has to predict the response bit r_i corresponding to a different challenge bits $(c_{i-k}, c_{i-k+1}, \ldots, c_i)$. If the corresponding session secret y_i is in the set A, then the probability that $r'_i = r_i$ is 1 by definition. The probability of the prediction will be between $1/2 + \frac{2^{k+1}-1}{2^{k+1}}$ and $1/2 + \frac{2^k+1}{2^{k+1}}$ if y_i is in the set B since this happens only if both the input vectors $(c_{i-k}, c_{i-k+1}, \ldots, c_i, y_i)$ and $(c'_{i-k}, c'_{i-k+1}, \ldots, c'_i, y_i)$ produce the same response even though the vectors are not equal. Similarly, the probability is $\frac{2^k-1}{2^{k+1}-1}$ if y_i is in the set C. Then, the probabilities that $r'_i \neq r_i$ are deduced straightforward.

The attacker has two strategies for predicting a response value corresponding to a different pair of challenge bits.

(i) Attacker sends the same response value received from the prover (r'_i) and the success probability of mafia fraud (P_{maf}^{noflip}) is computed as follows.

Algorithm 6.4 A generic mafia fraud attack for k-PCD protocol(n,a,c)

Require: n: Number of rounds
flip: Deciding on flipping the response
Send a random challenge $c'_0, \ldots, c'_{k-1} \in \{0,1\}$
if $c \geq (2^{k+1}-1) \cdot a$ **then**
flip$\leftarrow 1$
else
flip$\leftarrow 0$
end if
for $i \leftarrow k$ **to** n **do**
Send a random challenge $c'_i \in \{0,1\}$
Record the prover's response r'_i
end for
*/*Then, Mafia continues the protocol with the verifier*/*
Record first k challenge of the verifier
for $i \leftarrow k$ **to** n **do**
record i-th challenge of the verifier in c_i
if $c'_i = c_i$, $c'_{i-1} = c_{i-1}, \ldots,$ $c'_{i-k} = c_{i-k}$ **then**
Send r'_i
else
Send $r'_i \oplus flip$
end if
$c_{i-t-1} \leftarrow c_{i-t}$ for $t = 0, \ldots, k-1$
end for

$$P_{maf}^{no-flip} = \frac{1}{2^{k+1}} + \frac{2^{k+1}-1}{2^{k+1}} \cdot (\frac{a}{2^m} \cdot 1 + \frac{b}{2^m} \cdot P_b^{no-flip} + \frac{c}{2^m} \cdot \frac{2^k - 1}{2^{k+1} - 1})$$

(ii) Attacker sends the complement of the response value and the success probability of mafia fraud with this strategy is computed as follows.

$$P_{maf}^{flip} = \frac{1}{2^{k+1}} + \frac{2^{k+1}-1}{2^{k+1}} \cdot (\frac{a}{2^m} \cdot 0 + \frac{b}{2^m} \cdot (1 - P_b^{no-flip}) + \frac{c}{2^m} \cdot \frac{2^k}{2^{k+1} - 1})$$

Both P_{maf}^{noflip} and P_{maf}^{flip} probabilities depend on the characteristic of function f. The adversary chooses the larger probability. It can be seen that

$$P_{maf}^{no-flip} + P_{maf}^{flip} = \frac{1}{2^k} + \frac{2^{k+1}-1}{2^{k+1}} = 1 + \frac{1}{2^{k+1}}$$

Hence, we get

$$P_{maf} = \max(P_{maf}^{no-flip}, P_{maf}^{flip}) \geq \frac{1}{2} + \frac{1}{2^{k+2}}$$

Therefore,

$$P_{dis} + P_{maf} \geq \frac{1}{2} + \frac{1}{2} + \frac{1}{2^{k+2}} = 1 + \frac{1}{2^{k+2}}.$$

We compare mafia fraud attacks $P_{maf}^{no-flip}, P_{maf}^{flip}$ with an approximation

(neglecting b since $P_b^{no-flip} \approx 1/2$ (See Remark 10)) :

$$P_{maf}^{flip} \geq P_{maf}^{no-flip} \iff c \cdot \frac{1}{2^{k+1}-1} \geq a$$

Therefore, it can be seen that, if $c \geq a \cdot (2^{k+1}-1)$ flipping the response is more preferable and no-flipping for the other cases. Note that, our approximation does not related with the theoretical results, it is just for simplifying the choice between P_{maf}^{noflip} and P_{maf}^{flip}. □

Remark 10. *Note that, the set $\mathcal{B}$ consists the session secrets that produce the responses, majority of them are equal, for any $c_{i-k}, c_{i-k+1}, \ldots, c_i$. Therefore, $P_b^{no-flip}$ can be expressed as*

$$P_b^{no-flip} = \frac{1}{b} \cdot \sum_{i=1}^{2^k-1} P_{b_i}^{no-flip} \cdot b_i \quad and \quad b = \sum_{i=1}^{2^k-1} b_i$$

where b_i is the cardinality of $\{y \in \mathbb{F}_2^m : |\alpha_y| = 2^k - i\}$
and $P_{b_i}^{no-flip} = \frac{\binom{2^k+i}{2}+\binom{2^k-i}{2}}{\binom{2^{k+1}}{2}}$.

It can be seen that

$$P_{b_i}^{no-flip} = 1/2 + \frac{i^2 - 2^{k-1}}{2^{2k+1} - 2^k} \qquad and$$
$$P_{b_i}^{no-flip} \geq 1/2 \iff i \geq 2^{\frac{k-1}{2}}$$

Therefore, $P_b^{no-flip} \approx 1/2$ is not an unrealistic assumption.

Corollary 2. *For a k-PCD protocol $\mathcal{P}$, if $a \cdot (2^{k+1}-1) \geq c$ is satisfied then, $P_{dis}(\mathcal{P}) \geq 1/2 + 1/2^{k+2}$ and $P_{maf}(P) + P_{dis}(P) \geq 1 + 1/2^{k+1}$.*

Proof. By the given condition,

$$\begin{aligned} & a \cdot (2^{k+1} - 1) \geq c \iff \\ & a \cdot (2^{k+1}) \geq c + a \iff \\ & \frac{a}{2^m} \geq \left(\frac{a}{2^m} \cdot \frac{1}{2^{k+1}} + \frac{c}{2^m} \cdot \frac{1}{2^{k+1}} \right) \end{aligned} \tag{6.3}$$

Using the inequalities in Theorem 20 and Equation 6.3;

$$\begin{aligned} P_{dis} &\geq \frac{a}{2^m} \cdot 1 + \frac{b}{2^m} \cdot \frac{2^k + 1}{2^{k+1}} + \frac{c}{2^m} \cdot \frac{1}{2} \\ &\geq \frac{1}{2} \cdot \left(\frac{a}{2^m} + \frac{b}{2^m} + \frac{c}{2^m} \right) + \frac{1}{2} \cdot \frac{a}{2^m} + \frac{1}{2^{k+1}} \cdot \frac{b}{2^m} \\ &\geq \frac{1}{2} + \frac{1}{2} \cdot \left(\frac{a}{2^m} \cdot \frac{1}{2^{k+1}} + \frac{b}{2^m} \cdot \frac{1}{2^{k+1}} + \frac{c}{2^m} \cdot \frac{1}{2^{k+1}} \right) \\ &\geq \frac{1}{2} + \frac{1}{2^{k+2}}. \end{aligned}$$

Therefore,

$$P_{dis} + P_{maf} \geq \frac{1}{2} + \frac{1}{2^{k+2}} + \frac{1}{2} + \frac{1}{2^{k+2}} \quad = 1 + \frac{1}{2^{k+1}}.$$

□

For different k values, the corresponding $P_{maf} + P_{dis}$ values are depicted in Figure 6.4. It s clearly seen that the summation goes to 1 when k values increase. For all $k \geq 8$, $P_{maf} + P_{dis} \equiv 1$.

6.4 The Construction of a k-PCD Protocol

In the previous section, we have already proved that k-PCD protocols can provide better security level than the CDD protocols. In this section, we introduce a method to improve the security of CCD protocols by adapting

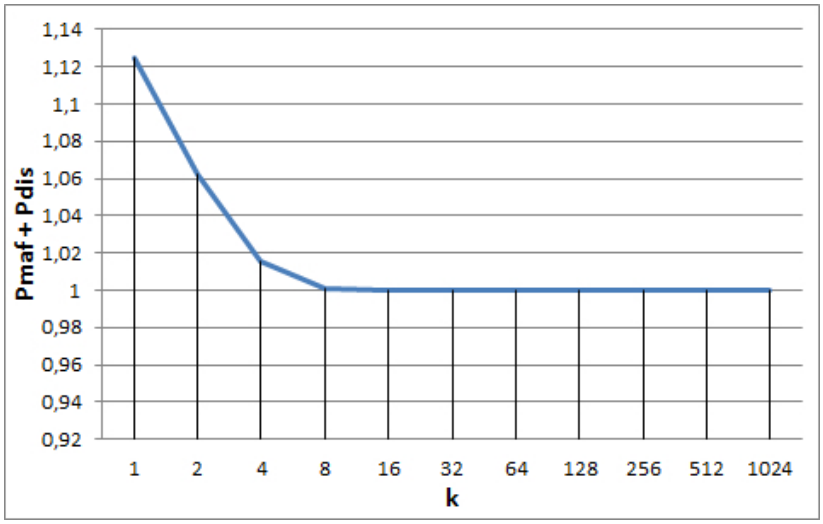

Figure 6.4: The $P_{maf} + P_{dis}$ for different k values

them to k-PCD protocols. In this context, we first give the notion of a natural extension. Then, we apply this extension on an existing protocol, HK protocol, to show the security enhancement.

Let $\mathcal{P}$ be a CCD protocol with the response function $f(c_i, y_0^i, \ldots, y_{m-1}^i) \rightarrow r_i$ and $\mathcal{P}'$ be a k-PCD protocol with the response function $g(c_{i-k}, \ldots, c_i, y_0, \ldots, y_{m-1}) \rightarrow r_i'$. We give the definition for a natural extension of a CCD protocol to provide a k-PCD protocol as follows.

Definition 18 (Natural Extension for CCD to k-PCD)**.** *$\mathcal{P}'$ is called a* natural extension *of $\mathcal{P}$ if $g(c_{i-k}, \ldots, c_i, y_0, \ldots, y_{m-1})$ is a Boolean function of the variables $f(Q(c_{i-k}, c_{i-k+1}, \ldots, c_i), y_0^i, \ldots, y_{m-1}^i)$ and $T(c_{i-k}, c_{i-k+1}, \ldots, c_i)$, where Q and T are Boolean functions of $k+1$ variables.*

We study HK protocol as an example of CCD protocols which has the security level as 3/4 against both mafia and distance frauds. We first apply the following natural extension for HK protocol to obtain an optimum security level for mafia fraud among k-PCD protocols.

$$
\begin{aligned}
g(c_{i-0},\ldots,c_{i-k},y_0,y_1) &= f(c_i,y_0^i,y_1^i) \\
&\oplus\ f(\bar{c}_{i-1},y_0^{i-1},y_1^{i-1}) \\
&\cdot \\
&\cdot \\
&\cdot \\
&\oplus\ f(\bar{c}_{i-k},y_0^{i-k},y_1^{i-k}) \\
&= y_{c_i}^i \oplus y_{\bar{c}_{i-1}}^{i-1} \ldots \oplus y_{\bar{c}_{i-k}}^{i-k},
\end{aligned}
\tag{6.4}
$$

where $\bar{c}_{i-1}$ is the complement of c_{i-1}. It is clearly seen that the response of each round depends on the current challenge and the complement of previous number of k challenges. In order to apply this extension into HK protocol, the verifier should send k random challenge in the initialization of the fast phase. The protocol steps are depicted in Figure 6.5.

In order to analyze this protocol, we look at how the response bits are distributed according to the challenge bits. Therefore, for each k values we generate two n-bit registers from a hash function (SHA-256). Then, we examine the distribution of the cardinality of the set $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{C}$ for each k values. Note that if $c/((2^{k+1}-1)a) \geq 1$, then the adversary can apply the attack described in Algorithm 6.4.

We did simulation on different k values and we observed that almost all of the ratios are greater than 1 and this means that we can apply the attack described in Algorithm 6.4. Hence, using this attack against this construction the success probability of mafia fraud would be $P_{maf} = 1/2 + 1/2^{k+2}$ (Corollary 2). To demonstrate the correctness of the corollary, we also simulate this attack for different k values and we see that there is no

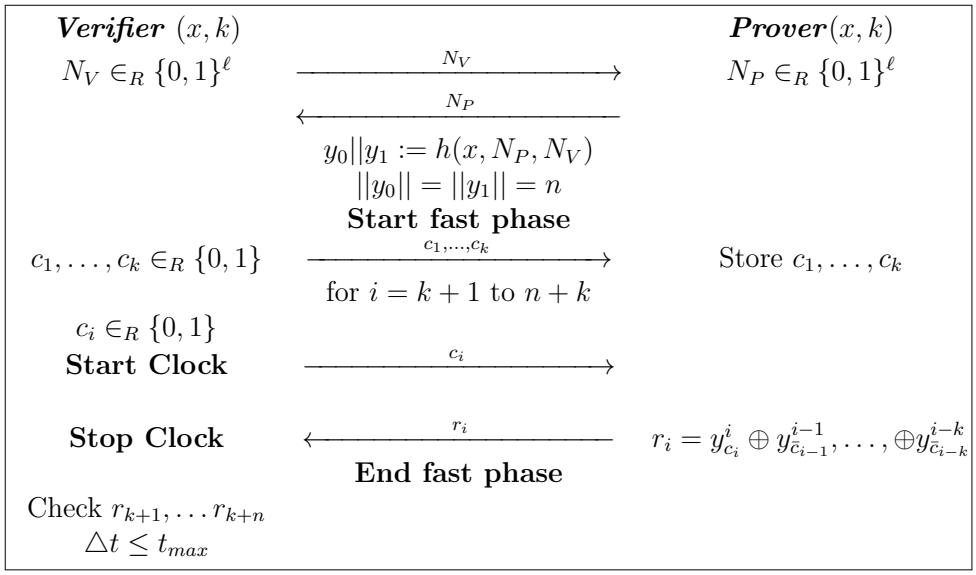

Figure 6.5: The proposed k-PCD Protocol

statistically significant difference between the observed success probability and the expected probability.

Now, let us analyze this protocol against distance fraud. In the security analysis, we consider the prover being in different regions. When the prover in region Z_1, the security level is same as the original HK protocol (i.e. 3/4) since the prover has access to all the previous challenges. When the prover is in region Z_2, the prover cannot access to both c_i and c_{i-1}, then the success probability would be $1/4 * 1 + 3/4 * 1/2 = 5/8$. Similarly, when the prover is in the region Z_{k+1}, the success probability would be $1/2^k + (1 - 1/2^k) * 1/2$. In order to enhance this protocol against distance fraud, we can extend this response function similar to [34]. The new response function would be as follows.

$$
\begin{aligned}
g(c_{i-k}, c_{i-k+1}, \ldots, c_i, y_0, y_1) &= f(c_i, y_0^i, y_1^i) \\
&\oplus\ c_{i-1} \\
&\oplus\ f(\bar{c}_{i-2}, y_0^{i-2}, y_1^{i-2}) \\
&\ \ . \\
&\ \ . \\
&\ \ . \\
&\oplus\ f(\bar{c}_{i-k}, y_0^{i-k}, y_1^{i-k}) \\
&= y_{c_i}^i \oplus c_{i-1} \oplus y_{\bar{c}_{i-2}}^{i-2} \ldots \oplus y_{\bar{c}_{i-k}}^{i-k},
\end{aligned}
\tag{6.5}
$$

In this response function, the challenge value c_{i-1} is not in the evaluation of the function f but it is used only for masking. Since the adversary cannot reach c_i and c_{i-1} in region Z_2, even though the adversary has unbounded computational resources, the success probability of computing the correct response cannot be more than 1/2. More generally, the prover in region Z_k where $k \geq 2$, she has no access to specifically c_{i-1} the success probability would be at most 1/2 for single round.

6.5 The Summary of the Chapter

In this chapter, we have explained RFID distance bounding protocols and briefly reviewed current challenge dependent protocols. We also introduced the notion of k-PCD protocols. Thus, we have shown that when we increase the dependency parameter k, security level against mafia fraud attack and distance fraud attack increase as they are expected. We have supported these expectations by calculating success probabilities of distance fraud and mafia

fraud attacks for k-PCD protocols. On the other hand, trade-of curve of k-PCD protocol is plotted and compared other versions. We also prove the conjecture that the best trade-off curve for k_1-PCD protocols lies above the best trade-off curve for k_2-PCD protocols where $k_1 < k_2$. Finally, we provide a way of constructing k-PCD protocols with only two registers and prove that this construction achieves the computed security.

Chapter 7

ACHIEVING OPTIMUM SECURITY: AN RFID DISTANCE BOUNDING PROTOCOL BASED ON PUFS

In this chapter, we first analyze the security of Sadeghi et al.'s PUF based RFID authentication protocol [3] by our stronger adversarial model in which an adversary has access to the volatile memory of the tag. We show that their protocol is not secure in this model and we propose a new technique to avoid this attack even if the adversary has the ability to access volatile memory.

Next, we apply this technique to propose a new PUF based RFID distance bounding protocol. To the best of our knowledge, this is the first proposal that introduces a PUF based RFID distance bounding protocol. It

is well-known that obtaining the long-term key of a tag is crucial in order to successfully perform the terrorist and the distance frauds. One of the main problems of existing distance bounding protocols is storing the long-term key into its memory which can be obtained by a fraudulent prover. Our protocol has the advantage that the long-term key will not be stored in the memory of the tag but will be reconstructed by using a PUF circuit.

Our first PUF based distance bounding protocol is based on the well-known Hancke and Kuhn's scheme [78]. Although their original protocol is known to be simple and efficient, the adversary's probability of success is high (namely $(3/4)^n$ for both the distance and the mafia frauds, and 1 for the terrorist fraud). By the use of PUF, the adversarial capabilities of the terrorist fraud is reduced to that of the mafia fraud. In this way, we improve the security of Hancke-Kuhn's protocol against the terrorist fraud from 1 to $(3/4)^n$ under an assumption that the victim tag is required to be alive after compromising.

Moreover, we propose our second distance bounding protocol which is an extension of the first one involving a hash-based final signature. To the best of our knowledge, this is the first protocol that achieves the ideal security levels $(1/2)^n$ against all frauds without any assumption.

The organization of the chapter is as follows. In Section 7.1, we illustrate the notion of PUF functions and its characteristics. Section 7.2 describes the adversary capabilities for both PUFs and distance bounding protocols. In Section 7.3, we propose our first distance bounding protocol and analyze its security. In Section 7.4, we present our second protocol and analyze its security. Section 7.5 concludes the chapter.

The results presented in Chapter 7 was published in [1].

7.1 Physically Unclonable Functions (PUFs)

In this chapter, we will focus on an ideal PUF P such that $P : \{0,1\}^{\ell} \rightarrow \{0,1\}^{m}$ where the challenge c_i is mapped to the response r_i. P is said to be an *ideal PUF* if the following properties are satisfied.

1. If $c_i = c_j$, then we have $r_i = r_j$ for a PUF on a particular device. Presenting the same challenge to the PUF on a different device will produce a different response.

2. The mapping between c_i and r_i is unpredictable and random. For instance, if r_i and r_j differ in only a single bit, knowledge of c_i does not reveal usable information to predict c_j.

3. Any attempt to physically tamper with the device implementing P causes to change its physical characteristics. Namely, P is then destructed and can no longer be evaluated correctly.

We note that the third property of the idealized PUF can be achieved by integrating PUF circuit with the chip on the tag. To do so, Tuyls *et al.* in [71] propose *Integrated PUFs* (I-PUFs). For further information we recommend reading [3, 71]. In this work, we use the ideal PUF for distance bounding protocols and show how the security is enhanced to ideal levels.

7.2 Adversary Capabilities

In this section, we first present a stronger adversarial model for analysis of PUF based RFID authentication protocols which considers the accessibility to the internal state of tags. We next discuss the notion of white and black box models for distance bounding protocols. We aim to unify and express the adversarial capabilities of PUFs and distance bounding protocols.

7.2.1 Adversary Capabilities on PUFs

In a PUF based authentication protocol, the shared secrets are stored in its physical characteristics instead of storing them in a non-volatile memory. These keys are reconstructed whenever needed during the execution of the protocol. As soon as the keys are reconstructed, they are stored in a volatile memory of the RFID chip. In some previous articles (e.g., [3, 71]), it is assumed that the communication between a PUF circuit and a chip is not tractable by any side-channel attack.

Unlike the previous works, in this chapter, we propose a more stronger adversary model where an attacker has the ability to compromise the tag and reaches the state in the volatile memory. Since the structure of the PUF circuit has been destroyed, the attacker is no longer able to re-evaluate the PUF again. Thus, a malicious tag owner can perform only one side-channel attack on the tag and access the volatile memory only once. For instance, Halderman *et al.* recently demonstrated a side-channel attack for DRAM, called *cold boot attack* [115]. In this attack, they first powered off the system and later showed how to extend the main memory persistence by 'freezing' the DRAM chips in order to maintain the memory cell state. In this way, an adversary will be able to retrieve any password or cryptographic key that was not disappeared before the system is switched off.

The protocol of Sadeghi et al. [3] is facing a similar attack described above. Their protocol is briefly described as follows (Figure 7.1). Let $l \in \mathbb{N}$ be a security parameter, and F:$\{0,1\}^k \times \{0,1\}^{2\alpha} \rightarrow \{0,1\}^\beta$ be a public pseudorandom (PRF) function. Each tag $\mathcal{T}$ is equipped with a PUF function P:$\{0,1\}^\gamma \rightarrow \{0,1\}^k$ and is initialized with a random state $S_1 \in_R \{0,1\}^\gamma$. The credential of each tag (ID, K), where $K \leftarrow P(S)$ and is stored in the database DB of the reader. The reader $\mathcal{R}$ first picks a random nonce a to the

tag $\mathcal{T}_{ID}$. Then, $\mathcal{T}_{ID}$ picks a random nonce b and evaluates the PUF function $K = P(S)$. $\mathcal{T}_{ID}$ computes $c = F_K(a, b)$ and sends the message c along with the random nonce b and immediately erases K, a, b and c from its volatile memory. Upon receiving of b and c, $\mathcal{R}$ evaluates $c' = F_K(a, b)$ for each tuple (ID,K) in DB until there is a match. If a matching (ID,K) is found, then it accepts $\mathcal{T}_{ID}$ and returns ID; otherwise, it rejects by sending $\perp$ back.

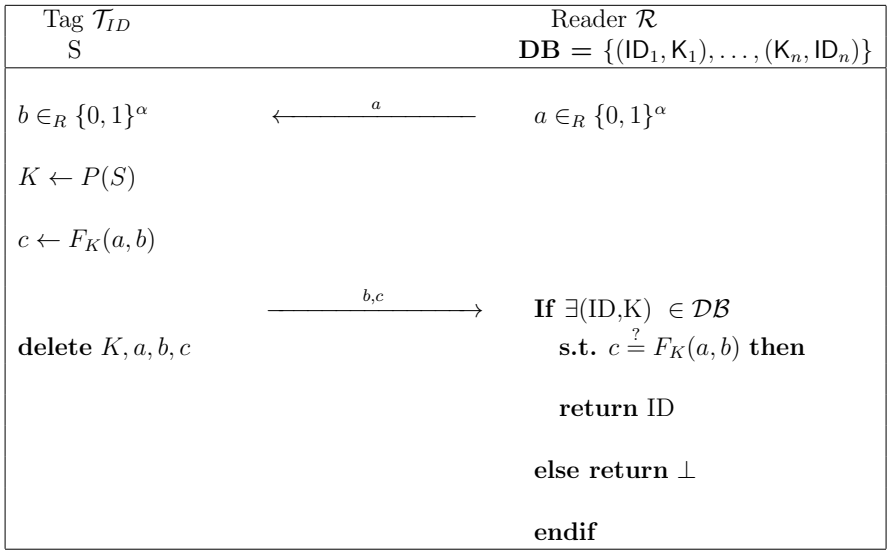

Figure 7.1: Sadeghi et al.'s authentication protocol

The authors claim that their protocol achieves destructive-privacy under the assumption that K is inaccessible. However, we show that their protocol suffers from the same above-mentioned cold-boot attack. Assume that an adversary sends a random nonce a to the tag $\mathcal{T}_{ID}$. $\mathcal{T}_{ID}$ then generates another random nonce b and reconstructs a secret K by evaluating the PUF with input S. The secret K is stored in the volatile memory during the computation of $c = F_K(a, b)$. The adversary compromises $\mathcal{T}_{ID}$ while $c = F_K(a, b)$ is computed and can capture the secret K. Hence, the tag can be successfully cloned although the structure of the PUF circuit has been destroyed.

In order to thwart this attack, instead of using only one key we propose to

use two different keys K, L which are consecutively generated as outputs of the PUF function. Note that K and L never appear in the volatile memory at the same time. First, K is used as an input of one-way PRF function, and then completely deleted from the memory. Next, in a similar way, L is generated and used in the PRF function. Hence, whenever an adversary applies the above-mentioned attack he will be able to obtain only one of the keys, and hence will not have sufficient information to defeat the privacy. Also, since the PUF circuit has been destroyed he will not be able to perform the same attack again. Thus, applying our technique avoids the tag cloning.

7.2.2 Adversary Capabilities on Distance Bounding Protocols

In the analysis of our protocols, Dolev-Yao adversary model are considered [116]. In this model, the adversary can perform polynomial number of computations and cannot obtain the secret keys from the honest parties. This assumption is then relaxed with the terrorist and distance frauds, where the prover has access to the keys [52]. However, he disagrees to share these keys with any third party. The adversary may use one of the three strategies to query a prover such as pre-ask strategy, post-ask strategy and early-reply strategy. The detailed explanations of these strategies are addressed in [52].

As in the conventional distance bounding protocols, we also assume that the verifier is an honest party where it faithfully follows the protocol specifications without cheating. Mafia fraud is a kind of man-in-the-middle attack where an adversary defeats both honest parties i.e., verifier and prover. Unlike mafia fraud, in distance and terrorist frauds, the prover himself is dishonest. The previous distance bounding protocols consider that the prover has a full control on the execution of the algorithm in the device. As it is discussed

in Section 7.1, PUFs can be used to provide resistance against side-channel attacks. Therefore, an adversary can be limited to the execution of the algorithm inside the device. In order to analyze distance bounding protocols, the generic capabilities of the adversary are addressed in [52]. The capabilities are categorized in two models, white-box model and black-box model. The following definitions of these two models are excerpted from [52].

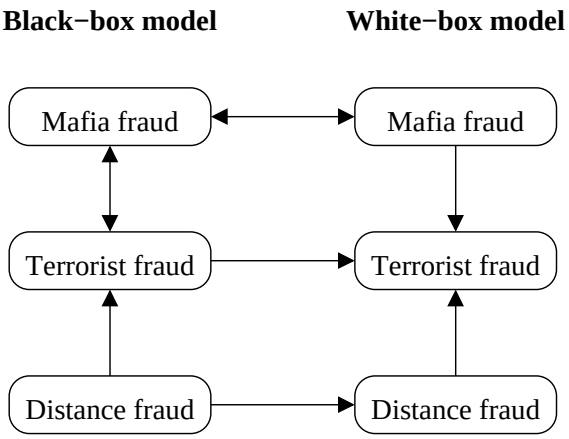

Figure 7.2: Relations between the frauds in the white-box and the black-box models.

Definition 19. ***(Black-box model)*** *In this model, the prover cannot observe or tamper with the execution of the algorithm.*

Definition 20. ***(White-box model)*** *In this model, the prover has full access to the implementation of the algorithm and has a complete control over the execution environment.*

Regarding to the white-box and the black-box models Figure 7.2 presents the relation between the distance, mafia and terrorist frauds. An arrow from X to Y means that, for any fraud in X that succeeds with probability p_X, then there exists an attack in Y that succeeds with probability p_Y such

that $p_Y \geq p_X$. Two side arrow means that the success probabilities of two corresponding frauds are equal [52].

It is interesting to note that in the black-box model, the success probabilities of the mafia and the terrorist frauds are equal (Figure 7.2).

7.3 Our First Distance Bounding Protocol

We now propose the first PUF based distance bounding protocol which is efficient for implementation in low cost devices. In the next section, we extend this protocol by adding a final signature to enhance the security against both mafia fraud, terrorist fraud and distance fraud.

The former achieves the security level of $(3/4)^n$ against mafia and distance frauds and $(3/4)^n$ against the terrorist fraud under an assumption that the tag is wanted to be still functional, where n is the number challenge/response bits during the fast phase. We show in the next section that the latter achieves the ideal security level against all the frauds (i.e., $(1/2)^n$).

7.3.1 Protocol Descriptions

Our first distance bounding protocol is based on Hancke and Kuhn's scheme [78], which is the starting point of this work. Although their protocol is simple and efficient the adversary's probability of success is high. The steps of our protocol are summarized below and depicted in Figure 7.3.

7.3.1.1 Initialization

Let $P_i : \{0,1\}^k \rightarrow \{0,1\}^\ell$ be a (unique) ideal PUF of the i-th legitimate prover $\mathcal{P}_i$. The credentials database DB of the verifier $\mathcal{V}$ stores a tuple (K_i, L_i) where $K_i = P_i(G_i^1)$ and $L_i = P_i(G_i^2)$ for random states G_i^1, $G_i^2 \in_R$

$\{0,1\}^k$. Let also $F : \{0,1\}^\ell \times \{0,1\}^{2\ell} \to \{0,1\}^{2\ell}$ be a one-way pseudo-random function. We denote n as the main security parameter of the fast phase where $3n = 2\ell$. $|S|$ denotes the bit-length of a bit-string S.

Our protocol consists of two phases: a slow phase and a fast phase.

Slow phase:

- First of all, $\mathcal{V}$ generates a random nonce r_V and sends it to $\mathcal{P}_i$.
- Upon receiving r_V, $\mathcal{P}_i$ generates a random nonce r_P and reconstructs $K_i = P_i(G_i^1)$. $\mathcal{P}_i$ computes $T = F_{K_i}(r_P, r_V)$, then immediately deletes K_i from the memory. After that, $\mathcal{P}_i$ reconstructs the secret key $L_i = P_i(G_i^2)$ and computes the message $F_{L_i}(T)$. Similarly, $\mathcal{P}_i$ immediately deletes L_i from the memory. The value $F_{L_i}(T)$ is divided into three registers v_1, v_2 and v_3 where $|v_1| = |v_2| = |v_3| = n$. Finally, $\mathcal{P}_i$ sends r_T and v_1 to $\mathcal{V}$.
- Upon receiving r_T and v_1, for each tuple (K_i, L_i) in DB $\mathcal{V}$ searches $v_1', v_2', v_3' = F_L(F_K(r_P, r_V))$ such that $v_1' = v_1$. If not found, $\mathcal{V}$ aborts the protocol.

Fast phase:

- The fast phase consists of n bit-wise challenge-response exchange. For each round $j \in \{1, \ldots, n\}$, $\mathcal{V}$ picks a random challenge bit c_j and sends it to $\mathcal{P}_i$.
- $\mathcal{P}_i$ immediately responds $r_j = v_2^j$ if $c_j = 0$, otherwise $r_j = v_3^j$.

7.3.1.2 Verification

Whenever the fast phase is finished $\mathcal{V}$ verifies that the responses from $\mathcal{P}_i$ are correct and checks whether $\triangle t_j \leq \triangle t_{max} \ \forall \ j = 1, \ldots, n$ where $\triangle t_{max}$ is a

timing bound.

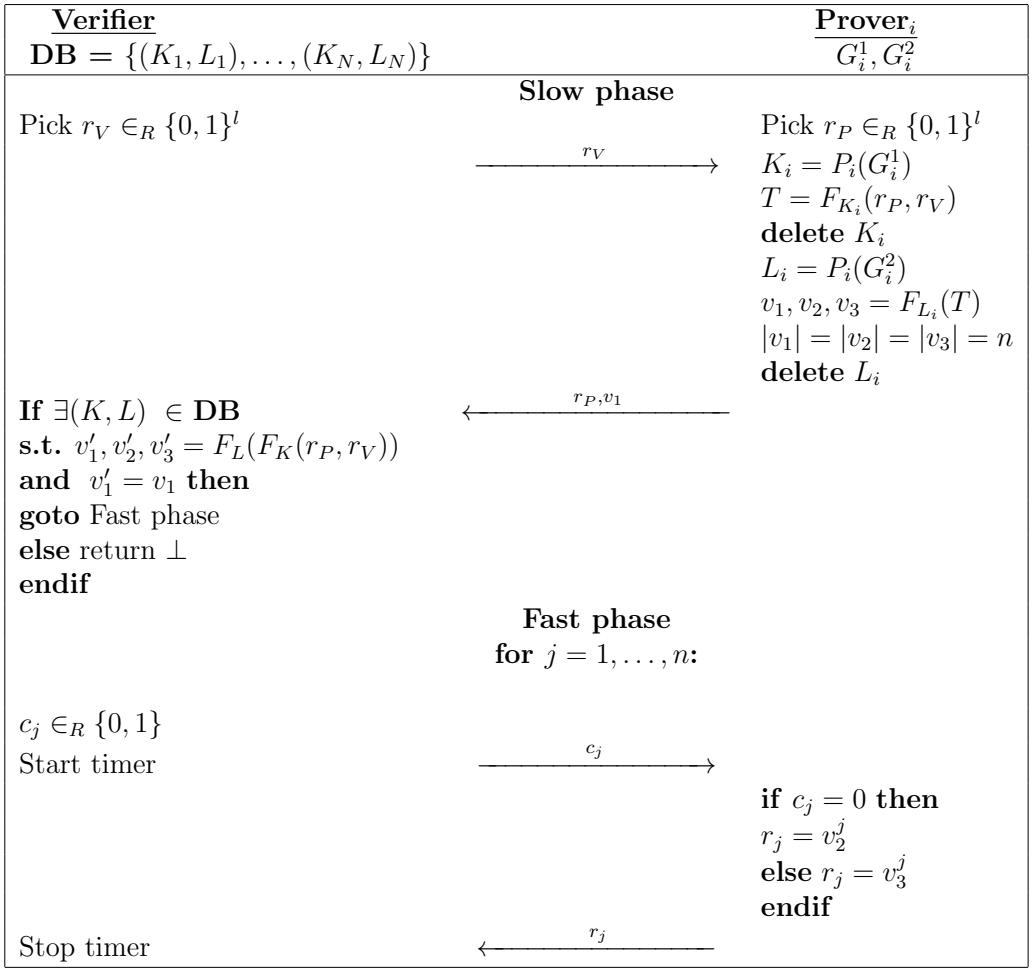

Figure 7.3: Our first PUF based distance bounding protocol without a final signature

7.3.2 Security Analysis of The First Protocol

Mafia, terrorist, and distance frauds are the three main security concerns when considering distance bounding protocols.

The following Theorem 21 indicates that no adversary (e.g., a malicious tag owner) can access to both secrets K_i and L_i. Thus, the use of PUF in

the protocol makes the RFID tags as tamper proof against any malicious adversary.

Theorem 21. *Let K_i, L_i be secrets of a tag $\mathcal{T}_i$ for some i in the above-mentioned protocol (see Figure 3). Assume that there is an adversary $\mathcal{A}$ with a full side-channel capability on the tag $\mathcal{T}_i$. If P_i is an ideal PUF, then $\mathcal{A}$ can only access either the secret K_i or the secret L_i, but not both in the same tag $\mathcal{T}_i$.*

Proof. (sketch) The pre-keys G_i^1 and G_i^2 are used as input for P_i function to reconstruct the real keys K_i and L_i. The real keys only appear during the execution of the protocol. Note that K_i and L_i never appear in the memory of $\mathcal{T}_i$ at the same time because K_i is first used as an input of a one-way PRF function, and then completely deleted from the memory. Next, in a similar way, L_i is generated and used in the PRF function. Whenever $\mathcal{A}$ applies a side channel attack to $\mathcal{T}_i$, the physical characteristics of the PUF P_i will be broken and will no longer be evaluated correctly. If $\mathcal{A}$ applies side-channel attack to extract K_i then the structure of P_i will be destroyed and L_i cannot be generated. Similarly, if $\mathcal{A}$ applies side-channel attack to extract L_i she cannot obtain K_i since it is already deleted. Therefore, $\mathcal{A}$ can access either K_i or L_i but not both. Hence, $\mathcal{A}$ will not be able to get the complete key of $\mathcal{T}_i$. □

Theorem 21 indicates that a malicious prover cannot obtain the secret keys, and thus cannot evaluate the registers v_1, v_2, v_3. In the black-box model, note that it is already proven that the capability of terrorist fraud is equivalent to the mafia fraud [52] (see also Figure 7.3). Hence, for the black-box model, we combine the security analysis of both mafia and terrorist frauds.

Note that a malicious prover can access to the registers v_1, v_2, v_3 by ap-

plying side-channel attack only once. Furthermore, she can complete only the current session successfully because of the destruction of PUF. However, since the registers v_1, v_2, v_3 are randomized this does not give any future advantage to the adversary.

For a distance bounding protocol, an adversary is able to use three different strategies to conduct her attack such that early-reply, pre-ask, and post-ask [52]. We denote by $\mathcal{A}$ a malicious adversary. Let also denote by MF, TF and DF the mafia fraud, the terrorist fraud and the distance fraud, respectively. Let F be a fraud and S be the strategy used by the adversary $\mathcal{A}$. Let $Pr_{F|S}$ be the success probability in the black-box model of the fraud F ($MF/DF/TF$) using the strategy S (*early/pre/post*). Note that the strategies can also be combined and this is denoted by an &. Next, we describe the success probability of each fraud as follows.

7.3.2.1 Mafia and Terrorist Fraud Analysis in Black-box Model

The adversary uses pre-ask or post-ask strategies in order to achieve mafia or terrorist fraud.

In **pre-ask strategy strategy** [52], $\mathcal{A}$ first relays the slow phase between $\mathcal{V}$ and $\mathcal{P}$. Then $\mathcal{A}$ executes the fast phase with $\mathcal{P}$. $\mathcal{A}$ sends predicted challenges c'_j to $\mathcal{P}$ and get the responses r'_j corresponding to her challenges. With this a strategy, $\mathcal{A}$ obtains only one of the register. Afterward, $\mathcal{A}$ executes the fast phase with $\mathcal{V}$ and receives the challenges c_js. There are two equal likely cases, (i) if $c_j = c'_j$ $\mathcal{A}$ sends the correct response with probability of 1; otherwise, (ii) $\mathcal{A}$ guess the response with probability of 1/2. Hence, the success probability of mafia fraud and terrorist fraud for n-round fast phase

is computed as follows.

$$Pr_{MF|pre} = Pr_{TF|pre} = \left(\frac{1}{2} \cdot 1 + \frac{1}{2} \cdot \frac{1}{2}\right)^n = \left(\frac{3}{4}\right)^n.$$

In **post-ask strategy** [52], $\mathcal{A}$ first relays the slow phase, then executes the fast phase with $\mathcal{V}$. The probability of sending a correct response for a challenge is 1/2. Then, $\mathcal{A}$ queries $\mathcal{P}$ with the correct challenges received during the fast phase to check whether she is succeed on cheating. The success probability of mafia fraud for this strategy is:

$$Pr_{MF|post} = Pr_{TF|post} = \left(\frac{1}{2}\right)^n.$$

To maximize the success probability the attacker chooses the best strategy. Hence, the success probability of both mafia and terrorist frauds are $(3/4)^n$.

7.3.2.2 Terrorist fraud analysis in the white-box model

In **pre-ask strategy strategy**, terrorist fraud performs the attack as follows.

- The terrorist fraud first gets the random nonce from the verifier and relays them to the prover.
- The prover executes the protocol and when v_1, v_2 and v_3 are computed, the prover compromise the tag and reaches the internal state and views r_P, v_1, v_2, v_3, L_i.
- The prover sends this four (r_P, v_1, v_2, v_3) to the terrorist.

- Now, the terrorist fraud can impersonate the prover with the success probability of 1.

However, this tampering causes the destruction of the prover's chip. This attack may not be accomplished by the prover because of the destruction. In this case, in the white-box analysis of the terrorist fraud, tampering the tag would be infeasible. The following remark considers this case.

Remark 11. *If the tag is required to be functional after an attack, the corruption of the tag would not be allowed. Hence, the success probability of terrorist fraud with pre-ask strategy will be the same as the success probability with pre-ask strategy in the black-box model,* $(3/4)^n$.

In **post-ask strategy strategy**, terrorist fraud performs the attack as follows.

- $\mathcal{A}$ first relays the slow phase, then executes the fast phase with $\mathcal{V}$.
- The probability of sending a correct response for a challenge is $1/2$.
- Then, $\mathcal{A}$ queries $\mathcal{P}$ with the correct challenges received during the fast phase to check whether she succeeds on cheating.
- The success probability of mafia fraud for this strategy is:

$$Pr_{TF|post} = \left(\frac{1}{2}\right)^n.$$

To maximize the success probability, the attacker chooses the best strategy. Hence, the success probability of terrorist fraud is $(3/4)^n$ when the target tag is not allowed to be destroyed. Otherwise, it is 1.

7.3.2.3 Distance Fraud Analysis in Black-box Model

In distance fraud, the tag owner herself is fraudulent who tries to cheat on her proximity from $\mathcal{V}$. It is important to highlight that unlike the existing protocols, the tag owner cannot control the internal executions of the tag in our protocol. The fraudulent prover can query its tag to get the responses. In distance fraud, since the prover is outside of the legal authentication region she should send the responses earlier in order to pass the proximity check (i.e., round trip time measurement). This is called *early-reply strategy* [52]. To ease our analysis, we denote the fraudulent tag owner by $\mathcal{A}$, and the tag by $\mathcal{T}$.

In **pre-ask combined with early-reply strategy strategy**, $\mathcal{A}$ first relays the slow phase between $\mathcal{V}$ and $\mathcal{T}$, then executes the fast phase with $\mathcal{T}$. $\mathcal{A}$ can only obtain n-bit responses corresponding to her predicted challenges. Since $\mathcal{A}$ is not inside the neighborhood of $\mathcal{V}$, she sends her responses in advance. Two cases occurs for each round of the fast phase. (i) $\mathcal{A}$ predicts $\mathcal{V}$'s challenge correctly, then she sends a correct corresponding response in advance. (ii) $\mathcal{A}$ cannot predict $\mathcal{V}$'s challenge correctly, but she can send a correct answer with probability of 1/2. Thus, the distance fraud success probability is:

$$Pr_{DF|pre\&early} = \left(\frac{1}{2} \cdot 1 + \frac{1}{2} \cdot \frac{1}{2}\right)^n = \left(\frac{3}{4}\right)^n.$$

Now, let us look at the success probability of the adversary with **post-ask combined with early-reply strategy**. Similar to the mafia fraud analysis, it is clear that using the post-ask strategy is equivalent to randomly guessing the responses,

$$Pr_{DF|post\&early} = \left(\frac{1}{2}\right)^n.$$

The distance fraud attacker chooses the strategy with the maximum success probability. Consequently, the success probability of distance fraud is $(3/4)^n$.

7.3.2.4 Distance Fraud Analysis in White-box Model

In white-box model, distance fraud, $\mathcal{A}$, has chance of compromising the tag and can execute the protocol algorithm with polynomial number of times. However, $\mathcal{A}$ cannot access to both secrets in the chip by Theorem 21, so $\mathcal{A}$ cannot perform such attack.

The distance fraud can perform pre-ask combined with early-reply strategy strategy. $\mathcal{A}$ first relays the slow phase between $\mathcal{V}$ and $\mathcal{T}$. Then, $\mathcal{A}$ can access to both registers only once and can use these registers to realize her attack. In her strategy, two cases occurs for each round of the fast phase. (i) $\mathcal{A}$ predicts $\mathcal{V}$'s challenge correctly when two register bits are equal, then she sends a correct corresponding response in advance. (ii) $\mathcal{A}$ cannot not predict $\mathcal{V}$'s challenge correctly because the register bits are different, but she can send a correct answer with probability of 1/2. Thus, the distance fraud success probability is:

$$Pr_{DF|pre\&early} = \left(\frac{1}{2} \cdot 1 + \frac{1}{2} \cdot \frac{1}{2}\right)^n = \left(\frac{3}{4}\right)^n.$$

7.4 Our Enhanced Distance Bounding Protocol

We are now ready to propose our extended protocol which is resistant to all the frauds.

7.4.1 Protocol Descriptions

In what follows, we present our second protocol which is an extension of the first one by adding a final signature. This protocol consists of three phases. The first two phases are exactly the same with the previous protocol.

In the third phase, the prover computes the following final signature as follows. It first evaluates the PUF with G_i^1 to reconstruct K_i and computes $T_{temp} = h(c_1, \ldots, c_n, T, K_i)$ where h denotes a collusion resistant and one-way hash function. Then, it erases K_i from memory and reconstructs $L_i = P_i(G_i^2)$ and computes $f_{sign} = h(T_{temp}, L_i)$ and deletes L_i. The prover sends f_{sign} to the verifier, then the verifier checks the correctness of this message.

7.4.2 Security Analysis of Extended Protocol

In the extended protocol, the challenges received by the tag are digested in f_{sign}. Therefore, in order to pass the authentication, the adversary must send a valid final signature to the verifier.

7.4.3 Security analysis in Black-Box Model

Considering the black-box model, there are two strategies for both mafia and terrorist frauds:

(i) In the pre-ask strategy, the adversary first executes the fast phase with the prover by sending $c'_1, \ldots, c'_n$ challenges, then prover replies with the corresponding responses $r'_1, \ldots, r'_n$. In the final phase, the adversary gets $f'_{sign} = h(c'_1, \ldots, c'_n, T, K_i)$. The final signature is valid if and only if all the challenges $c_1, \ldots, c_n$ sent by the verifier are equal to the ones predicted by the adversary. Thus, it is clear that the probability of $f_{sign} = f'_{sign}$ is equal to $(1/2)^n$.

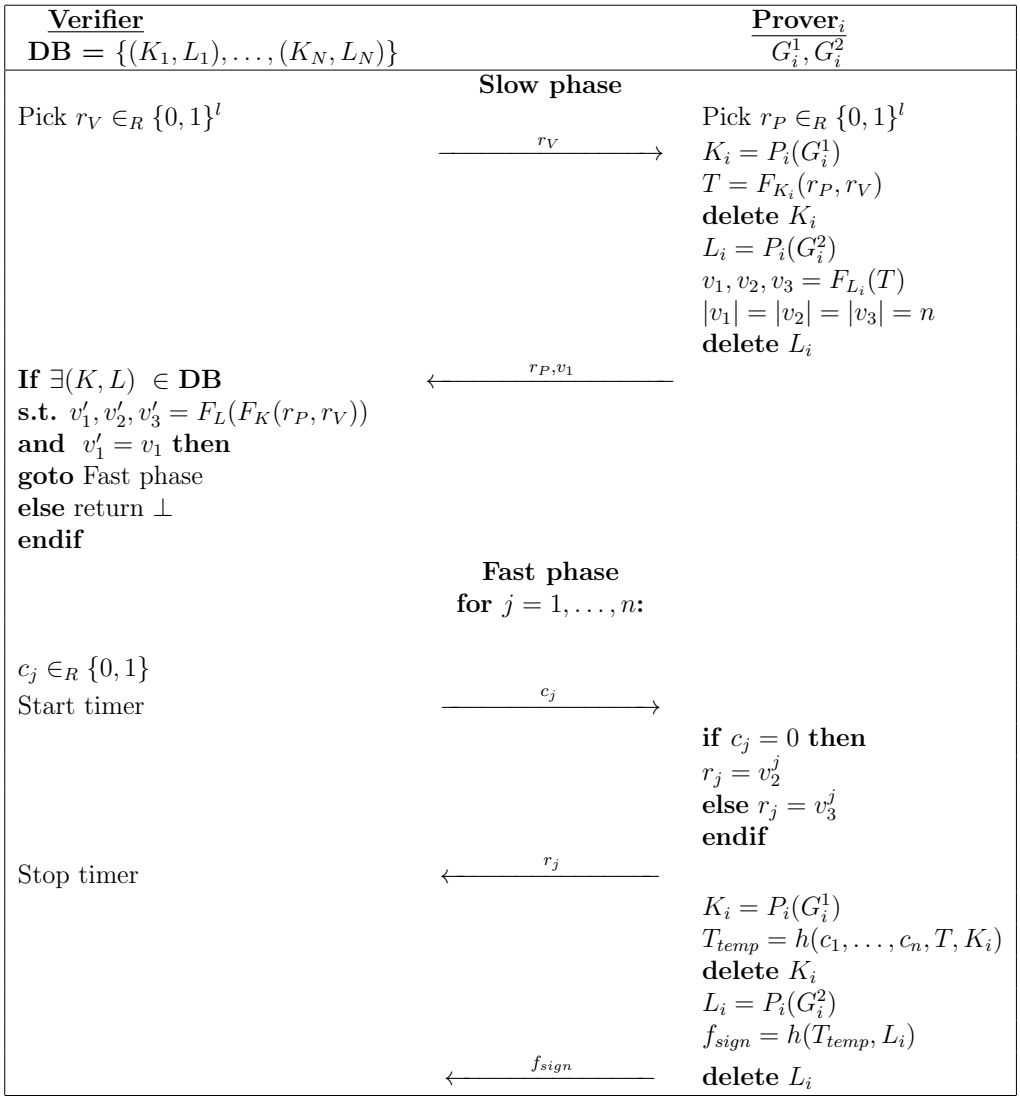

Figure 7.4: Our enhanced PUF based distance bounding protocol with a final signature

(ii) In the post-ask strategy, the adversary first plays with the verifier and guesses all the responses during the fast phase. If she passes the fast phase then it is easy to get the valid final signature from the prover by forwarding the challenges of the verifier. However, the probability of guessing all the correct responses during the fast phase is equal to $(1/2)^n$. Thus,

$$Pr_{MF} = Pr_{TF} = \left(\frac{1}{2}\right)^n.$$

Similarly, the security of the extended protocol for distance fraud is also bounded by $(1/2)^n$ because in order to receive a valid final signature from the tag the fraudulent prover should have queried the tag with all correct challenges in advance. Hence, the use of final signature enhances the security level of our extended protocol against the distance fraud to the ideal level $(1/2)^n$.

7.4.4 Security Analysis in White-Box Model

Note that, in the white box model, terrorist fraud collaborates with the prover. By the definition of PUFs, the prover has only one opportunity for compromising the tag because of the destruction of PUF.

Considering the white-box model, there are also two strategies for both distance and terrorist frauds such as pre-ask strategy and post-ask strategy. In both strategies, the fraudulent prover needs the computation of the final signature correctly. However, Theorem 21 indicates that a malicious prover cannot obtain both the secret K_i and L_i by corruption. Since one of the keys will be still unknown by the adversary, the final signature cannot be computed in advance by the adversary. The success probability of the adversary is bounded by either the probability of predicting of a valid final signature or

The Protocol	Black-Box Model			White-Box Model		
	Mafia	Terrorist	Distance	Mafia	Terrorist	Distance
First	$(3/4)^n$	$(3/4)^n$	$(3/4)^n$	$(3/4)^n$	$(3/4)^n$ / 1*	$(3/4)^n$
Extended	$(1/2)^n$	$(1/2)^n$	$(1/2)^n$	$(1/2)^n$	$(1/2)^n$	$(1/2)^n$
∗If the tag is required to be functional after the corruption, the success probability of terrorist fraud is $(3/4)^n$, otherwise 1.						

Table 7.1: The security analysis of our distance bounding protocols

predicting the challenge bits in advance. Therefore, we can concluded that the success probability of both terrorist fraud and distance fraud is at most $(1/2)^n$.

7.5 The Summary of the Chapter

Relay attacks are indeed practical threats for RFID systems since using only cryptographic primitives it is not easy to thwart mafia, distance and terrorist frauds. Distance bounding protocols are used to mitigate these threats. However, the existing distance bounding protocols cannot achieve ideal security level against all frauds.

In this chapter, we present the first PUF based distance bounding authentication protocol. Note that the protocols based on PUFs are known to be powerful since attacks can be easily prevented and the use of expensive cryptographic primitives can be minimized. In our protocol, we use the idea of key storage mechanism based on PUFs for public-key cryptography presented by Tuyls and Batina [71] (which is also later used for symmetric key storage by Sadeghi et al. [3]). We modified their protocol in such a way that all the keys are not constructed at the same time. This enables us to achieve a stronger assumption and there is no way to extract the whole secret key from the tag. We show that our first protocol achieves the security level of $(3/4)^n$ against mafia and distance frauds and $(3/4)^n$ against terrorist fraud

under the assumption of the tag is still functional. We also extend our protocol by adding a final signature to enhance the security levels. Namely, we achieve the security level $(1/2)^n$ against for all mafia, terrorist and distance frauds. To the best our knowledge, this is the first chapter that achieves the ideal security level $(1/2)^n$ against all frauds.

An interesting further question is whether it is possible to find an efficient protocol without a final signature having the ideal security level against all frauds.

Chapter 8

ARCS: ANONYMOUS RFID AUTHENTICATION BASED ON CLOUD SERVICES

Every potential application of RFID systems may require a different approach. As an illustration, manufacturers of consumer goods require a full range of compliance-tagging and verification solutions. When working to meet RFID compliance mandates, today's one foremost exigency is the need to implement a scalable solution that not only satisfies but also allows for future growth. Traditional RFID inventory management solutions are expensive for large amount of items, in the sense that they require self-server maintenance and significant IT intervention.

Moreover in some applications, multiple read points may be required to track the products throughout the workplace. In conventional systems, multiple number of databases can be established which cause several operational problems such that synchronization of the databases, expensive system, difficult and separate management. To realize the benefits of RFID, retailers will

need to upgrade their IT infrastructure in a number of areas, and their interfaces with other business will have to be closer. The verification of tagged items by RFID systems provides full traceability from sender (e.g. manufacturer) to receiver by maintaining a single database placed in a server. This provides assurance that a product has been shipped and delivered. This is where cloud computing may come in to provide flexibility to access to the database and authenticate the tagged items/individuals. A cloud system can be simply considered as a server farm that has great computational and storage capacity. In fact, this can greatly reduce the start-up costs as well as the drain that can be put on the IT staff for the RFID system maintenance. Thanks to cloud computing, retailers will not need to upgrade their IT infrastructure.

The real value and return on investment of RFID technology come from how the information derived from RFID tags and systems is applied to enterprise applications that control core business processes (inventory management, supply chain management, warehouse tracking, and location control applications). An RFID system using cloud service as a back-end database and computational capacity is strongly relevant when there is multiple facility providers (such as library, sport center, museum etc.) which are connected to an executive enterprise. In addition, centralizing the above RFID applications and integrating them with an executive systems will require a new level of systems integration capabilities. Figure 8.1 depicts an illustration of such a scenario, where each facility provider is connected to an executive enterprise through a cloud service. Using a unified cloud database empowers a single authentication system to more effectively manage pricing, events, reduces inventory losses, expands service offerings, and provides entire RFID infrastructures using a single system. The cloud paradigm provides the ability to

offer a single card to each user to get service from multiple applications.

Besides the usability and availability of cloud computing, the main question is to understand and manage the public concern such as the confidentiality and privacy issues. Therefore some skeptic questions may arise. Can we provide the confidentiality and privacy of the user's data in the public cloud domain? Can we maintain an authentication mechanism by using a remote cloud service like in our private database?

In RFID literature some protocols require exhaustive search on private identity [6,117] or asymmetric calculation [118–124] in order to have a strong authentication mechanism. For large systems, these strong private protocols may result in the need of heavy and expensive servers that have fast computational capacity or large storage.

Motivated from the innovations offered by cloud computing, the primary focus of this chapter is to propose a security and privacy model for the existing RFID systems melded with the cloud computing paradigm in order to improve the scalability, to boost the performance and to maintain the security and privacy of whole systems. We first define the system procedures for our new model. Contrary to the previous models [18–25, 27], we have an additional oracle that an adversary can query the cloud system. Then, the adversary classes are described and we give our security and privacy definition. Moreover, in order to illustrate our model, we propose two different RFID authentication protocols as case studies. We prove that the first proposal is destructive private and the second proposal is narrow-strong private according to our model. Both protocols are used to authenticate tags without violating privacy of the tag owner against the cloud owner but the tag related data are stored in the cloud in a encrypted form. Therefore, we finally present an efficient private information retrieval mechanism based on single

keyword search in order to retrieve tag related data from the cloud without violating the tag anonymity against the cloud. In this search protocol, we use only hash functions and Bloom filter in order to privately retrieve tag data. We prove that our search scheme satisfies data, query, and result pattern privacy.

The rest of the chapter is structured as follows. In Section 8.1, we give the problem statement and motivation behind this study. In Section 8.2, we introduce our novel privacy model which introduces system procedures, adversary oracles and adversary capabilities. Then we describe the security and privacy definitions with respect to the adversary classes. In Section 8.3, we propose a privacy preserving RFID authentication protocol which works with a cloud service and give its security and privacy analysis. In Section 8.4, we propose a more secure privacy-preserved authentication protocol and give its security and privacy analysis. Section 8.5 gives our private single-keyword search protocol and presents its security and privacy analysis. Finally we conclude the chapter in Section 8.6.

The results presented in Chapter 8 have been published in [36] and submitted to [37].

8.1 Problem Statement and Motivation

In this section, we illustrate how cloud computing can be utilized in an RFID authentication system as a cost effective computation and storage services. This illustration helps us to examine the restrictions of the technology, the capabilities of adversaries and the challenging issues in RFID application development and deployments.

Let us describe the scenario for the system depicted in Figure 8.1. Assume

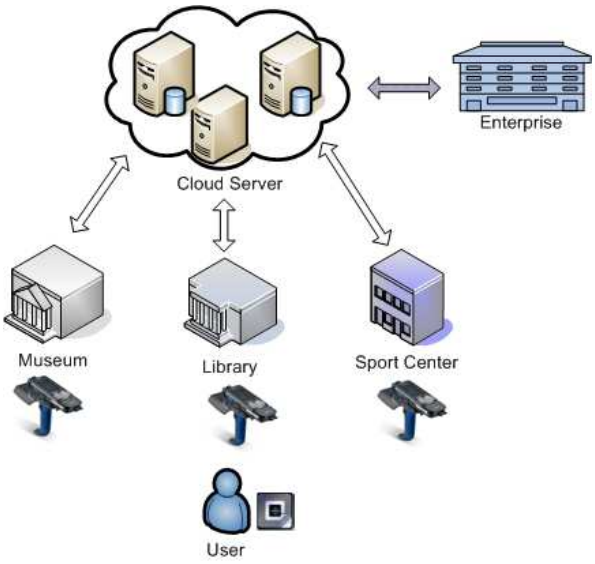

Figure 8.1: The scenario of cloud based RFID system

that we have an enterprise company that provides several social facilities (such as library, museum, sport center, etc.) that are physically placed in different areas. All the facility providers and the enterprise are connected to the cloud service via Internet. Each facility has its own access control based on the RFID system which is connected to the cloud computing. In order to benefit from some of these facilities, the clients first buy a membership from the enterprise. The enterprise company delivers an RFID membership card to its clients. Then, with the help of an RFID card, a client could use any of these facilities to authenticate itself to the centralized services.

In this scenario, all the clients' information (such as name, birthday, photo, biometric data etc.) are stored in the database of the cloud in an encrypted form. Whenever a legitimate client wants to access a facility, the facility provider will certainly identify and authenticate the person with the help of the cloud service. If the authentication protocol used between a user card and a valid reader in the facility does not consider privacy of the clients,

the cloud owner could profile and trace the user. However, for privacy of card owner the cloud should not be able distinguish transactions comes from the readers in the facilities.

Besides, after the facility provider authenticates a card, it may need the card related information such as its owner's private information. These information are stored in the cloud's database. Whenever the facility provider requests a card's information from the cloud, the privacy of the card owner is violated if the cloud is able to distinguish the request. In order to handle this issue, a Private Information Retrieval Protocol (PIR) should be run between the facility provider and the cloud service in order for retrieving tag data from the cloud while hiding the identity of the tag being retrieved.

The design of a secure privacy-preserving RFID authentication protocols rely on an accurate security and privacy analysis. In the literature, several models have been proposed to formalize security and privacy in the context of RFID system; however, none of them considers this scenario.

8.2 Our Privacy Model

Our privacy model borrows and extends the concepts from previous models [18,22]. Contrary to [22], we consider an RFID system consists of a cloud service, multiple tags, multiple readers where a tag and a reader carry out an identification protocol with the help of the cloud service. Each tag stores a state, the cloud keeps a database of all valid tags. Namely, the cloud is the central back-end server which is connected to multiple readers. A reader authenticates tags with the help of the cloud. Adversaries are allowed to interact with all tags and readers and the cloud. Our model is similar to the classical RFID model with many tags, many readers and a back-end server.

The main difference between our model and the classical model is that in our model the privacy of tag owner against the back-end server's owner is also taken into account. Moreover, the tag related information such as tag owner's information, photos, etc, are stored only in the database of the cloud but not in the reader. This information is stored in an encrypted form and the cloud cannot decrypt this.

In our model, we do not consider the physical characteristics of the radio links, which are studied in [125]. For privacy, we consider only the content of the exchanged messages between tags, readers, and the cloud. In this section, we first present the system procedures and the oracles that an adversary can query. Then, the adversary classes are described. Finally, we define our security and privacy definitions.

8.2.1 System Procedure

Throughout the chapter we use similar the oracle definitions introduced in [18, 126]. An RFID scheme is defined with the following procedures.

- SetupCloud(1^ℓ) : This algorithm first produces a public-private key pair (K_{C_P}, K_{C_S}) for cloud service where ℓ is the security parameter, then initializes its database $\mathcal{DB}$.

- SetupReader(1^ℓ) : This algorithm produces a public-private key pair (K_{R_P}, K_{R_S}) for reader where ℓ is the security parameter, then stores its secrets in its non-volatile memory.

- SetupTag$_{K_P}(ID)$: This algorithm generates a tag secret K and the initial state S of a tag with identifier ID. If this tag is legitimate, the pair (ID, K) is inserted into the database.

- IDENT: An interaction protocol between a tag and the reader to complete the authentication transcript.

Experiment $Exp^{b}_{\mathcal{S},\mathcal{A}}[k,\mathcal{P}]$:

1. Training phase
    - $\mathcal{A}$ may perform any number of oracles, limited by its class $\mathcal{P}$.
2. Challenge phase
    - $\mathcal{C}$ initializes the system, chooses a random bit b, and SETUPREADER(1^k) and sends $\mathcal{S}$'s public parameters to $\mathcal{A}$.
    - $\mathcal{A}$ interacts with the whole system, limited by its class $\mathcal{P}$.
    - $\mathcal{A}$ outputs a guess bit b'.

$Exp^{b}{}_{\mathcal{S},\mathcal{A}}$ is successful if b $\stackrel{?}{=} b'$.

Figure 8.2: Experiment for privacy of Hermans et al.

8.2.2 Adversary Oracles

Privacy is defined as a distinguish-ability game (or experiment Exp) between a challenger and an adversary. This game is defined as follows. First of all, the challenger picks a random challenge bit b and then sets up the system $\mathcal{S}$ with a security parameter k. Next, the adversary $\mathcal{A}$ can interact with the RFID system by the help of following generic oracles. First of all, $\mathcal{A}$ creates a new tag of identifier $\mathrm{ID}_{\mathcal{T}}$. Then, $\mathcal{A}$ interacts with following two collections of oracles.

Definition 21. *(Adversary Oracles-I)*

- LAUNCH$()\rightarrow \pi$: *It makes the reader $\mathcal{R}$ start a new Ident protocol transcript π.*

- CreateTag$(ID_{\mathcal{T}})$: *It creates a free tag* $\mathcal{T}$ *with a unique identifier* $ID_{\mathcal{T}}$ *by using* $\boldsymbol{SetupTag}_{K_{C_P}}$. *It also inserts* $\mathcal{T}$ *into* $\mathcal{DB}$.

- DrawTag$^b(\mathcal{T}_i, \mathcal{T}_j) \rightarrow vtag$: *on input a pair of tag references, this oracle generates a virtual tag reference, as a monotonic counter, vtag and stores the triple* $(vtag, \mathcal{T}_i, \mathcal{T}_j)$ *in a table* $\mathcal{D}$. *Depending on the value of* b, $vtag$ *either refers to* $\mathcal{T}_i$ *or* $\mathcal{T}_j$. *If* $\mathcal{T}_i$ *is already references as the left-side tag in* $\mathcal{D}$ *or* $\mathcal{T}_j$ *as the right-side tag, then this oracle also returns* $\perp$ *and adds no entry to* $\mathcal{D}$. *Otherwise, it returns vtag.*

- SendReader$(m, \pi) \rightarrow m'$: *This sends the message* m *to the reader* $\mathcal{R}$ *in the protocol transcript* π *and outputs the response* m'.

- SendCloud$(m, \pi) \rightarrow m'$: *This sends the message* m *to the cloud* $\mathcal{C}$ *in the protocol transcript* π *and outputs the response* m'.

- SendTag$(m, vtag)_b \rightarrow m'$: *on input vtag, this oracle retrieves the triple* $(vtag, \mathcal{T}_i, \mathcal{T}_j)$ *from the table* $\mathcal{D}$ *and sends the message* m *to either* $\mathcal{T}_i$ *(if* $b = 0$*) or* $\mathcal{T}_j$ *(if* $b = 1$*). It returns the reply from the tag* (m'). *If the above triple is not found in* $\mathcal{D}$, *it returns* $\perp$.

- Free$(vtag)_b$: *on input vtag, this oracle retrieves the triple* $(vtag, \mathcal{T}_i, \mathcal{T}_j)$ *from the table* $\mathcal{D}$. *If* $b = 0$, *it resets the tag* $\mathcal{T}_i$. *Otherwise, it resets the tag* $\mathcal{T}_j$. *Then it removes the entry* $(vtag, \mathcal{T}_i, \mathcal{T}_j)$ *from* $\mathcal{D}$. *When a tag is reset, its volatile memory is erased. The non-volatile memory, which contains the state* S, *is preserved.*

- Corrupt$(T_i) \rightarrow S$: *It returns volatile and non-volatile memory of the tag* T_i.

- Result$(\pi) \rightarrow x$: *When* π *completes, returns* $x = 1$ *if the tag is identified,* $x = 0$ *otherwise.*

In our model, we also define two another oracles as follows.

Definition 22. *(Adversary Oracles-II)*

- CORRUPT*(*$\mathcal{R}_i$*)*$\rightarrow S$ *: It returns volatile and non-volatile memory of the reader* $\mathcal{R}_i$*.*
- CORRUPT*(Cloud)*$\rightarrow S$ *: It returns volatile and non-volatile memory of the cloud.*

The advantage of the adversary $Adv_{\mathcal{S},\mathcal{A}}(k)$ is defined as:

$$\left|Pr\left[Exp^0_{\mathcal{S},\mathcal{A}}(k)=1\right]-Pr\left[Exp^1_{\mathcal{S},\mathcal{A}}(k)=1\right]\right|.$$

8.2.3 Privacy Classes

Contrary to previous models proposed in the literature, we consider two types of adversaries such as insider and outsider adversaries. The cloud is expected to be the insider adversary who runs the protocol between a legitimate reader and itself correctly, but might save the messages to distinguish the tags. Namely, the cloud is honest but curious during its protocol runs. However, for the outsider adversaries, similar to Vaudenay privacy class [18], we introduce four privacy classes of polynomial-time bounded adversaries, determined by $\mathcal{A}$'s access to RESULT or CORRUPT oracles. These classes are formally defined as follows.

Definition 23. *(Adversary Classes) An adversary* $\mathcal{A}$ *is a p.p.t. algorithm which has arbitrary number of accesses to either the oracles described in Definition 21 or the oracles described in Definition 22.*

- ***Insider*** $\mathcal{A}$ *cannot access to any oracles except* CORRUPT*(Cloud) oracle described in Definition 22.*

- ***Weak*** $\mathcal{A}$ *uses only the oracles given in Definition 21 except* CORRUPT(T_i) *oracle.*
- ***Destructive*** $\mathcal{A}$ *uses only the oracles given in Definition 21 but cannot use any oracle on a tag after using* CORRUPT(T_i)*.*
- ***Strong*** $\mathcal{A}$ *uses only the oracles given in Definition 21 without any restrictions.*
- ***Narrow*** $\mathcal{A}$ *has no access to* RESULT *oracle.*
- ***Wide****:* $\mathcal{A}$ *has access to* RESULT *oracle.*

Strong	$\Rightarrow$	Destructive	$\Rightarrow$	Weak		Active Insider
$\Downarrow$		$\Downarrow$		$\Downarrow$		$\Downarrow$
Narrow Strong	$\Rightarrow$	Narrow Destructive	$\Rightarrow$	Narrow Weak		Passive Insider

Table 8.1: The adversary classes

Remark 12. *In a real-life system,* ***Insider*** *adversary make sense when the RFID system owner would like to outsource his/her services to a cloud. In this attack, the cloud is able to access all the data and can analyze any interactions with itself. Therefore, the system owner may want his/her system to be secure against this attack.*

According to the capability of the attacker **Insider** adversary could be two types: passive and active.

Definition 24. *(Passive Insider Adversary) A passive* ***Insider*** *adversary is one who follows the protocol and does not modify any data but is curious to get some information and may keep all the data and its intermediate computations. In case the adversary is the cloud owner then one may call the cloud owner as semi-honest party.*

Definition 25. *(Active Insider Adversary) An active* ***Insider*** *adversary is one who covers the passive adversary and can actively modify the local data or internal computations. In case the adversary is the cloud owner then one may call the cloud owner as malicious party.*

We also define X^+ and X^* privacy notion variants, where X refers to the basic privacy notion. $^+$ refers to the notion that arises when the adversary has also access to CORRUPT($\mathcal{R}$) oracle. But * refers to the notion that arises when the capabilities of the adversary are further restricted with respect to CORRUPT oracle. The restricted CORRUPT oracle will only return the non-volatile state of the corrupted party (tag, reader or the cloud) but not the volatile memory state. With this restriction, we exclude trivial privacy attacks on multi-pass protocols in which the tags are required to store some information in volatile memory during the session of the protocols.

8.2.4 Notion of Security and Privacy

Definition 26. *(Correctness) An RFID scheme is correct if the identification of a legitimate tag only fails with negligible probability with respect to system's security parameter.*

Definition 27. *(Tag Authentication) An RFID system achieves tag authentication if for every strong adversary and for every tag in the system, the probability of attacker's impersonating any tag is at most negligible. The adversary may interact with the tag they want to impersonate. The adversary can corrupt all tags but not the impersonated tag.*

Definition 28. *(Privacy [22]). A privacy preserving protocol, modeled by an RFID system $\mathcal{S}$, is said to computationally provide privacy notion X, provided*

that for all polynomially bounded adversaries $\mathcal{A}$, it holds that $Adv^{X}_{\mathcal{S},\mathcal{A}}(k) \leq \epsilon$, for negligible ϵ.

8.3 The First Authentication Protocol

Our first case study protocol is based on low-cost symmetric primitives such as Physically unclonable functions and hash functions. We treat hash functions as random oracles. Namely, the function $\mathcal{H}$ responds to every query with a truly random response chosen uniformly from $\{0,1\}^{\alpha}$. The function always gives the same response for a given input word. Moreover, our first protocol also uses PUF function described Definition 14. In our proposal, the PUF function has the following mapping: P:$\{0,1\}^{\alpha} \rightarrow \{0,1\}^{\alpha}$.

8.3.1 The Protocol

In this section, we first give a brief explanation about how to distribute the secrets for each party in the RFID system. Then we present the steps of the authentication protocol.

Let $\mathcal{I}$ be a trusted issuer who sets up the system parameters and the secrets of each party. $\mathcal{I}$ first selects a random master secret $S \in_R \{0,1\}^{\alpha}$ and creates a counter ctr which is initially set to zero. The cloud stores the master secret S and the counter ctr. Integration of a reader into system is very simple by just sending a triple $(ID_R, S_R = \mathcal{H}(S, ID_R, ctr), ctr)$ to the reader via a secure channel. $\mathcal{I}$ defines a group size (say l) and creates a counter g which specifies the order of the group a tag belongs to. During the registration of a tag $\mathcal{T}_i$, $\mathcal{I}$ first selects a random unique $ID_i \in_R \{0,1\}^{\alpha}$, and a random challenge G_i and computes the masked master secret $MS_i \leftarrow S \oplus P_i(G_i)$ and specifies the order of the tag g_i and set its counter $ctr_i \leftarrow 0$. $\mathcal{T}_i$ stores the

values $(MS_i, ID_i, G_i, g_i, ctr_i)$.

The protocol steps are depicted in Figure 8.3. When a reader (e.g. NFC) $\mathcal{R}$ is connected to the cloud, the cloud sends a triple $S_R \leftarrow H(S, ID_R, ctr)$, $ID_R \in_R \{0,1\}^\alpha$ and ctr to the reader via secure channel. When a tag $\mathcal{T}$ comes in the range of the reader, the reader first chooses a random number $a \in_R \{0,1\}^\alpha$ and sends the triple (a, ID_R, ctr) to $\mathcal{T}$. Then, $\mathcal{T}$ first generates four random nonce $mt_1 \leftarrow_R \{0,1\}^\alpha$, $mt_2 \leftarrow_R \{0,1\}^\alpha$, $mt_3 \leftarrow_R \{0,1\}^\alpha$, $b \leftarrow_R \{0,1\}^\alpha$. $\mathcal{T}$ evaluates the PUF P_i with G_i and XOR it with MS_i to recover master key $S \leftarrow P_i(G_i) \oplus MS_i$. Then, $\mathcal{T}$ computes the session secret $K_g \leftarrow \mathcal{H}(\mathcal{H}(S, ID_R, ctr), g_i)$. Then, $\mathcal{T}$ computes $m_1 \leftarrow \mathcal{H}(K_g, a, b, 1)$, $m_2 \leftarrow \mathcal{H}(K_g, a, b, 2) \oplus ID_i$,$m_3 \leftarrow \mathcal{H}(K_g, ID_i, a, b, 1)$. $\mathcal{T}$ checks whether ctr is greater or equal to its counter ctr_i. If $ctr < ctr_i$, $\mathcal{T}$ sets $m_i \leftarrow mt_i$ for $i = 1, 2, 3$. $\mathcal{T}$ finally sends (b, m_1, m_2, m_3) to the reader. $\mathcal{T}$ deletes S from memory. Upon receiving m_1, m_2, m_3 and b, for all possible value of g, $\mathcal{R}$ computes $m_1' \leftarrow \mathcal{H}(\mathcal{H}(S_R, g), a, b, 1)$ to find a match $m_1' \stackrel{?}{=} m_1$. If a match is found, then $\mathcal{R}$ derives $ID_i' \leftarrow \mathcal{H}(\mathcal{H}(S_R, g), a, b, 2) \oplus m_2$. $\mathcal{T}$ also checks whether the integrity of ID_i' is protected by simply checking the equality of $m_3 \stackrel{?}{=} \mathcal{H}(K_g', ID_i', a, b, 1)$. Now, If every steps are on the right line, $\mathcal{R}$ authenticates $\mathcal{T}$. $\mathcal{R}$ finally calculates $m_4 \leftarrow \mathcal{H}(K_g', ID_i', a, b, 2)$ and sends it to $\mathcal{T}$. $\mathcal{T}$ checks whether both conditions hold $ctr > ctr_i$ and $\mathcal{H}(K_g, ID_i, a, b, 2) \stackrel{?}{=} m_4$. If these conditions hold, then $\mathcal{T}$ updates its counter $ctr_i \leftarrow ctr$. Finally, $\mathcal{T}$ deletes K_g from the memory. The last messages sent by the NFC is for updating counter in case of the fact that a facility is closed down.

Remark 13. *Note that whenever a strong adversary tries to apply a physical attack on a target tag, she cannot reach either the valid secret K_g or the valid master secret S. In order to achieve a micro-probing attack on the tag, she should first make a hole on the coating by using Focused Ion Beam. In this*

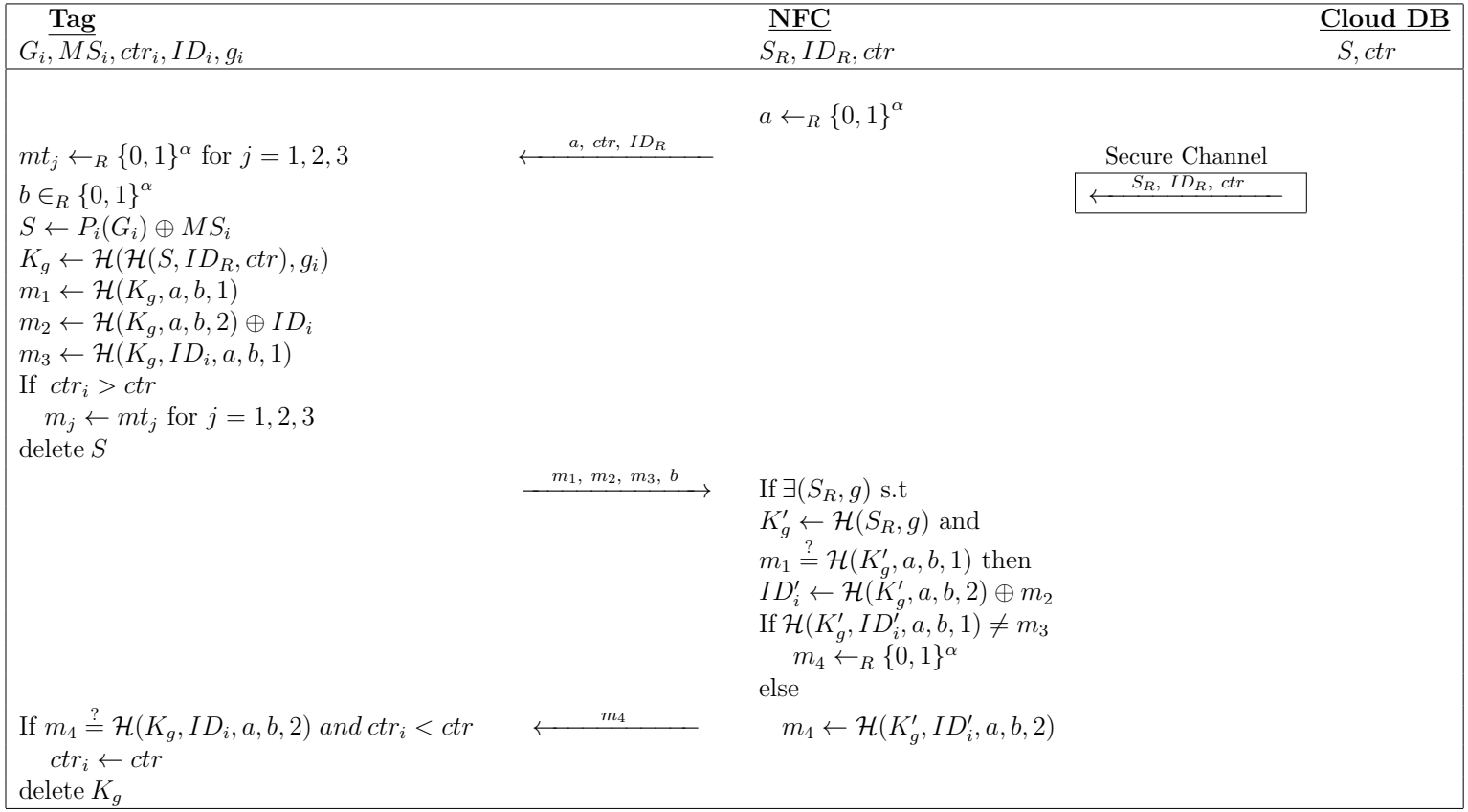

Figure 8.3: A destructive private authentication protocol^{+*}

case, the structure of the PUF most probably gets a damage that the response of the PUF would be very high level noisy and the PUF control will detect such level of noise and destroys the PUF. The response will not be valid and the master secret S and the session key K_g will not be computed correctly.

Remark 14. *When a reader is compromised or a facility closed down, the cloud increments its counter $ctr = ctr + 1$. Then, for each existing NFC R, the cloud computes $S_R = H(S, ID_R, ctr)$ and sends the triples $(S_R,\ ID_R,\ ctr)$ to the reader.*

After the reader authenticating the tag, the reader will run a Private Information Retrieval (PIR) protocol which is explained in Section 8.5.

8.3.2 The Security and Privacy Analysis

In this section, we provide the security and privacy analysis of the protocol depicted at Figure 8.3.

Remark 15. *Throughout this section, one can assume that there is one reader and many tags in the system. There is no loss in the generality with this assumption. To see that, for fixed a and b values, different ID_R values produce different K_g values. However, all these K_g values have same randomness (they are indifferent) in the view of the adversary. Thus, the adversary cannot distinguish whether only one or more readers are used in the system. Hence, one NFC is enough for the analysis. Moreover, we use a slightly enhanced version of* CreateTag *oracle in the proof of the privacy by adding extra parameter to the function which specifies the group of the tag.*

Theorem 22. *The proposed protocol depicted in Figure 8.3 satisfies tag authentication against destructive adversary.*

Proof. The proof is pretty trivial. Note that the adversary cannot get the values of either K_g or S regardless of how many tags she is allowed to use or corrupt. Moreover, by Definition 27 the adversary is not allowed to corrupt the target tag. It is a so low probability that the adversary get the ID of the target tag. Even if this event is realized, the adversary's producing correct m_3 value is at most negligible since reader sends the challenge values a randomly. Thus, the system satisfies tag authentication. □

Theorem 23. *The proposed protocol depicted in Figure 8.3 satisfies destructive privacy.*

Proof. The only way for adversary to destroy the privacy is to choose right tags from the same group and left tags from different groups and to expect having the same response to a specified challenge value. First of all, the adversary creates two tags by calling T_1 =CreateTag$(ID_1, 0)$ and T_2 =CreateTag$(ID_1, 1)$ oracles. Then she applies $vtag_1$ =DrawTag(T_1, T_2) and uses SendTag$(a, ctr, ID_R, vtag_1)$ for l times and stores the answers ${m^i}_{1_1}, {m^i}_{2_1}, {m^i}_{3_1}, {b^i}_1$ where $i \in \{1, \ldots, l\}$. Similarly, the adversary creates another two tags by calling T_3 =CreateTag$(ID_3, 0)$ and T_4 =CreateTag$(ID_4, 2)$ oracles. Then she applies $vtag_2$ =DrawTag(T_3, T_4) and uses SendTag$(a, ctr, ID_R, vtag_2)$ for k times and stores the answer of the ${m^j}_{1_2}, {m^j}_{2_2}, {m^j}_{3_2}, {b^j}_2$ where $j \in \{1, \ldots, k\}$. If ${b^{i_0}}_1 = {b^{j_0}}_2$ for some i_0 and j_0 but ${m^{i_0}}_{1_1} \neq {m^{j_0}}_{2_1}$ then the answer is the right tags. Otherwise the answer is the left tags. The probability of having wrong result after these observations is negligible. Note that the adversary does not need to create more tags as described above since having more protocol runs with these two tag groups has the same effect of creating new tags and having protocol rounds for the adversary. Therefore,

with given parameters the success probability of the adversary is

$$1 - \prod_{i=0}^{k-1}(1 - \frac{l}{2^\alpha - i}).$$

Let $P = \prod_{i=0}^{k-1}(1 - \frac{l}{2^\alpha - i})$, then

$$ln(P) = \sum_{i=0}^{k-1} ln(1 - \frac{l}{2^\alpha - i}) \approx -\sum_{i=0}^{k-1} \frac{l}{2^\alpha - i} > \frac{(k-1)l}{2^\alpha}.$$

So,

$$1 - P < 1 - e^{\frac{(k-1)l}{2^\alpha}}.$$

Note that, the probability above is negligible as k, l are polynomially bounded in α. Thus, the proposed protocol satisfies destructive privacy. □

Theorem 24. *The proposed protocol depicted in Figure 8.3 is resistant against passive insider adversary according to Definition 24.*

The correctness of the last theorem is obvious as the cloud does not even know whether NFC has a protocol transaction with any tag at a specified time. In this protocol, the role of the cloud is just initialize the reader for ctr and ID_R values.

8.3.3 The Protocol Enhancement

It is clearly seen that our protocol does not provide security against the adversary who reaches the volatile memory of the tag as soon as the secret S is constructed from the PUF evaluation. In [30], Kardas et al. proposed an approach for splitting key into parts and each part can be constructed

from the PUF evaluation. The authors also proved that the malicious adversary cannot reach both parts of the secret. This makes their protocol secure against the adversary who has access to volatile memory of the victim tag. Therefore, when we use the same approach for constructing the secrets digested in hash, our protocol satisfies destructive privacy.

8.4 The Second Authentication Protocol

Our second proposed protocol is based on Elliptic Curve Cryptography (ECC) and we work on the additive property of ECC.

8.4.1 The Proposed Protocol

In this section, we first give a brief explanation about how to set up the secrets for each party in the RFID system, then we present the identification protocol.

Let $\mathcal{I}$ be the trusted issuer which sets up the system parameters and the secrets of each party. $\mathcal{I}$ first constructs the elliptic curve and selects a generator P. Then $\mathcal{I}$ generates a random private key y for the cloud and computes the corresponding public key $Y = yP$. $\mathcal{I}$ also generates a random unique private key n for each NFC and computes the corresponding public key $N = nP$. For each tag, $\mathcal{I}$ selects a random unique identifier id and also computes the ECDSA signature pair (r, s) on the $ID_x = xcoord(idP)$. Note that, the secrets of the tag are id, r, s, the secret of reader is n, the secret of the cloud is y. On the other hand, the public values of the tag are Y, P, the public values of the reader and the cloud are N, Y, P. Moreover, tag related other information are stored on one or more independent clouds.

In our proposal, the cloud can distinguish whenever a NFC reader is cor-

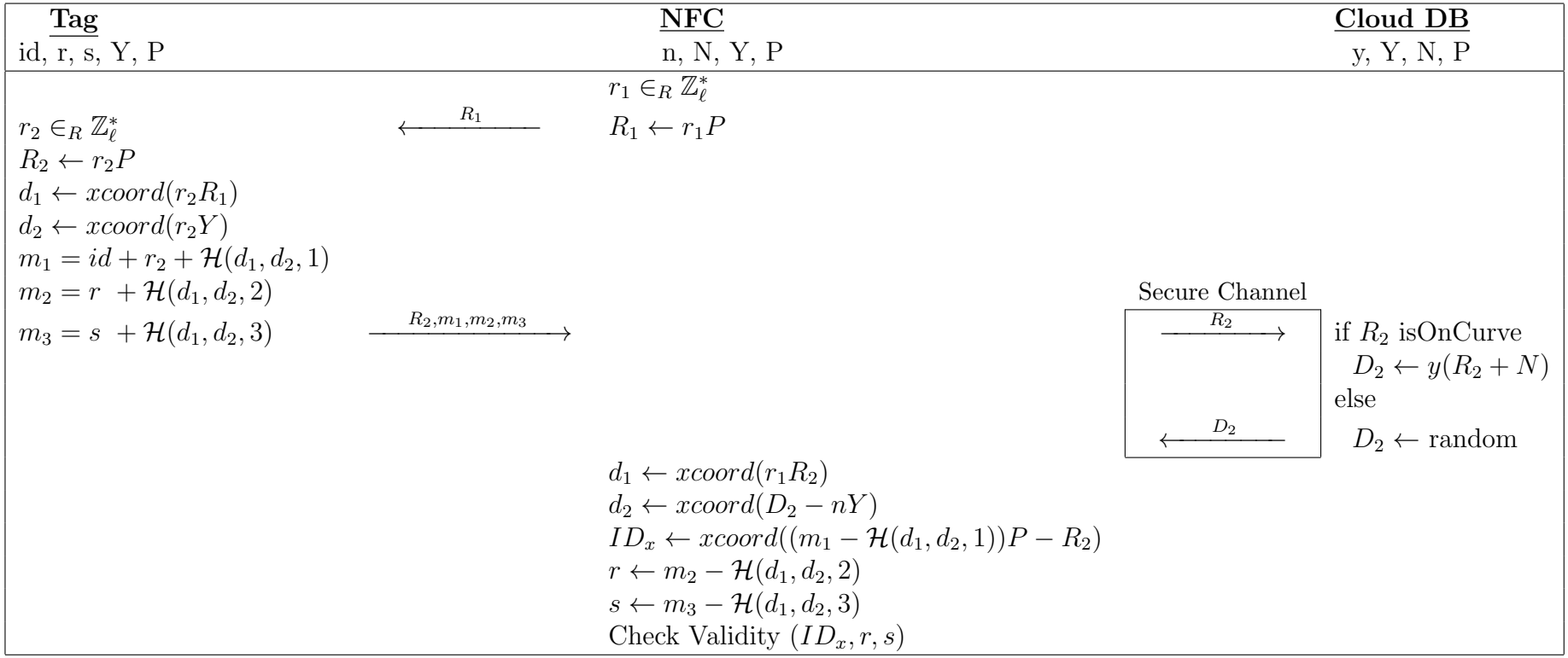

Figure 8.4: A narrow strong private authentication protocol^{+*}

rupted or simulated and disallow any interactions from the corrupted reader. Moreover, the cloud server and NFC have some sort of authentication mechanism such that the transactions between NFC and the cloud is not observed by the adversary (It can be thought that NFC and the cloud server have secure channel like SSL/TLS connection).

An overview of the proposed protocol is given in Figure 8.4. First of all, the reader $\mathcal{R}$ generates a random number r_1 which is used for soundness and ensuring privacy. $\mathcal{R}$ computes a point on the elliptic curve with r_1 ($R_1 = r_1 P$), then $\mathcal{R}$ sends it to tag $\mathcal{T}$. $\mathcal{T}$ verifies that $R_1 = \mathcal{O}$, the point at infinity and chooses a random number r_2 and calculates $R_2 = r_2 P$. Then, $\mathcal{T}$ computes $d_1 = xcoord(r_2 R_1)$, $d_2 = xcoord(r_2 Y)$, $m_1 = id + r_2 + \mathcal{H}(d_1, d_2, 1)$, $m_2 = r + \mathcal{H}(d_1, d_2, 2)$, and $m_3 = s + \mathcal{H}(d_1, d_2, 3)$ and sends R_2, m_1, m_2, m_3 to the reader. The reader sends R_2 to the cloud and the cloud computes $D_2 = y(R_2 + N)$ and sends to the reader. After that $\mathcal{R}$ computes $d_1 = xcoord(r_1 R_2)$, $d_2 = xcoord(D_2 - nY)$, $ID_x = xcoord((m_1 - \mathcal{H}(d_1, d_2, 1))P - R_2)$,$r = m_2 - \mathcal{H}(d_1, d_2, 2)$ and $s = m_3 - \mathcal{H}(d_1, d_2, 2)$. Finally, the reader checks whether the signature pair (r, s) is the valid signature on ID_x by using ECDSA verification algorithm.

Remark 16. *Any *-adversary(weak* , destructive* or strong*) can never see the cloud server's replies for queried R_2 values. The reason for this claim is, if the adversary does not corrupt the reader, then since NFC and the cloud server have secure channel, the adversary can not observe D_2 values. Moreover, if the adversary corrupts the reader, then due to the detection argument, the cloud server does not return any reply to the adversary.*

Similar to the protocol in Section8.3, after the reader verifying the signature pair (r, s) on ID_x for the tag, the reader will run a PIR protocol with the cloud, which is explained in Section 8.5., in order to get the tag related

information.

8.4.2 The Security and Privacy Analysis

In this section, we are going to provide the security and privacy analysis of the protocol depicted in Figure 4.

Theorem 25. *The proposed protocol is correct according to Definition 26*

Proof. Let T be a valid tag with the identifiers id, r, s, Y and let reader sends R_1 and the tag produces r_2 as a nonce at a protocol run. The correctness of the protocol can be shown by following arguments.

$$\begin{aligned}
d_1 &= xcoord(r_2 R_1) = xcoord(r_1 R_2) = d_1 \\
d_2 &= xcoord(D_2 - nY) = xcoord(yR_2 + yN - nY) \\
&= xcoord(r_2 Y) = d_2 \\
ID_x &= xcoord((m_1 - \mathcal{H}(d_1, d_2, 1))P - R_2 \\
&= xcoord((id + r_2)P - R_2) = xcoord(idP) \\
r &= m_2 \mathcal{H}(d_1, d_2, 2) = r + \mathcal{H}(d_1, d_2, 2) \\
&- \mathcal{H}(d_1, d_2, 2) = r \\
s &= m_3 \mathcal{H}(d_1, d_2, 3) = r + \mathcal{H}(d_1, d_2, 3) \\
&- \mathcal{H}(d_1, d_2, 3) = s.
\end{aligned}$$

Thus, if a tag is valid, then after a successful protocol run, reader successfully authenticates the tag. □

Theorem 26. *Let $\mathcal{A}$ be a strong adversary$^+$. Then, $\mathcal{A}$ cannot steal the all secret values of a tag, id, r, s, if the tag remains uncorrupted.*

Proof. Since the secrets of NFCs are not used in tag side calculations, without loss of generality we can assume that there is only one NFC in the system. This assumption does not result in loss of the generality as one can regain the advantage loss due to having one NFC for analysis by running more protocols on the NFC to be used for analysis. Let us fix a tag T and let $W = \{T_0, T_1, \ldots, T_k\}$ be the set of other tags in the system where k is polynomially bounded in l, where l is the security parameter. Let the adversary does not apply the CORRUPT oracle to T and she tries to figure out the secrets of this tag. However, the adversary can apply any number of CORRUPT oracle to tags in set W. First, may be the most remarkable observation is the adversary does not need to deal with tags in the set W to gain secrets of the target tag since the tag related secrets are not relevant to other tags' secrets in a deterministic way and if the adversary applies some oracle in the set W, the only useful information for her to get some (r_1, R_1) and (r_2, R_2) pairs. However, the adversary can get the same amount of information by preparing more (r_1, R_1) and (r_2, R_2) pairs beforehand or having more protocol run between the tag and the NFC. Therefore, tag authentication of target tag is not related to the number of tags in the system, but it is related to the number of pairs she prepares beforehand and protocol transactions she can observe or commit with the target tag.

Note that, if the adversary applies CORRUPT oracle on the reader, the adversary may get the values of d_1, d_2, ID_x, r and s. However, knowledge of the value of these parameters is not enough for adversary to figure out the value of id. To get the value of id, the adversary has to figure out the value of the chosen r_2 value at least one protocol transaction. Therefore, the adversary creates a (r_2, R_2) pairs before starting the attack. Then, the adversary uses SENDREADER(π) oracle for b times to initiate protocol run between the

NFC and the target tag, where a and b are polynomially bounded in l. Note that, the adversary does not need to know the value of r_1, that is why she does not use $SendTag(R_1)$ command for precomputed r_1 values. Therefore, the probability for the adversary to get the value of r_2 at least one protocol transaction is less than

$$(1 - \frac{a}{\#E - b})^b \approx e^{\frac{-ba}{\#E-b}},$$

where $\#E$ represents the order of the point P (as P generator of the curve, then $\#E$ represents the number of the points on the curve). Since the values a and b are polynomially bounded in l, then the probability is negligible. □

Corollary 3. *The proposed protocol depicted in Figure 8.4 satisfies tag authentication against strong adversary$^+$.*

Proof. This corollary is direct consequence of Theorem 26. □

Lemma 5. *Let $\mathcal{A}$ be a strong adversary$^+$$*$. Then, $\mathcal{A}$ can not get the value of D_2 at any protocol transaction. Moreover, the adversary can never figure out the cloud secret y.*

Proof. Remark 16 states the first fact that the adversary can never see the cloud server's replies for queried R_2 values. This deduction directly implies that the adversary can not get the value of y. However, even if the adversary receives some legitimate (R_2, D_2) pairs, then the adversary has polynomially bounded number of discrete logarithm problem for the elliptic curve under the assumption that she has the knowledge of the value of N. However by Definition 4 and Remark 1, it is infeasible for the adversary to solve this problem. Thus, the adversary gets the secret of the cloud only with negligible probability. □

Theorem 27. *The proposed protocol depicted in Figure 8.4 satisfies narrow-strong* privacy.*

Proof. Let l be a security parameter. Since the secrets of NFCs are not used in tag side calculations, without loss of generality we can assume that there is only one NFC in the system. This assumption does not result in loss of the generality as one can regain the advantage loss due to having one NFC for analysis by running more protocols on the NFC to be used for analysis.

Let A_n be a narrow-strong adversary*. Firstly, let the adversary applies CORRUPT oracle on the reader and get the values of n, N, Y, P. Since the secrets of the reader in non-volatile memory does not change protocol run to protocol run, it is enough for the adversary apply this oracle to the reader only once.

Let A_n call CREATETAG oracle two times and creates the tags T_0, T_1. Then, let the adversary call the DRAWTAG oracle to have $vtag_1$, which refers either T_0 or T_1 and applies the CORRUPT oracle for both tags to learn the secrets in these tags' non-volatile memory. Note that, it is enough for the adversary to apply CORRUPT oracle only once per tag as their secrets in the non-volatile memories do not change protocol run to protocol run.

Note that, the adversary has to learn the values of d_1 and d_2 in at least one protocol transaction to learn id, r and s values to figure out $vtag_1$ represents which tag. By Lemma 5, the adversary can not learn the cloud's secret. Thus, she has to figure out the value of r_2 value at least one protocol transaction (In this way, the adversary can calculate the value of D_2 and then the value of d_2). Therefore, let the adversary create a (r_2, R_2) pairs before applying other oracles. After that, the adversary only applies SENDTAG$(vtag_1, R_1)$ oracle for a fixed point R_1 on the curve for p_1 times, where p_1 is polynomially bounded in l. The reason for the adversary only applying one oracle is that

different R_1 values has no effect at destroying privacy and applying oracles on reader does not give any advantage to the adversary. Then, the advantage of the adversary to destroy the privacy is bounded above by

$$1 - (1 - \frac{a}{\#E - p_1})^{p_1} \approx 1 - e^{\frac{-p_1 a}{\#E - p_1}}.$$

since, p_1 is polynomially bounded in l, the probability stated above is negligible. Thus, creating just two tags is not enough for the adversary.
Now, let the adversary creates two more tags T_2, T_3 and applies DrawTag oracle to get $vtag_2$. Similarly, the adversary only uses SendTag$(vtag_1, R_1)$ oracle for the same R_1 point for p_2 times, where p_2 is polynomially bounded in l. In this case, the analysis of the adversary's advantage to destroy the privacy for just these two tags T_2 and T_3 is similar to the analysis of the adversary's advantage to destroy the privacy for tags T_0 and T_1. However, the adversary has more tools. If one of the R_2 value returned from $vtag_2$ is equals one of the R_2 value returned from $vtag_1$, then adversary also breaks the privacy. Thus, the total advantage of the adversary is less than

$$\begin{aligned} &2 - (1 - \frac{a}{\#E - (p_1 + p_2)})^{p_1+p_2} + (1 - \frac{p_1}{\#E})^{p_2} \\ &\approx 2 - e^{\frac{-(p_1+p_2)a}{\#E-(p_1+p_2)}} - e^{\frac{-(p_1 p_2)}{\#E}}. \end{aligned}$$

For the sake of generalization, let the adversary create $2k - 4$ more tags and as a total has k $vtag$ reference and let she follows the same steps as described above paragraphs of this proof. Let $M = p_1 + \ldots + p_k$ and $T =$

$\max\{p_1, \ldots, p_k\}$, then the total advantage of the adversary is less than

$$\begin{aligned}
&\binom{k}{2} + 1 - (1 - \frac{a}{\#E - (M)})^M + \binom{k}{2}(1 - \frac{T}{\#E})^T \\
&\approx \binom{k}{2} + 1 - e^{\frac{-Ma}{\#E-M}} - e^{\frac{-(T^2)}{\#E}}.
\end{aligned}$$

The probability above is negligible as a, M, T are polynomially bounded in l. Thus, the proposed protocol satisfies narrow-strong* privacy. □

Theorem 28. *The proposed protocol depicted in Figure 8.4 is resistant against passive insider adversary according to Definition 24.*

Proof. According to Definition 23, the insider adversary A_I is only allowed to use CORRUPT(Cloud), so she cannot learn tag related secrets and NFC secrets. Therefore, in terms of insider adversary, the only privacy concern is link-ability. Thus, we play the following game with the adversary. Let there be two tag, T_0 and T_1, the oracle $\mathcal{O}$ chooses $b \in_R \{0, 1\}$, and the tag T_b has p protocol transaction, after that the oracle chooses $b' \in_R \{0, 1\}$ and $T_{b'}$ has k protocol transaction. After that step, the adversary returns 1 if she believes $b == b'$, and returns 0 otherwise. If her guess is correct, she destroys the privacy, otherwise we conclude that the system satisfies privacy against insider attacks.

Let before starting play the game, A_I prepares S (r_1, R_1) pairs and H (r_2, R_2) pairs. Then $\mathcal{O}$ chooses $b \in_R \{0, 1\}$, $b' \in_R \{0, 1\}$, the oracle and the adversary plays the game described above. Before returning the guess, the adversary analyzes the followings: If any of R_1 point or R_2 point in these p transactions is equal to the any of R_1 or R_2 points prepared before the game started by the adversary, then she destroys the privacy, as in each transaction, she knows the value of d_2 and if the above condition satisfied, the she also learns d_1 value of an protocol ran. Thus, she learns the r and s secret of T_b, so she can link

this tag's transactions. If this is not the case, similar to the above approach, if any of R_1 point or R_2 point in k transactions with $T_{b'}$ is equal to the any of R_1 or R_2 points prepared before the game started by the adversary, then she destroys the privacy. Moreover, if any of chosen R_1 by NFC in k transactions is equal to any of chosen R_1 by NFC in p transactions, she again destroys the privacy. If any of mentioned analysis does not work, the adversary flips a coin, and returns her guess. Therefore, the advantage of the adversary is bounded above by

$$\frac{1}{2} + 3 - ((1 - \frac{S}{\#E})(1 - \frac{H}{\#E}))^p + ((1 - \frac{S}{\#E})(1 - \frac{H}{\#E}))^k + (1 - \frac{p}{\#E})^k \approx \frac{1}{2} + 3 - (e^{-\frac{pH+S}{\#E}} + e^{-\frac{kH+S}{\#E}} + e^{-\frac{kp}{\#E}}).$$

Since, S, H, p and k are polynomially bounded in l, then A_I's advantage to destroy the privacy is negligible. Hence, the system is resistant against insider adversaries. □

8.4.3 Performance Considerations

Our proposal requires only one-way hash functions, scalar-ECC point multiplications and the generation of a random number. In order to work on 80-bit security level, the elliptic field size should be at least 160-bits. We can implement our proposal in one of the recent ECC architectures [127, 128]. The architecture [127] for ECC coprocessor needs less than 15 kGE consumes 13, 8 μW of power and takes around 85 ms for one scalar-ECC point multiplication [127]. Wenger and Hutter [128] proposed an ECC coprocessor that only needs 9 kGEs, consumes 32, 3μW of power and requires about 286 ms for one scalar-EC point multiplication. For the implementation of hash functions, in architecture of [129], we need 330 operation clocks for one hash function of

160-bit data and 19.5 μW power consumption at 100 kHz operation clock.

8.5 Private Information Retrieval: Private Keyword Search

In the previous sections, we proposed two identification and authentication mechanisms. In these protocols, the readers are able to authenticate RFID tags and gets their ID value. The readers do not store tag related information such as the information about the tag owner but these data are stored in an encrypted form in the database of the cloud. After authenticating a tag, the reader needs to access the tag data with the help of the ID. During the access, the cloud service should not be able to violate the privacy of the tag owner. In other words, the query created by the reader should be randomized and the result should not directly address ID value. Motivated by this need, in this section, we first provide a related work for private information retrieval. Then, we define the privacy definitions for single-keyword search and finally propose a private and efficient keyword search based on symmetric cryptography and Bloom filter.

8.5.1 Related work

Several scientific studies have been done on private information retrieval (PIR) since the PIR problem was first formulated in [130]. PIR problem is formulated as follows. The database with n-bit string x where the user, holding some retrieval index i, wishes to learn the i^{th} data bit x_i. The trivial solution of the PIR is sending the whole database from the cloud to the user and this solution provides perfect privacy. However, in practice the size of

the database could be very large and the solution is not reasonable. Chor et al. proved that any perfectly private trivial PIR solution has a communication with lower bound greater than or equal to the database size [130]. There are two approaches in order to thwart this issue. In the first approach, the database is replicated at k number of servers that can communicate only with the user not among themselves. The servers could learn no information about the index of the retrieved item regardless of its computational power. The PIR with this setting is called *information-theoretic* PIR. In these schemes, The user constructs the queries in a such way that they give no information to the servers about the user's interest. However, using the results from the queries, the user can construct the desired record. This approach can also be extended that when up to t of the servers are allowed to cooperate against the user. These kinds of theoretical PIR approaches have been extensively researched [131–135]. The second approach is the computational PIR, which relaxes the user's privacy requirement to computational privacy, or adds the requirement of database privacy that states that the user may learn a single data item but nothing else. Chor and Gilboa in [136] first proposed a multi-server computational PIR scheme which based on one-way function. Following this work, several more efficient computational protocols, based on various hardness assumptions, were constructed [137–142].

8.5.2 The Privacy Model for Private Search

The common privacy definition for search mechanisms in private search is that the cloud learn nothing but only views search results. We establish a set of privacy requirements over this privacy definition for single private keyword search protocol. Tag related information stored in the cloud should not give any advantage to the cloud because of the data privacy. The data

privacy definition is given as follows.

Definition 29. ***Data privacy.*** *A single keyword-search protocol achieves data privacy if for all polynomially bounded adversary, given the retrieved encrypted data, learns no information about the data.*

As soon as tag owner enters a facility and the tags is identified and authenticated by the reader of the facility. When the reader needs to retrieve tag data from the cloud, it will generates a query. If the cloud can distinguish a query from others, the privacy of the tag owner would be violated. The query privacy is defined as follows.

Definition 30. ***Query Privacy.*** *A single keyword-search protocol achieves query privacy if for all polynomially bounded adversary, given the queries, learns no information about on which data (i.e. tag) the query is applied.*

On the other hand, the cloud observes the queries and the result set of the queries. These information may give some advantage to the cloud for identifying which tags is queried. The definition of the result pattern privacy is given as follows.

Definition 31. ***Result Pattern privacy.*** *A single keyword-search protocol achieves result pattern privacy if for all polynomially bounded adversary, given queries and the retrieved a set of encrypted data, learns no information about on which data (i.e. tag) the query is applied.*

8.5.3 Our Private Keyword Search

Our private keyword search system consists of four parties such as $\mathcal{DB}$, DO, CLD, $\mathcal{R}$, where $\mathcal{DB}$ is a database, DO is the owner of the database, CLD is the cloud service which provides storage and computation service, and $\mathcal{R}$ is the

set of authorized readers which can query CLD. Our private search scheme consist of building index, query generation, search and tag data retrieval phases.

Let κ be the security parameter and ℓ be block size. Each tag has a unique ID and their data is encrypted as follows. First of all, DO constructs the tag key K from a cryptographic hash function ($\mathcal{H} : \{0,1\}^* \rightarrow \{0,1\}^\kappa$), $K = \mathcal{H}(ID)$. Then, DO encrypts the data with a secure symmetric encryption algorithm, $d' = Enc_K(ID||data)$. The database $\mathcal{DB}$ is created by the a variety number of blocks $\mathcal{DB} = \{d_1, \ldots, d_n\}$ where n is the number of tags in the system and each d_i consists of the encrypted tag data and an empty Bloom filter, $d_i = \{d'_i, BF_i\}$.

8.5.3.1 Building Index

Our indexes use Bloom filter, which was first proposed in 1970 by Burton H. Bloom [143]. Bloom filter is a memory efficient and probabilistic data structure that represents a set of keywords. The Bloom filter is used for testing whether an element is a member of a given set or not.

Let us explain how a Bloom filter (BF) is created. Let $BF = (b_1, \ldots, b_m)$ be an array of bits with length of m. Initially, all bits are set to zeroes. Assume that k number of independent hash functions $(h_1, \ldots, h_k)$ are used for the Bloom filter. Each hash function takes κ bits and output the value between 1 and m. In order to add a tag ID into BF, k number of index values $(r_1 = h_1(ID), \ldots, r_k = h_k(ID))$ are derived from the hash functions. For each value of $i = 1, \ldots, k$, the bit with the offset r_i on BF is set to one.

Now, in order to check whether a tag ID is membership of the BF, k index values $(r_1 = h_1(ID), \ldots, r_k = h_k(ID))$ are computed by the same process above. Then, check for each value of i whether the bit with the offset

r_i on BF is zero. If at least one of the bits is zero then this implies that the tag ID is not stored in the set. Note that when the element is already in the set, the Bloom gives always the correct answer. However, it may give some false answer when the element is actually not in the Bloom. The size of the Bloom filter and the number of elements, which are added into Bloom filter affects the probability of false positives. For a given false positive probability, these parameters can be optimized. For the computing the probability and optimizing parameters we refer to [143, 144].

In our proposal, let p be defined as the false positive probability of a Bloom and ℓ be the number of elements that are added into the Bloom. Then, the size of Bloom $m$and the number of hashes k used for Bloom indexes are optimized according to the value of p and ℓ. For each tag in the database, DO first creates an empty Bloom filter and computes the encryption of the tag data by $d = Enc_K(ID||data)$ where $K = \mathcal{H}(ID)$. Then, DO computes the set $W = (\mathcal{H}(ID, 1), \ldots, \mathcal{H}(ID, \ell))$ adds each element in the set W into the Bloom filter using the approach mentioned above. Note that the tag ID values are kept secret against CLD.

8.5.3.2 Query Generation

As soon as a reader, $\mathcal{R}$, authenticates a tag and $\mathcal{R}$ gets tag's ID. $\mathcal{R}$ then generates a query by computing k index values of $ID_1 = \mathcal{H}(ID, 1)$ ($r_1 = h_1(ID_1), \ldots, r_k = h_k(ID_1)$) in order to retrieve the tag data from the cloud. For each value of $i = 1, \ldots, k$, the bit with the offset r_i on the Bloom is set to one. The result bit-string of the Bloom is assigned to a query. It is clearly seen that this query is deterministic and can be distinguished from the other queries and so this construction violates tag owner privacy. Therefore, in order to hide search pattern against the cloud, we use a modified version of a

randomization method similar to [145]. The randomization method of [145] does not really provide query privacy because the authors stated that the privacy is violated when the number of genuine keyword is known by the adversary. However, all the keywords used in our query construction are meaningful.

Before constructing a query, $\mathcal{R}$ randomly selects s number of elements ($w \in_R \{1, \ldots, \ell\}$) and adds them into a set W. Note that the number of selection can be done in the number of $\binom{\ell}{s}$ ways and when ℓ increases the success probability of the distinguishing two queries would decrease. Then, for each element $w \in W$, $ID_w = \mathcal{H}(ID, w)$ is computed and ID_w is added into the Bloom filter. The final output of the Bloom is assigned into a result query $q(.) = BF_i$.

We highlight that when the ID values are known by the cloud, the cloud can distinguish with high probability whether the queries (Bloom filters) are generated from the same search term or not correctly. However, when the ID values are kept secret against the cloud, the success probability is negligible.

8.5.3.3 Search

Upon receiving of a search query $q(w)$, CLD scans $\mathcal{DB}$ for each $d_i \in \mathcal{DB}$ if BF_i contains all elements of $q(w)$, then the encrypted tag data d'_i is added into the result set. Let p be the false positive probability and n be the number of the tags in the database. Then, the expected size of the result set would be np. The success probability of distinguishing two queries based on the result set is $1/np$. When n or p increases, the size of result set increases and the probability increases.

8.5.3.4 Tag Data Retrieval

Upon receiving of the result set, *Result*, $\mathcal{R}$ computes the encryption key $K = \mathcal{H}(ID)$ where ID is the tag identifier. Then, it scans each tag data $d'_j \in Result$ as follow. $\mathcal{R}$ first decrypts $Dec_K(d_i)$ and then checks whether the decryption starts with ID. If a match is found, it returns the tag data, otherwise it continues with the remaining candidate tag data in the set.

8.5.4 Security Analysis

Theorem 29. *The proposed protocol satisfies data privacy with non-negligible probability according to Definition 29.*

Proof. Note that, the theorem states that whether or not the adversary learns the content of the data. Thus, it is independent of the query applied. Let the adversary pre-calculate the hash values of some IDs which is polynomially bounded in the security parameter κ and all the data in the database is provided to the adversary, i.e. n data. The adversary does the following: for each $\mathcal{H}(ID)$ that she has calculated, decrypts every data and compares whether or not it starts with corresponding value of ID. Therefore in the worst case (all data are encrypted with different keys and all precomputed hash values are different), the probability that the adversary retrieves any data value is less than

$$\sum_{i=0}^{poly(\kappa)-1} \frac{k \times n}{2^\kappa - i} < \frac{k \times n \times poly(\kappa)}{2^\kappa - poly(\kappa)},$$

which is negligible as polynomial growth cannot compensate exponential one. Therefore, the proposed protocol satisfies the data privacy with non-negligible probability. □

Theorem 30. *The proposed protocol satisfies query privacy with non-negligible probability according to Definition 30.*

Proof. Let a query for ID_{i_0} is given to the adversary where $i_0 \in \{1, \ldots, n\}$. Then, let us provide another query which is for ID_{j_0} where $j_0 \in \{1, \ldots, n\}$. Let us calculate the expected number of t's such that $BF_i[t] = BF_j[t] = 1$ for two different cases $i_0 = j_0$ and $i_0 \neq j_0$. Note that

Case 1: $(i = j)$ First of all, we have to calculate the expected number of common random word is used in both query. The formula below gives the desired result

$$c_0 = \sum_{i=0}^{s} i \frac{\binom{s}{i}\binom{\ell-s}{s-i}}{\binom{\ell}{s}}.$$

Thus, it is guaranteed that these arrays have $c_0 k$ common 1's. For the remaining slots, the expected number of common 1's can be calculated the formula given below

$$c_1 = \sum_{i=0}^{(s-c_0)k} i \frac{\binom{(s-c_0)k}{i}\binom{m-sk}{(s-c_0)k-i}}{\binom{m-c_0 k}{(s-c_0)k}}.$$

Therefore, expected number of common 1's if $i = j$ is $c_0 k + c_1$.

Case 2: $(i \neq j)$ For this case the expected number of t's such that $BF_i[t] = BF_j[t] = 1$ can be calculated the formulate given below

$$c_2 = \sum_{i=0}^{sk} i \frac{\binom{sk}{i}\binom{m-sk}{sk-i}}{\binom{m}{sk}}.$$

Therefore, the difference between the expected number of 1's between two cases is $c = c_0 k + c_1 - c_2$.

Note that, the size of Bloom filter has almost no effect on the value of c for

small c_0 values. The decrease or increase in the value of Bloom size decreases or increases the value of c_0+c_1 and c_2 almost in the same amount. Therefore, we have to focus on decreasing the value of c_0. This can be achieved in two ways. We either decrease values of s or k or increase the value of ℓ. Note that for any given α, one can choose the values of ℓ, s and k such that $c < \alpha$. Thus, thanks to suitable parameter selection, the proposed protocol satisfies query privacy with non-negligible probability. □

Corollary 4. *The proposed protocol satisfies result pattern privacy with non-negligible probability according to Definition 31*

Proof. The previous theorem states that from given a pair of queries the adversary can distinguish whether these two queries are applied for the same tag or not with negligible probability. Additionally, if the result sets of the queries are given to the adversary, the expected number of tag data in a result set is np per query. By Theorem 8, it is known that from the results, the adversary has only negligible advantage over tag information. Therefore, the adversary can break the result pattern privacy with the sum of $1/np$ and the probabilities of the previous two theorems. □

8.5.5 Practical Setups for Single-Keyword Search

Let the security level of the query privacy be at least 80-bit. Then, let the number of tags n in the system be 2^{20} and let the probability of false positive of a Bloom filter be 2^{-10}. With these assumptions, we optimize that the Bloom size $m = 2^{14}$ bits (2KB), the number of hashes used in the filter $k = 6$. The number of elements that will be inserted into a Bloom filter is $\ell = 1024$ and the number of elements that would be added into the Bloom of a query is $s = 11$. The query can be represented by at most $s \times k$ index

values and each index can be represented by $log_2(m) = 14$ bits, hence the query size will be at most $11 \times 6 \times 14$ ($\approx$ 116Byte) instead of 2KB.

With ℓ and s, there will be $\binom{1024}{11} \approx 2^{84}$ number of variants. Given two queries based on either the same ID value or different ID value, the expected the number of common bits on their Bloom for the both cases is less than 0.8. Besides, given a query, distinguishing the correct result from the result set is $1/(2^{20} \times 2^{-10}) = 2^{-10}$.

8.6 The Summary of the Chapter

In this chapter, we provide a new security and privacy model for RFID authentication systems, which use cloud services to leverage the availability and scalability. In this context, we first define the capabilities of the adversary and give the definitions related to security and privacy. After that, we present two RFID authentication protocols and provide the security analysis using our new privacy model. We prove that the first proposal based on symmetric cryptography and PUFs achieves destructive privacy whereas the second proposal based on ECC is narrow-strong private. Finally, we introduce an efficient private information retrieval between reader and the cloud. This search mechanism is proven that it satisfies data, query and result pattern privacy.

Chapter 9

CONCLUSIONS

The rapid increase of deployment of RFID technology into our daily-life has created new security and privacy challenges. The tags/labels, which are the main components of the RFID systems, are ubiquitous elements and they can easily be abused. In order to protect the security and privacy of the tag owners, several cryptographic algorithms and protocols have been proposed in the literature. However, not all cryptographic solutions can be applied to the tags because of their chip area, time, power and energy constraints. Therefore, lightweight cryptographic solutions are getting important to handle such issues. Nevertheless, developing such lightweight cryptographic protocols is also a very challenging task, because these tags are susceptible to tampering. Namely, the secrets used for authentication can be extracted from the tags. On the other hand, while designing a privacy-preserving RFID authentication protocol, the security and privacy goals should be should be proven using a formal privacy model.

Motivated by these challenges, this dissertation addressed the security and privacy issues in RFID systems from the cryptography and information security points of view. The contributions of the thesis are summarized as

follows.

In Chapter 3, Vaudenay's privacy model is revisited and a privacy deficiency is addressed. In order to cover this deficiency, we introduced two new notions of adversary classes, k-strong adversary and k-forward adversary. With these adversary classes, two new privacy classes are derived such as k-strong privacy and k-forward privacy. Moreover, we study existing PUF definitions and the assumption behind these definitions. In order to make PUFs more robust and secure, we proposed a new extended PUF definition k-PUFs, in which the PUFs are not destroyed up to k corruptions. These type of PUFs are more realistic than prior proposals. To demonstrate our model, we give two PUF based authentication protocols. The first one achieves the strong privacy in Vaudenay's model (∞-strong privacy in our model). The second protocol satisfies both strong privacy and reader authentication.

In Chapter 4, we study security and privacy of offline RFID systems and introduce the notion of compromised reader attacks. In this context, we present the notion of *privacy+*. In order to demonstrate our privacy model, we propose an RFID mutual authentication protocol based on PUF functions. We proved that the protocol satisfies destructive privacy for tag owner. This is the first proposal in the literature that provides destructive privacy in case of compromised reader for offline model.

In Chapter 5, we first give a formal security and privacy analysis of a recently published RFID authentication protocol [31]. Then, we introduced a unilateral authentication protocol and proved that this protocol achieves narrow strong privacy. We also proposed the enhanced version of our protocol that satisfies both destructive privacy and reader authentication.

In Chapter 6, we studied RFID distance bounding protocols, briefly reviewed current challenge dependent protocols and introduced the notion of

k-PCD protocols. We proved that when we increase the dependency parameter k, security level against mafia fraud attack and distance fraud attack increases, as expected. We also proved the conjecture that the best trade-off curve for k_1-PCD protocols lies above the best trade-off curve for k_2-PCD protocols where $k_1 < k_2$. Finally, we gave a method of constructing secure k-PCD protocols with only two registers.

In Chapter 7, we proposed two distance bounding protocols based on PUF function for the first time in the literature. We showed that our first protocol achieves the security level of $(3/4)^n$ against mafia and distance frauds and $(3/4)^n$ against terrorist fraud under the assumption that the tag is still functional after an attack. The second protocol is the extension of the first one by adding a final signature to enhance the security levels. We proved that second protocol achieves the security level $(1/2)^n$ against for all mafia, terrorist and distance frauds. To the best our knowledge, this is the first protocol in the literature that achieves the ideal security level $(1/2)^n$ against all frauds.

In Chapter 8, we provided a new security and privacy model for RFID authentication systems that use cloud services to leverage the availability and scalability. To do so, we first introduced the capabilities of the adversary and gave the definitions related to security and privacy. Then, two RFID authentication protocols were proposed. The first proposal is based on symmetric cryptography and PUF functions, and it achieves destructive privacy. The second proposal is based on ECC and satisfies narrow-strong privacy. Finally, we introduced an efficient private information retrieval mechanism between reader and the cloud. This mechanism satisfies data, query and result pattern privacy.

Bibliography

[1] S. Kardaş, M. S. Kiraz, M. A. Bingöl, and H. Demirci, "A Novel RFID Distance Bounding Protocol Based on Physically Unclonable Functions," in *Workshop on RFID Security – RFIDSec'11* (A. Juels and C. Paar, eds.), vol. 7055 of *Lecture Notes in Computer Science*, (Amherst, Massachusetts, USA), pp. 78–93, Springer Berlin Heidelberg, June 2012.

[2] G. Avoine and P. Oechslin, "A scalable and provably secure hash-based RFID protocol," in *Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops*, PERCOMW '05, (Washington, DC, USA), pp. 110–114, IEEE Computer Society, 2005.

[3] A.-R. Sadeghi, I. Visconti, and C. Wachsmann, "PUF-Enhanced RFID Security and Privacy," in *Secure Component and System Identification – SECSI'10*, (Cologne, Germany), April 2010.

[4] D. Henrici and P. Müller, "Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers," in *Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, PERCOMW '04, (Washington, DC, USA), pp. 149–, IEEE Computer Society, 2004.

[5] C. H. Lim and T. Kwon, "Strong and robust RFID authentication enabling perfect ownership transfer," in *Conference on Information and Communications Security ICICS'06*, pp. 1–20, Springer, 2006.

[6] B. Song and C. J. Mitchell, "RFID authentication protocol for low-cost tags," in *Proceedings of the first ACM conference on Wireless network security*, WiSec '08, (New York, NY, USA), pp. 140–147, ACM, 2008.

[7] D. Molnar and D. Wagner, "Privacy and security in library RFID: issues, practices, and architectures," in *Proceedings of the 11th ACM conference on Computer and communications security*, CCS '04, (New York, NY, USA), pp. 210–219, ACM, 2004.

[8] D. Molnar and D. Wagner, "Privacy and security in library RFID: issues, practices, and architectures," in *CCS '04: Proceedings of the 11th ACM conference on Computer and communications security*, pp. 210–219, ACM, 2004.

[9] I. Erguler and E. Anarim, "Practical attacks and improvements to an efficient radio frequency identification authentication protocol," *Concurrency and Computation: Practice and Experience*, October 2011.

[10] M. Ohkubo, K. Suzuki, and S. Kinoshita, "Cryptographic Approach to "Privacy-Friendly" Tags," in *RFID Privacy Workshop*, (MIT, Massachusetts, USA), November 2003.

[11] B. Alomair, A. Clark, J. Cuellar, and R. Poovendran, "Scalable rfid systems: a privacy-preserving protocol with constant-time identification," *Dependable Systems and Networks, International Conference on*, vol. 0, pp. 1–10, 2010.

[12] S. Kardaş, A. Levi, and E. Murat, "Providing Resistance against Server Information Leakage in RFID Systems," in *New Technologies, Mobility and Security – NTMS'11*, (Paris, France), pp. 1–7, IEEE, IEEE Computer Society, February 2011.

[13] M. Burmester, B. de Medeiros, and R. Motta, "Anonymous rfid authentication supporting constant-cost key-lookup against active adversaries," *IJACT*, vol. 1, no. 2, pp. 79–90, 2008.

[14] J. Ha, S.-J. Moon, J. M. G. Nieto, and C. Boyd, "Low-cost and strong-security rfid authentication protocol," in *EUC Workshops*, pp. 795–807, 2007.

[15] B. Song and C. J. Mitchell, "Scalable RFID Security Protocols supporting Tag Ownership Transfer," *Computer Communication, Elsevier*, March 2010.

[16] A. Fernandez-Mir, R. Trujillo-Rasua, and J. Castella-Roca, "Scalable RFID Authentication Protocol Supporting Ownership Transfer and Controlled Delegation," in *Workshop on RFID Security – RFIDSec'11*, (Amherst, Massachusetts, USA), June 2011.

[17] T. Dimitriou, "A Lightweight RFID Protocol to protect against Traceability and Cloning attacks," in *SECURECOMM '05: Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks*, (Washington, DC, USA), pp. 59–66, IEEE Computer Society, 2005.

[18] S. Vaudenay, "On privacy models for RFID," in *Proceedings of the Advances in Crypotology 13th international conference on Theory and*

application of cryptology and information security, ASIACRYPT'07, (Berlin, Heidelberg), pp. 68–87, Springer, 2007.

[19] G. Avoine, "Adversary Model for Radio Frequency Identification," Technical Report LASEC-REPORT-2005-001, Swiss Federal Institute of Technology (EPFL), Security and Cryptography Laboratory (LASEC), Lausanne, Switzerland, September 2005.

[20] J. Lai, R. H. Deng, and Y. Li, "Revisiting unpredictability-based rfid privacy models," in *Proceedings of the 8th international conference on Applied cryptography and network security*, ACNS'10, (Berlin, Heidelberg), pp. 475–492, Springer, 2010.

[21] A. Juels and S. A. Weis, "Defining strong privacy for rfid," *ACM Trans. Inf. Syst. Secur.*, vol. 13, pp. 7:1–7:23, November 2009.

[22] J. Hermans, A. Pashalidis, F. Vercauteren, and B. Preneel, "A new rfid privacy model," in *Proceedings of the 16th European conference on Research in computer security*, ESORICS'11, (Berlin, Heidelberg), pp. 568–587, Springer, 2011.

[23] J. Ha, S. Moon, J. Zhou, and J. Ha, "A new formal proof model for rfid location privacy," in *Proceedings of the 13th European Symposium on Research in Computer Security: Computer Security*, ESORICS '08, (Berlin, Heidelberg), pp. 267–281, Springer, 2008.

[24] T. Van Deursen, S. Mauw, and S. Radomirović, "Untraceability of rfid protocols," in *Proceedings of the 2nd IFIP WG 11.2 international conference on Information security theory and practices: smart devices, convergence and next generation networks*, WISTP'08, (Berlin, Heidelberg), pp. 1–15, Springer, 2008.

[25] R. H. Deng, Y. Li, M. Yung, and Y. Zhao, "A new framework for rfid privacy," in *Proceedings of the 15th European conference on Research in computer security*, ESORICS'10, (Berlin, Heidelberg), pp. 1–18, Springer, 2010.

[26] S. Canard, I. Coisel, J. Etrog, and M. Girault, "Privacy-preserving rfid systems: Model and constructions," 2010.

[27] M. Burmester, T. van Le, and B. de Medeiros, "Provably Secure Ubiquitous Systems: Universally Composable RFID Authentication Protocols," in *Securecomm and Workshops, 2006*, pp. 1 –9, 28 2006-sept. 1 2006.

[28] I. Coisel and T. Martin, "Untangling RFID privacy models," *Journal of Computer Networks and Communications*, 2013.

[29] S. Kardaş, Şerkan Celik, M. A. Bingol, M. S. Kiraz, H. Demirci, and A. Levi, "k-Strong Privacy for RFID Authentication Protocols Based on Physically Unclonable Functions," *Wireless Communications and Mobile Computing(Accepted)*, pp. 1–17, 2014.

[30] S. Kardaş, S. Çelik, M. Yildiz, and A. Levi, "PUF-enhanced offline RFID security and privacy," *Journal of Network and Computer Applications*, vol. 11, no. 12, pp. 1–11, 2012.

[31] T.-C. Yeh, C.-H. Wu, and Y.-M. Tseng, "Improvement of the rfid authentication scheme based on quadratic residues," *Computer Communications*, vol. 34, no. 3, pp. 337 – 341, 2011.

[32] S. Kardas, S. Celik, M. Sariyuce, and A. Levi, "An efficient and private authentication protocol for rfid systems," *Journal of Communications Software & Systems*, vol. 9, no. 2, pp. 128–136, 2013.

[33] S. Kardas, S. Celik, M. Sariyuce, and A. Levi, "A secure and private rfid authentication protocol based on quadratic residue," in *Software, Telecommunications and Computer Networks (SoftCOM), 2012 20th International Conference on*, pp. 1–6, 2012.

[34] O. Kara, S. Kardaş, M. A. Bingöl, and G. Avoine, "Optimal Security Limits of RFID Distance Bounding Protocols," in *Workshop on RFID Security – RFIDSec'10* (S. O. Yalcin, ed.), vol. 6370 of *Lecture Notes in Computer Science*, (Istanbul, Turkey), pp. 220–238, Springer, June 2010.

[35] S. Kardaş, M. A. Bingol, O. Ersoy, and A. Levi, "Optimum Security Limits of k-PCD RFID Distance Bounding Protocols," *a Journal (Submitted)*, 2014.

[36] S. Kardas, S. Çelik, M. A. Bingöl, and A. Levi, "A new security and privacy framework for rfid in cloud computing," in *Proceedings of the IEEE International Conference on Cloud Computing Technology and Science*, Cloudcom2013, pp. 171–176, 2013.

[37] S. Kardaş, S. Çelik, M. A. Bingol, and A. Levi, "ARCs: Anonymous RFID authentication based on Cloud Services," *Journal (Submitted)*, 2014.

[38] P. Lopez, *Lightweight Cryptography in Radio Frequency Identification (RFID) Systems*. PhD thesis, Computer Science Department, Carlos III University of Madrid, 2008.

[39] G. Avoine, C. Lauradoux, and T. Martin, "When Compromised Readers Meet RFID," in *Workshop on Information Security Applications –*

WISA'09 (H. Youm and M. Yung, eds.), vol. 5932 of *Lecture Notes in Computer Science*, (Busan, Korea), pp. 36–50, Springer, August 2009.

[40] RFIDea, "Engineering & Applications in Electronic Traceability." http://www.rfidea.com, 2012.

[41] G. Avoine, I. Coisel, and T. Martin, "A privacy-restoring mechanism for offline rfid systems," in *Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks*, WISEC '12, (New York, NY, USA), pp. 63–74, ACM, 2012.

[42] Boycott Benetton Web Site, December 2005. `http://www.boycottbenetton.com`.

[43] S. Ahson and M. Ilyas, *RFID Handbook: Applications, Technology, Security, and Privacy*. CRC Press, 2008.

[44] Y. Desmedt, C. Goutier, and S. Bengio, "Special Uses and Abuses of the Fiat-Shamir Passport Protocol," in *Advances in Cryptology – CRYPTO'87* (C. Pomerance, ed.), vol. 293 of *LNCS*, (Santa Barbara, California, USA), pp. 21–39, IACR, Springer, August 1988.

[45] T. Beth and Y. Desmedt, "Identification tokens - or: Solving the chess grandmaster problem," in *Advances in Cryptology - CRYPTO '90, 10th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1990*, vol. 537 of *Lecture Notes in Computer Science*, pp. 169–177, Springer, 1990.

[46] J. H. Conway, *On Numbers and Games*. No. 6 in London Mathematical Society Monographs, Academic Press, London-New-San Francisco, 1976.

[47] G. P. Hancke, "A Practical Relay Attack on ISO 14443 Proximity Cards." Manuscript, February 2005.

[48] G. P. Hancke, "Practical Attacks on Proximity Identification Systems (Short Paper)," in *IEEE Symposium on Security and Privacy – S&P '06*, (Oakland, California, USA), IEEE, IEEE Computer Society, May 2006.

[49] G. Hancke, K. Mayes, and K. Markantonakis, "Confidence in Smart Token Proximity: Relay Attacks Revisited," in *Elsevier Computers & Security*, June 2009.

[50] M. Hlaváč and T. Rosa, "A Note on the Relay Attacks on e-Passports: The Case of Czech e-Passports." Cryptology ePrint Archive, Report 2007/244, 2007.

[51] K. Markantonakis, M. Tunstall, G. Hancke, I. Askoxylakis, and K. Mayes, "Attacking smart card systems: Theory and practice," *Information Security Technical Report*, vol. 14, no. 2, pp. 46 – 56, 2009. Smart Card Applications and Security.

[52] G. Avoine, M. A. Bingöl, S. Kardaş, C. Lauradoux, and B. Martin, "A Framework for Analyzing RFID Distance Bounding Protocols," *Journal of Computer Security – Special Issue on RFID System Security*, vol. 19, pp. 289–317, March 2011.

[53] L. Sportiello and A. Ciardulli, "Long distance relay attack," in *Workshop on RFID Security – RFIDSec'13*, (Graz, Austria), July 2013.

[54] D. C. Ranasinghe and P. H. Cole, "Addressing Insecurities and Violations of Privacy ," in *Networked RFID Systems and Lightweight Cryptography*, pp. 101–145, Springer Berlin Heidelberg, 2008.

[55] D. R. Brown, "Generic groups, collision resistance, and ECDSA," *Des. Codes Cryptography*, vol. 35, pp. 119–152, Apr. 2005.

[56] D. W. Bauder., "An Anti-Counterfeiting Concept for Currency Systems." Research report PTK- 11990. Sandia National Labs, 1983.

[57] G. J. Simmons, "A system for verifying user identity and authorization at the point-of sale or access," *Cryptologia*, vol. 8, no. 1, pp. 1–21, 1984.

[58] G. Simmons, "Identification of data, devices, documents and individuals," in *Security Technology, 1991. Proceedings. 25th Annual 1991 IEEE International Carnahan Conference on*, pp. 197 –218, oct 1991.

[59] D. Naccache and P. Fremanteau, "Unforgeable identification device, identification device reader and method of identification." Patent-EP0583709, 1994.

[60] P. Ravikanth, *Physical One-Way Functions*. PhD thesis, Massachusetts Institute of Technology, March 2001.

[61] R. S. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, pp. 2026–2030, 2002.

[62] B. Gassend, D. Clarke, M. van Dijk, and S. Devadas, "Silicon physical random functions," in *Proceedings of the 9th ACM conference on Computer and communications security*, CCS '02, (New York, NY, USA), pp. 148–160, ACM, 2002.

[63] S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, "Extended abstract: The butterfly puf protecting ip on every fpga," in *Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on*, pp. 67–70, June 2008.

[64] R. Maes, P. Tuyls, and I. Verbauwhede, "Intrinsic pufs from flip-flops on reconfigurable devices," in *3rd Benelux Workshop on Information and System Security (WISSec 2008)*, (Eindhoven,NL), p. 17, 2008.

[65] D. E. Holcomb, W. P. Burleson, and K. Fu, "Initial SRAM state as a fingerprint and source of true random numbers for RFID tags," in *In Proceedings of the Conference on RFID Security*, 2007.

[66] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM J. Comput.*, vol. 38, no. 1, pp. 97–139, 2008.

[67] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *SIAM J. Comput.*, vol. 38, no. 1, pp. 97–139, 2008.

[68] F. Armknecht, R. Maes, A.-R. Sadeghi, F.-X. Standaert, and C. Wachsmann, "A formalization of the security features of physical functions," in *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, SP '11, (Washington, DC, USA), pp. 397–412, IEEE Computer Society, 2011.

[69] D. C. Ranasinghe, D. W. Engels, and P. H. Cole, "Security and Privacy: Modest Proposals for Low-Cost RFID Systems," in *Systems, Proc. Auto-ID Labs Research Workshop*, 2004.

[70] S. Devadas, E. Suh, S. Paral, R. Sowell, T. Ziola, and V. Khandelwal, "Design and Implementation of PUF-Based 'Unclonable' RFID ICs for Anti-Counterfeiting and Security Applications," in *RFID, 2008 IEEE International Conference on*, pp. 58–64, 2008.

[71] P. Tuyls and L. Batina, "RFID-Tags for Anti-counterfeiting," in *Topics in Cryptology – CT-RSA 2006*, vol. 3860 of *LNCS*, pp. 115–131, 2006.

[72] L. Bolotnyy and G. Robins, "Physically Unclonable Function-Based Security and Privacy in RFID Systems," in *Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications*, (Washington, DC, USA), pp. 211–220, IEEE Computer Society, 2007.

[73] Y. S. Lee, Y. Park, S. Lee, T. Kim, and H. J. Lee, "Rfid mutual authentication protocol with unclonable rfid-tags," in *Mobile IT Convergence (ICMIC), 2011 International Conference on*, pp. 74 –77, 2011.

[74] K. Yang, K. Zheng, Y. Guo, and D. Wei, "Puf-based node mutual authentication scheme for delay tolerant mobile sensor network," in *Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on*, pp. 1 –4, sept. 2011.

[75] M. Akgun and M. Caglayan, "Puf based scalable private rfid authentication," in *Availability, Reliability and Security (ARES), 2011 Sixth International Conference on*, pp. 473 –478, aug. 2011.

[76] Z. Li and G. Gong, "Computationally efficient mutual entity authentication in wireless sensor networks," *Ad Hoc Networks*, vol. 9, no. 2, pp. 204 – 215, 2011.

[77] P. Koeberl, J. Li, A. Rajan, C. Vishik, and W. Wu, "A practical device authentication scheme using sram pufs," in *Trust and Trustworthy Computing*, vol. 6740, pp. 63–77, Springer Berlin / Heidelberg, 2011.

[78] G. Hancke and M. Kuhn, "An RFID Distance Bounding Protocol," in *Conference on Security and Privacy for Emerging Areas in Com-*

munication Networks – SecureComm 2005, (Athens, Greece), IEEE, September 2005.

[79] S. Brands and D. Chaum, "Distance-Bounding Protocols," in *Advances in Cryptology – EUROCRYPT'93*, vol. 765 of *Lecture Notes in Computer Science*, (Lofthus, Norway), pp. 344–359, Springer, May 1993.

[80] S. Capkun, L. Butty'an, and J.-P. Hubaux, "SECTOR: Secure Tracking of Node Encounters in Multi-hop Wireless Networks," in *ACM Workshop on Security of Ad Hoc and Sensor Networks – SASN'03*, (Fairfax, Virginia, USA), pp. 21–32, ACM, October 2003.

[81] G. Avoine, C. Lauradoux, and B. Martin, "How secret-sharing can defeat terrorist fraud," in *Proceedings of the 4th ACM Conference on Wireless Network Security – WiSec'11*, (Hamburg, Germany), ACM, ACM Press, June 2011.

[82] Y.-J. Tu and S. Piramuthu, "RFID Distance Bounding Protocols," in *First International EURASIP Workshop on RFID Technology*, (Vienna, Austria), September 2007.

[83] J. Reid, J. Gonzalez Neito, T. Tang, and B. Senadji, "Detecting relay attacks with timing based protocols," in *Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security – ASIACCS '07* (F. Bao and S. Miller, eds.), (Singapore, Republic of Singapore), pp. 204–213, ACM, March 2007.

[84] J. Munilla and A. Peinado, "Distance bounding protocols for RFID enhanced by using void-challenges and analysis in noisy channels," *Wireless Communications and Mobile Computing*, vol. 8, no. 9, pp. 1227–1232, 2008.

[85] M.-H. Park, K.-G. Nam, J. S. Kim, D. H. Yum, and P. J. Lee, "Unilateral distance bounding protocol with bidirectional challenges," *IEICE Transactions on Information and Systems*, vol. E96-D, pp. 134–137, January 2013.

[86] J. Hermans, R. Peeters, and C. Onete, "Efficient, secure, private distance bounding without key updates," in *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks – WiSec'13*, WiSec '13, (Budapest, Hungary), pp. 207–218, ACM, April 2013.

[87] I. Boureanu, A. Mitrokotsa, and S. Vaudenay, "Secure and Lightweight Distance-Bounding," in *Second International Workshop on Lightweight Cryptography for Security and Privacy – LightSec 2013*, (Gebze, Turkey), May 2013.

[88] A. Yang, Y. Zhuang, and D. S. Wong, "An efficient single-slow-phase mutually authenticated RFID distance bounding protocol with tag privacy," in *International Conference on Information and Communications Security – ICICS'12* (T. Chim and T. Yuen, eds.), vol. 7618 of *Lecture Notes in Computer Science*, (Hong Kong, China), pp. 285–292, Springer Berlin Heidelberg, October 2012.

[89] J. S. Kim, K. Cho, D. H. Yum, S. J. Hong, and P. J. Lee, "Lightweight distance bounding protocol against relay attacks," *IEICE Transactions on Information and Systems*, vol. E95.D, pp. 1155–1158, April 2012.

[90] R. Trujillo Rasua, B. Martin, and G. Avoine, "The Poulidor Distance-Bounding Protocol," in *Workshop on RFID Security – RFIDSec'10*

(S. O. Yalcin, ed.), vol. 6370 of *LNCS*, (Istanbul, Turkey), pp. 239–257, Springer, June 2010.

[91] G. Avoine, C. Floerkemeier, and B. Martin, "RFID Distance Bounding Multistate Enhancement," in *Proceedings of the 10th International Conference on Cryptology in India – Indocrypt 2009*, vol. 5922 of *Lecture Notes in Computer Science*, (New Delhi, India), pp. 290–307, Springer Berlin / Heidelberg, December 2009.

[92] C. H. Kim, G. Avoine, F. Koeune, F.-X. Standaert, and O. Pereira, "The Swiss-Knife RFID Distance Bounding Protocol," in *International Conference on Information Security and Cryptology – ICISC'08*, vol. 5461 of *Lecture Notes in Computer Science*, (Seoul, Korea), pp. 98–115, Springer, December 2008.

[93] V. Nikov and M. Vauclair, "Yet Another Secure Distance-Bounding Protocol." Cryptology ePrint Archive, Report 2008/319, 2008.

[94] D. Singelée and B. Preneel, "Distance Bounding in Noisy Environments," in *European Workshop on Security in Ad-hoc and Sensor Networks – ESAS'07*, vol. 4572 of *Lecture Notes in Computer Science*, (Cambridge, UK), pp. 101–115, Springer, July 2007.

[95] G. Avoine and A. Tchamkerten, "An Efficient Distance Bounding RFID Authentication Protocol: Balancing False-acceptance Rate and Memory Requirement," in *Information Security Conference – ISC'09*, vol. 5735 of *Lecture Notes in Computer Science*, (Pisa, Italy), Springer, September 2009.

[96] G. Kapoor, W. Zhou, and S. Piramuthu, "Distance Bounding Protocol for Multiple RFID Tag Authentication," in *EUC '08: Proceedings of*

the 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, (Shanghai, China), pp. 115–120, IEEE Computer Society, December 2008.

[97] C. H. Kim and G. Avoine, "RFID distance bounding protocol with mixed challenges to prevent relay attacks," in *International Conference on Cryptology and Network Security – CANS*, vol. 5888 of *Lecture Notes in Computer Science*, (Kanazawa, Ishikawa, Japan), pp. 119–133, Springer, December 2009.

[98] R.-I. Paise and S. Vaudenay, "Mutual authentication in rfid: security and privacy," in *Proceedings of the 2008 ACM symposium on Information, computer and communications security*, ASIACCS '08, (New York, NY, USA), pp. 292–299, ACM, 2008.

[99] F. Armknecht, A.-R. Sadeghi, I. Visconti, and C. Wachsmann, "On rfid privacy with mutual authentication and tag corruption," in *Proceedings of the 8th international conference on Applied cryptography and network security*, ACNS'10, (Berlin, Heidelberg), pp. 493–510, Springer, 2010.

[100] M. H. Habibi and M. R. Aref, "Two RFID privacy models in front of a court." Cryptology ePrint Archive, Report 2011/625, 2011.

[101] J. Guajardo, S. Kumar, G.-J. Schrijen, and P. Tuyls, "Physical unclonable functions and public-key crypto for fpga ip protection," in *Field Programmable Logic and Applications, 2007. FPL 2007. International Conference on*, pp. 189–195, 2007.

[102] S. Maubach, T. Kevenaar, and P. Tuyls, "Information-theoretic analysis of coating pufs," 2006.

[103] F. Armknecht, R. Maes, A.-R. Sadeghi, B. Sunar, and P. Tuyls, "Memory leakage-resilient encryption based on physically unclonable functions," in *Advances in Cryptology ASIACRYPT 2009*, vol. 5912 of *Lecture Notes in Computer Science*, pp. 685–702, Springer Berlin Heidelberg, 2009.

[104] P. Tuyls, G.-J. Schrijen, B. Škorić, J. van Geloven, N. Verhaegh, and R. Wolters, "Read-proof hardware from protective coatings," in *Proceedings of the 8th international conference on Cryptographic Hardware and Embedded Systems*, CHES'06, (Berlin, Heidelberg), pp. 369–383, Springer, 2006.

[105] P. Tuyls and B. Skoric, "Secret key generation from classical physics: Physical uncloneable functions," in *AmIware Hardware Technology Drivers of Ambient Intelligence* (S. Mukherjee, R. Aarts, R. Roovers, F. Widdershoven, and M. Ouwerkerk, eds.), vol. 5 of *Philips Research*, pp. 421–447, Springer Netherlands, 2006.

[106] C. C. Tan, B. Sheng, and Q. Li, "Secure and Serverless RFID Authentication and Search Protocols," *IEEE Transactions on Wireless Communications*, vol. 7, no. 4, pp. 1400–1407, 2008.

[107] Y. Chen, J.-S. Chou, and H.-M. Sun, "A novel mutual authentication scheme based on quadratic residues for rfid systems," *Computer Networks*, vol. 52, no. 12, pp. 2373 – 2380, 2008.

[108] S. Older and S.-K. Chin, "Formal methods for assuring security of protocols.," *Comput. J.*, vol. 45, no. 1, pp. 46–54, 2002.

[109] B. Blanchet, "An efficient cryptographic protocol verifier based on prolog rules," in *Proceedings of the 14th IEEE workshop on Computer*

Security Foundations, CSFW '01, (Washington, DC, USA), pp. 82–, IEEE Computer Society, 2001.

[110] A. Armando, D. Basin, Y. Boichut, Y. Chevalier, L. Compagna, J. Cuellar, P. H. Drielsma, P. C. Heám, O. Kouchnarenko, J. Mantovani, S. Mödersheim, D. von Oheimb, M. Rusinowitch, J. Santiago, M. Turuani, L. Viganò, and L. Vigneron, "The avispa tool for the automated validation of internet security protocols and applications," in *Proceedings of the 17th international conference on Computer Aided Verification*, CAV'05, (Berlin, Heidelberg), pp. 281–285, Springer, 2005.

[111] P. Ryan and S. Schneider, *The modelling and analysis of security protocols: the csp approach*. Addison-Wesley Professional, first ed., 2000.

[112] M. Abadi and C. Fournet, "Mobile values, new names, and secure communication," *SIGPLAN Not.*, vol. 36, pp. 104–115, Jan. 2001.

[113] B. Blanchet and B. Smyth, "Proverif 1.86pl3: Automatic cryptographic protocol verifier, user manual and tutorial." http://www.proverif.ens.fr/manual.pdf, 2012.

[114] R. Küsters and T. Truderung, "Using proverif to analyze protocols with diffie-hellman exponentiation," in *Proceedings of the 2009 22nd IEEE Computer Security Foundations Symposium*, CSF '09, (Washington, DC, USA), pp. 157–171, IEEE Computer Society, 2009.

[115] J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum, and E. W. Felten, "Lest we remember: cold-boot attacks on encryption keys," *Commun. ACM*, vol. 52, pp. 91–98, May 2009.

[116] D. Dolev and A. C.-C. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–207, 1983.

[117] G. Avoine, E. Dysli, and P. Oechslin, "Reducing time complexity in rfid systems," in *Proceedings of the 12th international conference on Selected Areas in Cryptography*, SAC'05, (Berlin, Heidelberg), pp. 291–306, Springer, 2006.

[118] G. Shen and B. Liu, "Research on embedding ecc into rfid authentication protocol," *IEEE TrustCom/IEEE ICESS/FCST, International Joint Conference of*, vol. 0, pp. 1835–1838, 2012.

[119] M. A. Bingöl, F. Birinci, S. Kardaş, and M. S. Kiraz, "Anonymous RFID Authentication for Cloud Services," *International Journal of Information Security Science*, vol. 1, pp. 32–42, June 2012.

[120] G. Avoine, "Privacy issues in rfid banknote protection schemes," in *Smart Card Research and Advanced Applications CARDIS*, pp. 33–48, IFIP, Kluwer Academic Publishers, 2004.

[121] S. Vaudenay, "RFID privacy based on public-key cryptography," in *ICISC 2006. LNCS*, pp. 1–6, Springer, 2006.

[122] S. V. Kaya, E. Savas, A. Levi, and O. Ercetin, "Public key cryptography based privacy preserving multi-context rfid infrastructure," *Ad Hoc Networks*, vol. 7, no. 1, pp. 136 – 152, 2009.

[123] Y. Oren and M. Feldhofer, "A low-resource public-key identification scheme for RFID tags and sensor nodes," in *Proceedings of the second ACM conference on Wireless network security*, WiSec '09, (New York, NY, USA), pp. 59–68, ACM, 2009.

[124] H.-Y. Chien, "Sasi: A new ultralightweight rfid authentication protocol providing strong authentication and strong integrity," *Dependable and Secure Computing, IEEE Transactions on*, vol. 4, pp. 337 –340, oct.-dec. 2007.

[125] B. Danev, T. S. Heydt-Benjamin, and S. Čapkun, "Physical-layer identification of RFID devices," in *Proceedings of the 18th conference on USENIX security symposium*, SSYM'09, (Berkeley, CA, USA), pp. 199–214, USENIX Association, 2009.

[126] S. Canard, I. Coisel, J. Etrog, and M. Girault, "Privacy-Preserving RFID Systems: Model and Constructions." Cryptology ePrint Archive, Report 2010/405, 2010.

[127] Y. K. Lee, L. Batina, D. Singelée, and I. Verbauwhede, "Low-cost untraceable authentication protocols for rfid," in *Proceedings of the third ACM conference on Wireless network security*, WiSec '10, (New York, NY, USA), pp. 55–64, ACM, 2010.

[128] E. Wenger and M. Hutter, "A hardware processor supporting elliptic curve cryptography for less than 9 kges," in *Proceedings of the 10th IFIP WG 8.8/11.2 international conference on Smart Card Research and Advanced Applications*, CARDIS'11, (Berlin, Heidelberg), pp. 182–198, Springer, 2011.

[129] Y. Choi, M. Kim, T. Kim, and H. Kim, "Low power implementation of sha-1 algorithm for rfid system," in *Consumer Electronics, 2006. ISCE '06. 2006 IEEE Tenth International Symposium on*, pp. 1 –5, 0-0 2006.

[130] B. Chor, E. Kushilevitz, O. Goldreich, and M. Sudan, "Private information retrieval," *J. ACM*, vol. 45, pp. 965–981, Nov. 1998.

[131] A. Ambainis, "Upper bound on communication complexity of private information retrieval," in *Proceedings of the 24th International Colloquium on Automata, Languages and Programming*, ICALP '97, (London, UK, UK), pp. 401–407, Springer, 1997.

[132] Y. Ishai and E. Kushilevitz, "Improved upper bounds on information-theoretic private information retrieval (extended abstract)," in *Proceedings of the thirty-first annual ACM symposium on Theory of computing*, STOC '99, (New York, NY, USA), pp. 79–88, ACM, 1999.

[133] T. Itoh, "Efficient private information retrieval," *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E82-A, pp. 11–20, 1999.

[134] J.-F. Raymond, "Private information retrieval: Improved upper bound, extension and applications," tech. rep., McGill University, 2000.

[135] A. Yamamura and T. Saito, "Private information retrieval based on the subgroup membership problem," in *Proceedings of the 6th Australasian Conference on Information Security and Privacy*, ACISP '01, (London, UK, UK), pp. 206–220, Springer, 2001.

[136] B. Chor and N. Gilboa, "Computationally private information retrieval (extended abstract)," in *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, STOC '97, (New York, NY, USA), pp. 304–313, ACM, 1997.

[137] E. Kushilevitz and R. Ostrovsky, "Replication is not needed: Single database, computationally-private information retrieval (extended abstract)," in *IN PROC. OF THE 38TH ANNU. IEEE SYMP. ON FOUNDATIONS OF COMPUTER SCIENCE*, pp. 364–373, 1997.

[138] H. Lipmaa, "An oblivious transfer protocol with log-squared communication," in *Proceedings of the 8th international conference on Information Security*, ISC'05, (Berlin, Heidelberg), pp. 314–328, Springer, 2005.

[139] E. Mann, "Private access to distributed information," in *Master's thesis, Technion - Israel Institute of Technology*, 1998.

[140] J. P. Stern, "A new efficient all-or-nothing disclosure of secrets protocol," in *Proceedings of the International Conference on the Theory and Applications of Cryptology and Information Security: Advances in Cryptology*, ASIACRYPT '98, (London, UK, UK), pp. 357–371, Springer, 1998.

[141] C. Gentry and Z. Ramzan, "Single-database private information retrieval with constant communication rate," in *Automata, Languages and Programming* (L. Caires, G. Italiano, L. Monteiro, C. Palamidessi, and M. Yung, eds.), vol. 3580 of *Lecture Notes in Computer Science*, pp. 803–815, Springer Berlin Heidelberg, 2005.

[142] C. Cachin, S. Micali, and M. Stadler, "Computationally private information retrieval with polylogarithmic communication," in *Proceedings of the 17th international conference on Theory and application of cryptographic techniques*, (Berlin, Heidelberg), pp. 402–414, Springer, 1999.

[143] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Commun. ACM*, vol. 13, pp. 422–426, July 1970.

[144] D. Starobinski, A. Trachtenberg, and S. Agarwal, "Efficient pda synchronization," *IEEE Transactions on Mobile Computing*, vol. 2, pp. 40–51, Jan. 2003.

[145] C. Örencik and E. Savaş, "Efficient and secure ranked multi-keyword search on encrypted cloud data," in *Proceedings of the 2012 Joint EDBT/ICDT Workshops*, EDBT-ICDT '12, (New York, NY, USA), pp. 186–195, ACM, 2012.