SECURITY AND PRIVACY OF RFID PROTOCOLS

by

Mete Akgün

B.S, Computer Engineering, Bahçeşehir University, 2005

M.S, Computer Engineering, Boğaziçi University, 2009

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Computer Engineering

Boğaziçi University

2016

SECURITY AND PRIVACY OF RFID PROTOCOLS

APPROVED BY:

Prof. Mehmet Ufuk Çağlayan (Thesis Supervisor)	.....
Prof. Fatih Alagöz	.....
Prof. Emin Anarım	.....
Prof. Albert Levi	.....
Prof. Semih Bilgen	.....

DATE OF APPROVAL: 11.12.2015

ACKNOWLEDGEMENTS

This thesis is dedicated to my daughter Zeynep.

I would like to thank my thesis advisor Prof. Mehmet Ufuk Çağlayan for his guidance, endless support and contributions throughout the preparation of my thesis. He always supports and encourages me during my studies.

I am deeply grateful to my institution TÜBİTAK-BİLGEM-UEKAE for supporting my research. I would like to express my gratitude to my unit manager Yıldız Uludağ and my project managers Mahmut Şamil Sağıroğlu and Hüseyin Demirci for their support and help during the thesis. Many thanks to Pınar Kavak, Ali Osman Bayrak and Mehmet Yağmur Gök for their support, friendship, and their unique notion of having fun.

Finally, I wish to thank to my family for their endless love, support, patience and encouragement. I especially want to thank my wife Yasemin Akgün for her love, patience and understanding to me.

This research is supported by the Turkish State Planning Organization (DPT) under the TAM Project, number 2007K120610

ABSTRACT

SECURITY AND PRIVACY OF RFID PROTOCOLS

This thesis studies security and privacy issues of Radio Frequency Identification (RFID) technology that enhances ubiquitous computing environment. Privacy is one of main issues to adopt RFID technology in daily use. Due to resource constraints of low cost RFID tags in terms circuit size, power consumption and memory size, it is very restricted to design a private authentication protocol based on existing cryptographic functions.

In this thesis, we focus on the security of low cost RFID tags. Our contributions are as follows. First, we analyze the security of recent RFID authentication protocols with respect to two security requirements: mutual authentication and availability. We propose impersonation and de-synchronization attacks and improvements to recent RFID authentication protocols.

Secondly, we analyze the security of chaotic-map based RFID protocols. We propose secret disclosure, tracking, impersonation and de-synchronization attacks against chaotic-map based RFID protocols. We propose revised protocols resistant to our proposed attacks.

Finally, we study privacy and scalability issues in RFID. All former RFID protocols giving the desired level of privacy required linear work in the back-end server. We propose PUF-based scalable authentication protocols for RFID systems. They provide destructive privacy according to the Vaudenay's privacy and security model. They defend against compromising attack by using PUFs as a secure storage to keep secrets of the tag. To the best of our knowledge, they are the first to provide this level of privacy with constant identification time.

ÖZET

RFID PROTOKOLLERİNİN GÜVENLİĞİ VE GİZLİLİĞİ

Bu tez her yerde birden bulunan hesaplama ortamını geliştiren, radyo frekanslı tanımlamanın (RFID) güvenlik ve gizlilik konuları üzerinde durmaktadır. RFID teknolojisini günlük kullanıma uygun hale getirmedeki en önemli mesele gizliliktir. Düşük maliyetli RFID etiketlerinin devre boyutu, güç tüketimi ve hafıza boyutu açısından kaynak sınırlamaları olduğu için, varolan kriptografik fonksiyonlara dayanarak gizli kimlik denetim protokolleri tasarlamak çok güçtür. Bu nedenle, hafif kriptografiye dayanan yeni gizli kimlik denetim protokolleri gerekmektedir.

Biz bu tezde, düşük maliyetli etiketler üzerinde odaklandık. Bu tez başlıca üç başlık altında katkı şağlar. İlk olarak, RFID protokollerininin karşılıklı kimlik doğrulama ve kullanılabilirlik açısından güvenliğini analiz ediyoruz ve bu protokollere karşı kimliğe bürünme ve uyumsuzluk saldırıları öneriyoruz.

İkinci olarak, kaotik-harita tabanlı RFID protokollerinin güvenliğini analiz ediyoruz. Bu protokollere karşı gizli ahahtar açıklama, izleme, kimliğe bürünme ve senkronizasyon saldırıları öneriyoruz. Önerdiğimiz saldırılara dayanıklı revize edilmiş protokoller öneriyoruz.

Son olarak, RFID gizlilik ve ölçeklenebilirlik sorunlarını inceliyoruz. İstenilen düzeyde gizlilik sağlayan önceki tüm RFID protokolleri arka-uç sunucuda lineer çalışma gerektirir. RFID sistemleri için PUF tabanlı ölçeklenebilir kimlik doğrulama protokolleri öneriyoruz. Önerdiğimiz protokoller Vaudenay'ın gizlilik ve güvenlik modeline göre yıkıcı gizlilik sağlar. Önerdiğimiz protokoller anahtarları PUF kullanarak saklar ve bozma saldırılarına karşı güvenlik sağlar. Bildiğimiz kadarıyla, önerdiğimiz protokoller sabit tanımlama zamanı ile bu seviyede gizlilik sağlayan ilk protokollerdir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$Adv_P^{UNT}(\mathcal{A})$	Advantage of adversary
$\mathcal{A}$	Adversary
$f()$	Pseudorandom number generator
$H()$	Hash function
$\mathcal{G}$	Attack game
$h()$	Hash function
M	Message
N	The number of tags
r	Pseudorandom number generated by PRNG
$\mathcal{R}$	RFID reader
$\mathcal{T}$	RFID tag
T_i	The i-th Tag
$\gg$	Right circular shift operator
$\ll$	Left circular shift operator
$\oplus$	XOR operator
$\|\|$	Concatenation operator
$\in$	The random choice operator

LIST OF ABBREVIATIONS

AES	Advanced Encryption Standard
CR	Challenge Response
CRC	Cyclic Redundancy Code
DB	Database
DoD	Department of Defense
DoS	Denial of Service
EAS	Electronic Article Surveillance
EPC	Electronic Product Code
EPCC1G2	EPCglobal Class 1 Generation 2
HF	Hopper and Blum
HF	High Frequencies
ID	Identifier
IFF	Identify Friend or Foe
LF	Low Frequencies
OSK	Ohkubo Suzuki Kinoshita
PUF	Physically Unclonable Function
PRF	Pseudo Random Function
PRNG	Pseudo Random Number Generator
RA	Restricted Access
RC4	Rivest Cipher 4
RF	Radio Frequency
RFID	Radio Frequency Identification
UA	Universal Access
UHF	Ultra High Frequencies
XOR	Exclusive-or

1. INTRODUCTION

In Radio Frequency Identification (RFID), radio transmissions are used to transmit the identity of an object or person in the form of a unique serial number. RFID is similar to barcodes in terms of functionality. However RFID technology has several advantages when compared with barcodes. In RFID, line-of-sight is not required this means RFID tags must be within the radio range of RFID reader so the contactless identification is possible [2]. RFID tags are resistant to dirt, paint, heat and solvents. Furthermore, RFID tags can collect information and store them [3].

There are three basic components in typical RFID systems: RFID tags, RFID readers and a back-end server. Tags that are used by attaching to objects are the fundamental elements of RFID systems. An RFID tag consists of three elements: a power system, a microchip and an antenna. Passive RFID tags use the RF signal from the reader to obtain the power. The reader or transceiver initiates RFID tags to carry out the necessary procedures by transmitting data. It seats between RFID tags and a back-end server to connect them. It transmits a signal to possible RFID tags. Tags detecting this signal send their identifiers to the reader. The reader sends these identifiers to the back-end server. The back-end server queries its database with these identifiers to obtain elaborated information about tags.

Utilization of RFID technology is growing at a great pace. Business and government have already integrated this technology into their applications such as item and asset tracking, supply chains, logistics, library applications, vehicle payment systems, product identification and passports.

Identification of people or products is the main use of RFID technology. It offers many advantages over previous technologies. The biggest obstacle for applicability of RFID technology in critical areas is that the security problems of this technology is not yet solved. Computation performance of the server is also important issue for scalability of the RFID systems. Another important factor is asymmetric communication channel.

Figure 1.1. A typical RFID system

Signals from reader to tag are more vulnerable to eavesdropping than signals from tag to reader. Therefore the protocol designed for RFID authentication must consider not only security and privacy threats but also storage and computation capabilities of RFID tags, servers and properties of protocol environments. Many studies in the literature regarding the security and privacy of RFID systems present a solution based on some cryptographic methods or algorithms. In real world, almost all of these solutions are not applicable because RFID tags are highly constrained devices in terms of storage, computing and power. There are also some EPC compliant solutions. However, these solutions can not provide adequate security level.

1.1. Contributions

In this thesis, we investigate the security and privacy requirements of RFID protocols which are designed for low cost RFID tags. Our main contributions are as follows:

- *Revealing security vulnerabilities in recent protocols:* Recently, several RFID authentication protocols have been proposed to meet the security requirements of RFID systems more accurately. We analyze the security of recent RFID authentication protocols with respect to two security requirements: mutual authentication and availability. We propose impersonation and de-synchronization attacks against recent RFID authentication protocols.
- *Designing more secure chaotic-map based RFID protocol:* Recently, researchers try to utilize chaotic maps in RFID authentication protocols. We analyze the

security of recently proposed chaotic-map based RFID protocols. We show that message generation arises some weaknesses and these protocols are vulnerable to tracking, tag impersonation and desynchronization attacks. The success probabilities of the proposed attacks are significant and their complexities are polynomial. Furthermore, we propose an RFID authentication protocol. Our protocol utilizes the Chebyshev chaotic map hard problem and conforms to the EPCglobal Class 1 Generation 2 (EPC C1-G2) standard. Our protocol eliminates the weaknesses of previous protocols.

- *PUF-based scalable and private RFID authentication:* We study privacy and scalability issues in RFID. Previously proposed RFID protocols require linear work in the back-end server while providing the desired level of privacy. Physically Unclonable Functions (PUFs) are a promising approach for solving the issues challenging RFID systems. We have utilized PUFs to solve scalability issue in RFID systems. We have proposed RFID authentication protocols based on PUFs. We prove that our protocols provide destructive privacy. Our protocols do not need a search operation on the server side to identify a tag. To the best of our knowledge, they are first to provide this privacy level without requiring lookup.

1.2. Outline

The remainder of the thesis is organized as follows:

- In Chapter 2, we present RFID system primer. The components of RFID systems are described.
- In Chapter 3, we describe the identified security and privacy requirements of RFID systems. Also, attacks related to privacy of tags and attacks related to security of tags and readers are described.
- In Chapter 4, we review some security models that are used to determine the privacy and security level of RFID protocols.
- In Chapter 5, we introduce some of related proposals and their attributes. We explain the basic principles of each proposal and their security vulnerabilities discovered in other works.

- In Chapter 6, we have presented security or privacy flaws in some recent RFID protocols that have received no attacks yet. Also, we have proposed some revisions, if possible, to eliminate weakness in these target schemes. Parts of this chapter are published in [4–6].
- In Chapter 7, we analyze the security of Chaotic-map based RFID protocols. We show that they are vulnerable to tracking, tag impersonation and de-synchronization attacks. Furthermore, we propose improved RFID authentication protocols based on the Chebyshev chaotic map hard problem. Our protocols eliminates the weaknesses of previous protocols. Parts of this chapter are published in [7, 8].
- In Chapter 8, we propose four RFID authentication protocols. Our protocols completely solve the scalability problem by utilizing master keys shared by all tags. They are *destructive* private under the Vaudenay-Model, which means that our protocols provide privacy against adversaries who are capable of destroying tags permanently. They provide resistance against physical attacks (corrupting) by using Physically Unclonable Functions (PUFs) as a secure storage mechanism to preserve the privacy of the tag. To the best of our knowledge, our protocols are first to provide such a privacy level without requiring lookup. Parts of this chapter are published in [9–11].
- In Chapter 9, we summarize the contributions of this thesis and point out some open problems.

As mentioned above, the main results presented in this thesis have been submitted or published in [4–11].

2. OVERVIEW OF RFID SYSTEMS

2.1. RFID Systems

There are three fundamental parts of characteristic RFID systems: tags, readers and a back-end server. RFID systems basically work as follows: firstly, the reader queries tags. After that, the reader collects tags' responses and sends them to the back-end server. In the back-end server, there is database in which identifiers and descriptive information about tags are stored.

2.2. RFID Tags

RFID tag is a radio device composed of a small antenna and a small silicon chip with some data storage, a control logic, a transmitter, a receiver and a power supply. The size of the antenna is the most important factor for producing small-sized RFID tags. The size of the antenna is proportional to the range of the tag and inversely proportional to the frequency. In [12], RFID tags are categorized based on three main criteria.

(i) **The Frequency of Operation:**

- **Low Frequency (LF) Tags:** Frequency range is 125-134.2 KHz. LF tags are less affected by metal or liquids. They are more expensive than other tags because they require copper in its production.
- **High Frequency (HF) Tags:** Frequency is 13.56 MHz. HF tags are cheaper than LF tags. However, they can be affected by metal or liquids.
- **Ultra High Frequency (UHF) Tags:** Frequency range is 860-960 MHz. They are cheaper to manufacture. However, they can be affected easily by metal or liquids.
- **Microwave Tags:** Frequencies are 2.4 GHz and 5.8 GHz. Microwave tags require line of sight for long distance communication.

(ii) **Powering Technique:** The power system of a tag can be completely powered by the incoming RF signal or it can have its own battery. Table 2.1 shows three categories of RFID tags [13].

Table 2.1. Comparison of passive, semi-passive and active tags

	Passive Tags	Semi-Passive Tags	Active Tags
Powering	Electromagnetic field	Battery	Battery
Transmission Power	Electromagnetic field	Electromagnetic field	Battery
Typical Read Range	3 m -5 m (UHF)	10 m -20 m	1000 m

- **Passive Tags:** Passive tags are smaller and cheaper than active tags because they do not have a battery. However, they must be very close to the antenna to work because they completely obtain its power from incoming RF signal by coupling the electromagnetic field of the antenna.
- **Semi-Passive Tags:** Semi-passive tags are between passive and active tags. They have a battery, but they rely on power obtained from the electromagnetic field of the antenna to transmit a message.
- **Active Tags:** Active tags have their own battery. They do not need to be powered by the electromagnetic field of the reader's antenna. They are able to send or receive data over longer distances [14].

(iii) **Reprogrammability:**

- **Read Only Tags:** "Read only" tags are programmed by the manufacturer.
- **Write-Once, Read Many Tags:** "Write-once, read many" tags are programmed by the customer at application level but they are not reprogrammed.
- **Rewritable Tags:** The customer can program "rewritable" tags more than once.

2.3. Electronic Product Code (EPC)

The Auto-ID Center at the MIT Department of Mechanical Engineering developed a universal scheme for object identification. This scheme is called as The Electronic Product Code (EPC). This scheme identifies each object uniquely with the help of unique numbers stored in RFID tags.

EPC tags store data in binary format shown in Table 2.2. EPC format is specified by the Header field. The General Manager Number indicates organization or company. Object class allows us to determine the different product groups. Each object is uniquely identified by the Serial Number.

Table 2.2. A binary format of EPC tag data standard

For 96 bit format			
Header	General Manager Number	Object Class	Serial Number
8-bit	28-bit	24-bit	36-bit

RFID tags are classified based on their functionality. Table 2.3 gives the tag class hierarchy. The tag class hierarchy is important. Because each RFID application needs different environmental, functional and computational requirements. Moreover, these applications require different security levels. RFID tags with different specifications can be developed based on this hierarchy.

2.4. RFID Readers

Another part of typical RFID systems is the reader composed of a separated or an integrated antenna, a power supply, a cryptographic encoding and decoding circuitry and a control unit. Reader queries tags to collect information from them by using radio signals. RFID readers have more computation and storage capabilities than those of tags. They are capable of carrying out all kind of cryptographic operations. Depending on the application, RFID readers can be manufactured in different size changing from the size of postage stamp to the size of a desktop personal computer [14].

Table 2.3. RFID tag class hierarchy

Class Name	Definition
Class I	Class 1 RFID tags are passive tags that have read only write once EPC code and a password. They also have a CRC for the verification of transmission.
Class II	Class II RFID tags are passive tags with limited read range. In addition to abilities of Class I, they have many ability and provision for security and privacy.
Class III	Class III tags contain a battery. They can be active or semi passive tags
Class IV	Class IV tags can communicate with other Class IV tags.

Readers come in different prices, depending on their functionality. Basic readers have their own computing capability. They can filter and store information and run applications. Advanced readers are able to read tags using different frequencies with the help of various communication protocols.

Readers can have serial, Ethernet, Wi-Fi or USB ports. They use one or more ports for connecting separated antennas. They are also able to connect to external devices, a computer and a network.

2.5. Middleware

Middleware is the interface needed among the back-end server and the reader. Each event must be managed in RFID systems to keep information among tags and the back-end server in synchronization. Middleware provides the right information to readers and the back-end server. For example, RFID readers can query the same tag several times per second. The middleware filters the raw data taken by the reader and forwards filtered data to the back-end server [14].

Middlewares have different functionality. Basic middlewares perform basic filtering. Advanced middlewares also perform additional functions to filter data. Some middlewares manage RFID readers. They monitor their health, configure them, send software updates and so on.

2.6. Back-End Server

The back-end database that holds all required information for tags can be a standard commercial database such as SQL Server and Oracle. It is assumed that there is a secure communication channel between the back-end server and readers. Depending on the application, the back-end server can run on a single PC or multiple mainframes networked together via global communication systems [14].

3. SECURITY AND PRIVACY ISSUES IN RFID SYSTEMS

In the literature, numerous security and privacy threats have been defined for RFID systems. Furthermore, researchers have proposed many solutions that try to overcome these threats. Wide deployment of RFID technology needs well designed security protocols that consider these threats.

3.1. Privacy Issues

The privacy of tags is the main privacy issue for RFID systems. Tags store some information about objects, persons or products. Because of these information, tags need privacy. Interests of adversaries determine two different privacy: behaviour privacy and data privacy. The former is that adversaries try to learn some behavioral information such as location and the latter is that adversaries try to learn information stored in tags.

3.1.1. Information Privacy

If tags store sensitive information, a passive adversary could get this information by simply querying tags. For example, when you borrow a book in a library, an attacker having a reader discover what kind of book you like, the name of the book and the name of the author, if a tag attached to your book does not authenticate the legitimate reader and sends messages in plaintext. This type of attack can be avoided by encrypting messages.

3.1.2. Location Privacy

If tags send their static ID in plaintext or send same encrypted message for each challenge, a passive adversary can distinguish a specific tag from others. Therefore, an attacker having a reader can trace the location of an individual. For example,

if you have a travel card with RFID chip, the attacker can find out locations you travel because usually RFID chips in travel cards give same response to all queries. Anonymous answers that are generated by encrypting static ID with some random values can avoid this type of attack.

3.2. Security Issues

Known attack types such as eavesdropping, replay attacks, man-in-the-middle attacks can be posed a threat to RFID systems.

3.2.1. Eavesdropping

In eavesdropping attacks, the communication channel between tag and reader is monitored by passive or active adversary. The forward channel (reader-to-tag channel) can be monitored from a long distance. Compared to readers, tags send very weak signals. Therefore, the backward channel (tag-to-reader channel) can be monitored from a short distance.

3.2.2. Replay attack

In replay attacks, passive adversaries try to eavesdrop messages and use them to impersonate one communication party to another.

3.2.3. Man-in-the-middle attack

Adversaries monitoring the communication between tag and reader try to modify messages and send them to communication parties. In this way, adversaries are able to get some information.

3.2.4. DoS attack

DoS attacks are applied to keep communication parties from taking some services. Adversaries can put numerous fake tags in communication range of the reader for slowing down the identification process of tags. Furthermore, adversaries can break the synchronization between communication parties by altering or dropping some messages.

3.2.5. Cloning attack

Adversaries can obtain information stored in the valid tag and write them to the fake tag. There is no way for the reader to test the validity of tags. For example, thieves can steal some products by using fake tags.

3.2.6. Compromising Attack

Strong adversaries can obtain information stored in low-cost tags by compromising them. Low-cost tags do not have tamper resistance because tamper-proofing increase their production costs.

3.2.7. Backward Traceability

Okhubo [15] defined the notion of forward security for RFID systems. An RFID security protocol provides forward security if a strong adversary compromising a tag at time t can not trace its past interactions that occurred before time t. Some studies such as [16, 17] have renamed forward security as *backward untraceability*.

3.2.8. Forward Traceability

Lim et al. [16] defined the notion of forward untraceability. An RFID security protocol provides forward untraceability if a strong adversary compromising a tag at time t can not trace its future interactions that occurred after time t.

3.2.9. Server Impersonation

Song and Mitchell [17] introduced a new type of attack for RFID systems: server impersonation attack. In this attack type, an adversary obtained secrets of a tag in any way try to impersonate the back-end server to the tag. These attacks can result in degradation of the synchronization between the tag and the back-end server.

3.3. Challenges

The production cost of a tag is the main factor that makes the RFID technology more widespread easily. The price of a tag can be reduced by using less silicon or another material cheaper than silicon in chip production. The scarcity of RFID tags in terms of storage and power makes providing security and privacy to RFID systems a challenging issue. When designing a protocol, these limitations should not be ignored.

An RFID chip can be designed up to $1mm^2$ in size. Stephen Weis says that 10000 gates can be squeezed into this area [18, 19]. Martin Feldhofer estimates the number of gates squeezed into this area as 20000. He also says that 5000 gates can be used for security purposes [20]. In [20], he implements encryption only AES block cipher in an RFID chip using 3628 gates.

Passive tags take its power from electromagnetic field of reader signal. The power supplied to tags can be increased by using low frequency instead of high frequency or increasing the transmission strength or using a larger antenna [18].

RFID systems have to be scalable because it is very difficult to manage billions of tags. Therefore, security solutions designed for RFID systems have to be scalable too. To achieve this, response times of tags should be short. That means tags should carry out cryptographic operations at high speed. We know that low-cost tags are limited in terms of computation so there is a trade-off between security and scalability. In addition to tags, the server should perform the process of identification quickly. The vast majority of security solutions providing high security level require linear search

on the back-end database for identification. These solutions can not be considered scalable for RFID systems having millions of tags.

4. RFID PRIVACY MODELS

RFID privacy model is formal definition of RFID protocols covering their security and privacy and the abilities of an adversary. RFID protocols must have security properties to provide tag-reader authentication and must have privacy properties to resist adversaries aiming to identify, trace or link tags [1]. Privacy models are used to determine the privacy level of RFID protocols. They can have several privacy levels according to the abilities of the adversary.

There are several proposed RFID privacy models in the literature. This section summarizes some of these models.

4.1. Avoine's Model

The first privacy model for RFID systems was proposed by Avoine [21]. This model gives the strong privacy notion of untraceability for RFID protocols. It has different level of privacy and defines different abilities for an adversary. It formalizes privacy by testing the attacker's capability to distinguish two known tags. It excludes the availability of side-channel information. In this model, an adversary uses some oracles to interact with tags and readers. She has also ability to use oracle called Reveal that returns the current state of the tag. The adversary using Reveal oracle can not use other oracles. Avoine defines untraceability as a privacy game. In this game, the adversary interacts with a target T. Then the adversary tries to choose tag T among two tags T_1 and T_2 . In order to make decision, the adversary can interact both T_1 and T_2. At the end of the game, RFID protocol is considered private, if the advantage of an adversary is not significant to win the game. Avoine defines the notion of existential and universal untraceability separately. In existential traceability, the adversary can trace the tag for restricted period of time, while in universal traceability, the adversary can trace the tag for all time periods.

4.2. Juels and Weis's Model

Juels and Weis modified Avoine's model by characterizing a very strong adversary [22]. Their model considers RFID protocols in which tags have correlated keys for authentication. Juels and Weis considered tags and readers as probabilistic interactive tuning machine. These machines have unlimited storage capacity, independent source of randomness and interfaces to send and receive messages. They defined a privacy game in which an adversary tries to distinguish between two different tags. In the first phase of privacy game, the adversary can use any tag and reader functionalities under the condition in which at least two tags remains uncorrupted. In the second phase, two uncorrupted tags are given to the adversary. The adversary is allowed to interact with them and to interact with and corrupt all other tags. In this model, side-channel information is used. Juels and Weis found a powerful desynchronization attack on Avoine's model. Avoine showed OSK protocol in [23] is secure under his model. In this model, this protocol is considered as insecure.

4.3. Lim and Kwon's Model

Lim and Kwon proposed a security model for untraceability [16]. They considered some restrictions in terms of access time and frequency to make Avoine's model [21] more flexible. They defined two access model: unrestricted access model (UA) and the restricted access model (RA). The restricted access model (RA) puts some restrictions to the adversary. For example, the adversary can read limited number of consecutive valid sessions. This model gives definitions for privacy notion of forward and backward untraceability.

4.4. Ouafi and Phan's Model

Ouafi and Phan provided a general untraceable RFID privacy model [24, 25]. They showed that recently proposed secure RFID protocols do not provide untraceable privacy as claimed. Lim and Kwon considered the protocol in [16] as secure according to their model. Ouafi and Phan showed that this protocol is not secure.

4.5. Vaudenay's Model

Vaudenay proposed a complete, hierarchical model for RFID security and privacy that classifies privacy in RFID [1]. This model has eight classes of privacy levels. Vaudenay showed that strong privacy is not possible. He also presented an open question whether symmetric-key cryptography based protocols provide forward privacy under a strong privacy model. Vaudenay classified the adversary according to the oracles she can query, the game and how she can interact with tags and readers. A strong adversary can access all oracles without any limitation. A destructive adversary can access all oracles. After she corrupts a tag, she can not interact with it because corruption destroys the tag. A forward adversary can access corrupt oracle at the end of the attack. A weak adversary can access all oracles except corrupt oracle. Vaudenay also classified the adversary according to the ability of seeing whether the RFID protocol is complete or not.

Vaudenay defined a privacy experiment in which the adversary having access to all oracles initiates an attack. The hidden table is given to the adversary and she analyse the hidden table by using the previously obtained information. At the end, the adversary wins if she gives the correct output. In this model, some privacy levels are determined for RFID protocols by using the polynomial-time algorithm, namely blinder. A blinder can see the same messages with the adversary and can simulate the Launch, SendReader, SendTag, and Result oracles. If the blinded adversary wins the privacy experiment with the similar probability obtained in the above experiment without needing Launch, SendReader, SendTag, and Result oracles, she is considered as trivial.

Paise and Vaudenay [26] extended the model in [1] by considering reader authentication in RFID tags. They showed that narrow-forward privacy is impossible when a tag is corrupted according to the model in [1]. They solved this problem by assuming that some temporary memory are erased when the tag is corrupted.

Armknecht et. al [27] revisited the model in [26] by considering several privacy notions in which adversaries are able to corrupt tags. They showed that achieving reader authentication and any notion of privacy is impossible under the assumptions in [26].

Avoine et al. [28] showed that an adversary can identify a specific tag by observing how much time the back-end server spends to identify this tag. They modified Vaudenay's Model [1] by adding this time notion.

Akgün et al [29] extended Vaudenay's model [1] by considering the notion of forward untraceability. They defined the minimum restrictions for forward untraceability by considering all protocol rounds between RFID reader and tags.

4.6. Ha et al.'s Model

In 2008, Ha et al. proposed a privacy model for RFID systems [30] based on random oracle and indistinguishability. They defined the notion of location privacy and two privacy game for indistinguishability and forward secrecy. They also defined the notion of weak location privacy and strong location privacy based on indistinguishability and forward secrecy. In 2010, van Deursen and Radomirović [31] showed that protocols that are location private according to models in [21], [22], [1] and [32] are not location private according to Ha et al.'s model. They also showed that protocols that does not provide location privacy according to models in [21], [22], [1] and [32] are location private in Ha et al.'s model.

4.7. Ng et al.'s Model

In 2008, Ng et al. [33] analysed RFID privacy model in [1] in great detail. They simplified eight privacy classes into tree classes under some assumptions. They also showed that tags need an additional reliable random source to achieve to strong privacy in addition to public key cryptography. Furthermore in contrast to model in [1] they showed that strong privacy is achievable without public key cryptography.

Ng et al. [34] classified synchronization based symmetric RFID authentication protocols into four types and determined the highest privacy level that these protocols can achieve based on the RFID privacy models in [1,26] and [33]. They claim that synchronization based symmetric RFID authentication protocols can not provide forward privacy.

4.8. D'Arco et al.'s Model

D'Arco et al. extended Vaudenay's model [1] by considering DoS attacks in [35]. They showed that an adversary can win the privacy experiment by stopping the activities of a tag so Vaudenay's model can not measure the privacy level of the protocol when DoS attacks occur. They redefined the privacy notion by considering DoS attacks. In privacy experiment, an adversary draws two tags and call a special query to make inactive one of them. After that, it will recognize the previous tag from the fact that no answer is received when querying it when it will draw a tag from the poll.

4.9. Ma et al.'s Model

Ma et al. [36] redefined the notion of unp-privacy originally defined in [30] by considering the behavior of the whole RFID system. Ma et al. proved that unp-privacy refers the notion of ind-privacy based the indistinguishability of two tags. They also proved that ind-privacy does not imply unp-privacy and strong or weak unp-privacy requires pseudorandom function (PRF) family or its equivalents minimally.

4.10. Bruso et al.'s Model

The pi-calculus [37] is a calculus of communicating systems that is used to describe concurrent computations whose structure may change during the computation. Abadi and Fournet [38] extended pi-calculus with primitive functions, value passing and equations between terms and introduced the applied pi-calculus. Bruso et al. [39] proposed a privacy model for RFID systems by using the applied pi-calculus. In this

model, an adversary communicates with an RFID tag using a tag interface that provides any access to the tag. The authors gave the notion of unlinkability. In the privacy game, an adversary tries to distinguish a tag having two interfaces from two tags having one interface. They also defined the notion of forward security that means an adversary having ability of corrupting tags can not distinguish a tag having two interfaces from two separate tags having one interface.

4.11. Lai et al.'s Model

Lai et al. [40] examined the RFID privacy model in [36] and showed that this model has a flaw while analysing 3-round mutual authentication RFID protocols. In the guessing stage of the unp-privacy game, an adversary does not allowed to use oracles to the challenge tag. Based on their observations, the authors proposed a privacy model for RFID systems. In this model, unp-privacy game was redefined by granting adversaries access to oracles in the guessing stage. This eliminates the drawbacks of previous model in [36].

4.12. Deng et al.'s Model

Deng et al. [41] proposed a definitional framework for RFID privacy based on a zero-knowledge. They defined the notion of adaptive completeness, matching sessions and authentication. They also gave the formal definition of zero-knowledge based RFID privacy (zk-privacy). In privacy experiment, an adversary is defined with two algorithms $(\mathcal{A}_1, \mathcal{A}_2)$. The adversary interacts with the set $\mathcal{B}$ of all tags and reader by using $\mathcal{A}_1$ and gets a set $\mathcal{C}$ of tags that are not corrupted and are not currently used in any protocol sessions. A challenge tag chosen from a set $\mathcal{C}$ is given to the adversary. The adversary interacts with the set of tags $\mathcal{D} = (\mathcal{B} - \mathcal{C})$ and reader by using $\mathcal{A}_2$. At the end of the experiment, an RFID protocol is considered as zk-private if an adversary can obtain information that are derived in the second stage of privacy experiment without interacting with the challenge tag. This privacy notion is called as zk-privacy because $\mathcal{A}_2$ does not interact with the challenge tag. The authors also gave definitions of forward and backward zk-privacy. They compared their framework with

previously proposed frameworks in [1, 22, 26, 30] and [36].

4.13. Hermans et al.'s Model

In [42], Hermans et al. showed that some previously proposed protocols have some weaknesses stemming from insufficient generality and unrealistic assumptions. They combined existing models by eliminating their drawbacks and proposed a new RFID privacy model based on the notion of indistinguishability.

5. OVERVIEW OF AUTHENTICATION IN RFID

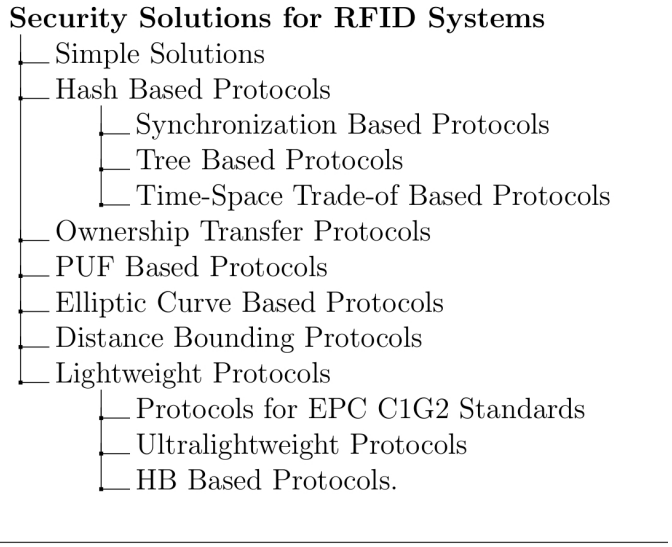

Figure 5.1. A typical RFID system

5.1. Simple Solutions

5.1.1. Kill Command

Auto-ID Center proposed a privacy scheme called "killing" [43]. Each tag is programmed to store unique password "PIN" at the time of manufacture. The tag deactivates itself when it takes this password. In EPC Class-1 Gen-2 standard, PIN is 32 bits long. In [44], Juels et al. showed that kill command approach is unlikely to be a fully satisfactory solution.

5.1.2. Blocker Tag

Juels et al. [44] proposed simple blocker-tag scheme that uses selective blocking for privacy protection. In this approach, a blocker tag overloads the tree-walking singulation protocol by simulating the full spectrum of possible serial numbers for tags to engage in a passive form of jamming. In [45], Juels and Brainard proposed a variant of blocking scheme called soft blocking. In this approach, a soft blocker tag has

a software module "tag privacy agent TaPA" instead of "blocker"-like functionality. This software module is used to express the privacy preferences of tag's owners to RFID readers.

5.2. Hash Based Protocols

5.2.1. Weis et al.'s Protocols

Weis et al. [46] proposed HashLock, the first hash based RFID authentication protocol. This protocol uses only a hash function for providing cryptographically controlled identification. The reader starts a protocol session by sending request to tags. A tag receiving this request send its $metaID$ to the reader. The reader compares the respond of a tag with all the $metaID$s in its database. If it finds a match, it sends the key k to the tag. The tag compares $h(k)$ with its $metaID$ to validate the reader. If the reader is valid, the tag sends its ID to the reader. HashLock is described in Figure 5.2. The HashLock protocol does not provide location privacy because tags always respond with their $metaID$ to the reader's query. Furthermore, it does not provide resistance to any type of attacks because adversaries can easily eavesdrop the key k.

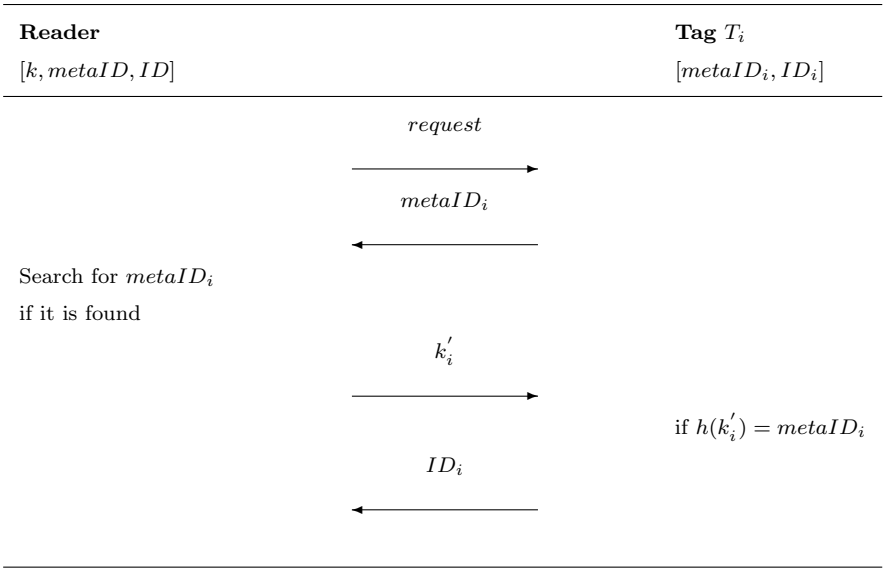

Figure 5.2. The HashLock protocol

Weis et al. [19] proposed another RFID identification protocol which is the modified version of the HashLock protocol. This protocol uses a pseudo-random number generator in addition to a hash function. Tags respond to the reader with the hashed value of their ID and pseudo-random number r. The reader searches the proper key by computing $h(ID, r)$ for each ID in its database. Therefore, the reader performs linear search operation for each identification process. In randomized HashLock, sending tags' ID raises many security and privacy problems.

Many approaches have been proposed to avoid brute force searches that are done to identify tags. These approaches are classified by Juels [47] into three type: synchronization based protocols, tree based protocols, time-space trade-off based protocols.

5.2.2. Synchronization Based Protocols

In synchronization approach, the reader and the tag have a value in synchronization. When responding to the reader, the tag uses this value. The reader can store several consecutive values for each tag. After each successful authentication session, the reader and the tag update this value to keep synchronization.

Okhubo et al. proposed an RFID authentication protocol based on hash-chains [23]. The reader keeps seeds of hash chains in the back-end database. In each session, tags take the hash of their identifier and send the hashed value to the reader. After sending the hashed value, tags take the hash of their identifier using another hash function and replace their identifier with the hashed value. The reader identifies tags by constructing hash chains from each initial value. The proposed protocol is forward secure because it is infeasible to reverse hash functions. However, adversaries can use previous responses of tags for replay attacks. The protocol is summarized in Figure 5.3

Okhubo et al. [15] proposed modified version of the protocol in [23]. Modifications aim to speed up the process of identification. The authors proposed not to update the identifier after each identification request. They use a counter c and define upper bound

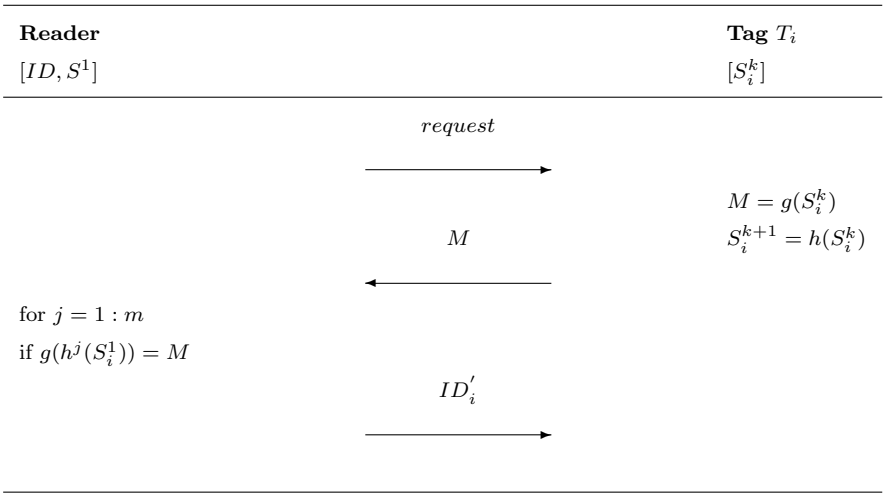

Figure 5.3. The OSK protocol

for it. The counter is augmented after each identification request. If the counter reaches its upper bound, the tag updates its identifier. However, this protocol degrades forward security because the last c interactions of the tag can be traced without corrupting it and unsuccessful authentications make this protocol vulnerable to replay attacks.

Henrici and Müller proposed another synchronization based protocol for RFID systems [48]. In the tag side, read attempts are counted and sent to the reader to prevent replay attacks. After each successful authentication, the counter is reset. However, a passive attacker can perform tracking attacks by augmenting the counter for a specific tag. In [49], Avoine showed that a passive adversary can easily break the synchronization between a tag and the back-end server by replacing some messages.

Dimitriou [50] proposed a mutual authentication protocol for RFID systems and solved the desynchronization problem entirely. In this protocol, there is a shared secret between the the back-end database and the tag. After each successful authentication, both parties update the shared secret to prevent tracing attacks. However, this protocol is vulnerable to tracing because the shared secret remains unchanged until the next successful authentication.

5.2.3. Tree Based Protocols

Tree structure was first used by Molnar and Wagner for their RFID protocol [51]. Each node of the tree has a unique key and each leaf node of the tree represent a different tag. A tag stores keys in the path from the leaf node in which the tag is represented to the root node. Authentication is done by running a challenge-response protocol for each key stored in a tag. The reader traverses nodes level by level. It visits the child nodes of the root node for the first response of the tag. It visits the child nodes of the node which is determined from the child nodes of the root node for the second response of the tag and etc. For example, there are N tags in the systems and α is the branching factor of the tree. Therefore, each tag stores $log_\alpha N$ keys and the reader uses $log_\alpha N * \alpha$ keys for an authentication. Figure 5.5 describes one round of the proposed protocol and Figure 5.4 shows a sample tree. In this protocol, the key-updating seems infeasible because of the shared keys among tags. Therefore, it does not provide forward security. In his thesis [52], Molnar stated that "the updating mechanism leads to the reader work in the number of possible time periods, but the reader does not know at which time period the tag is."

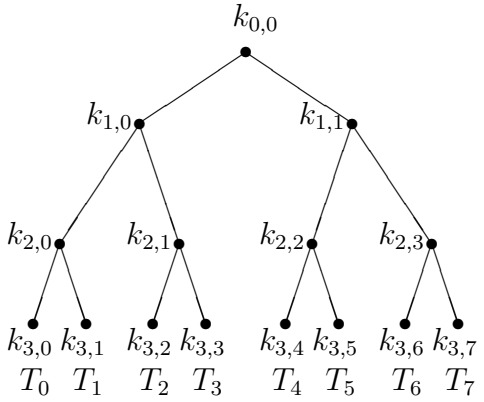

Figure 5.4. A binary balanced key tree with eight tags.

A single session of Molnar and Wagner's protocol requires $O(\log N)$ communication rounds where N is the number of tags. This protocol can not be used in two-round legacy systems. Dimitriou modified Molnar and Wagner's protocol to be used in legacy systems [53]. His protocol uses one message from tag to reader.

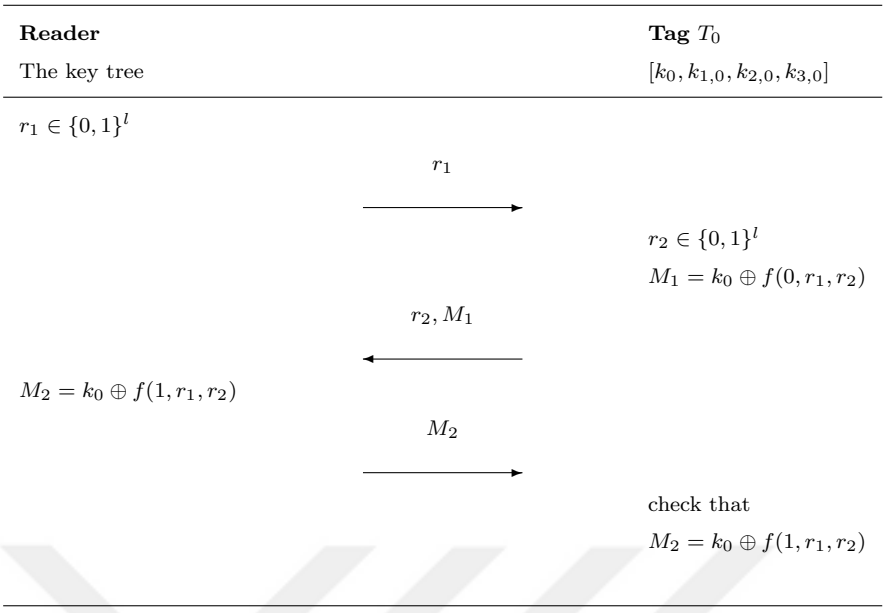

Figure 5.5. A round of Molnar and Wagner's protocol

Lu et. al [54] proposed a tree based RFID Private Authentication protocol. The authors proposed a mechanism for solving the key-updating problem of tree based protocols. In their mechanism, each non-leaf node has state bits to track the update status of its child nodes. The non-leaf node updates its key after its child nodes update their keys. The authors claim that their protocol is secure against active and passive attacks. However, this proposal does not provide a complete solution to key-updating problem. Tags can not update some of their keys so there are still key relationships between tags.

Wang et al. [55] proposed to use sparse tree structure for RFID authentication. In this solution, tags store their binary paths in the tree as a secret. Each tag has a unique secret so there is no key relationship between tags. After each successful authentication, the reader removes the path of the identified tag from the key tree, updates the key and inserts a new path corresponding to the updated key. This protocol is not location private and information private. Akgün et al. [56] presented an disclosure attack in which the path key of a tag is disclosed by using weakness of the hash chain.

5.2.4. Time-Space Trade-off Based Protocols

Avoine et al. [57,58] proposed an RFID authentication protocol that is the modified version of the protocol in [23]. One modification is that the reader queries tags with fresh challenges. This modification prevents replay attacks. Another modification is that the reader sends a message to tags after tag identification. This modification provides reader authentication. Avoine et al. [58] finds out that the Hellman's method used for the key breaking problem [59] can be used to solve the key search problem in RFID authentication. The last modification is to use Hellman's method to speed up key lookup process.

5.2.5. Lim and Kwon's Protocol

Lim and Kwon [16] defined a new concept: forward untraceability. They also proposed an ownership transfer protocol that provides both backward and forward untraceability. This protocol uses probabilistic way to update tag secrets if the authentication is successful. It uses deterministic updating mechanism otherwise. Lim and Kwon claim that forward untraceability can be provided under some assumptions. Their assumption is based on the probability of missing some interactions between the reader and the tag. The protocol becomes forward untraceable if an attacker can not eavesdrop a valid session between the reader and the tag. This protocol uses backward key chain to resist server impersonation attacks. The cost of validation process is very high because the backward hash chain is in reverse order. In [24, 25], authors show that this protocol does not provide location privacy.

5.2.6. Song and Mitchell's Protocol

Song and Mitchell [17] proposed an RFID authentication protocol and defined the notion of server impersonation. The protocol is summarized in Figure 5.6. The proposed protocol provides resistance to forward traceability and server impersonation under some assumptions. It also provides improved performance of the protocol in [16] in memory space, computation time, communication overhead with same security and

privacy features. Deursen et al. presented a replay attack on this protocol. Cai et al. [60] showed that inexpensive operations such as $\ll$ and $\oplus$ make the protocol in [17] vulnerable to some attacks. They presented an attack in which the passive adversary impersonates any legitimate reader to tags and break the synchronization between tags and the back-end server. Cai et al. also proposed an RFID authentication protocol that eliminates the vulnerabilities of the protocol in [17] by increasing the computational cost in the tag side. Akgün et al. [61] presented how a strong adversary break forward untraceability of Song and Mitchell's protocol. They also proposed modifications to this protocol. Their modified protocol provides resistance to server impersonation and forward traceability. Akgün et al. claim that their modified protocol provides resistance to server impersonation without any assumption. However, Kardaş et al. [62] showed that this protocol provides resistance to server impersonation under an assumption.

Reader		**Tag** T_i
$[(u_i, t_i)_{new}, (u_i, t_i)_{old}]$		$[t_i]$
	r_1	
$r_1 \in \{0,1\}^l$	$\longrightarrow$	
		$r_2 \in \{0,1\}^l$
	M_1, M_2	$M_1 = t_i \oplus r_2$
	$\longleftarrow$	$M_2 = f_{t_i}(r_1 \oplus r_2)$
Search for t_i		
$r_2 = M_1 \oplus t_i$		
if $M_2 = f_{t_i}(r_1 \oplus r_2)$,	M_3	
$M_3 = u_i \oplus (r_2 \gg l/2)$	$\longrightarrow$	
$u_{i(old)} = u_i$		$u_i^{'} = M_3 \oplus (r_2 \gg l/2)$
$t_{i(old)} = t_i$		if $h(u_i^{'}) = t_i$
$u_{i(new)} = (u_i \ll l/4)$		$t_i = h((u_i^{'} \ll l/4) \oplus$
$\oplus (t_i \gg l/4) \oplus r1 \oplus r2$		$(t_i \gg l/4) \oplus r1 \oplus r2))$
$t_{i(new)} = h(u_{i(new)})$		

Figure 5.6. Song and Mitchell's protocol

5.3. Scalable Protocols

Wu and Stinson [63] proposed an RFID protocol to solve scalability problem. The proposed protocol provides security and privacy by using the difficulty of reconstructing a polynomial with noisy data. The time complexity of identifying a tag is to solve *mb*

Table 5.1. Comparison of hash based protocols

Protocol	Information Privacy	Location Privacy	Tag Impersonation	Replay Attack	DoS Attack	Backward Traceability	Forward Traceability	Server Impersonation
Weis et al. [46]	−	−	−	−	−	−	−	−
Weis et al. [19]	∘	∘	−	−	−	−	−	−
Okhubo et al. [23]	∘	∘	−	−	∘	∘	−	−
Okhubo et al. [15]	∘	∘	−	−	∘	−	−	−
Henrici and Müller [48]	∘	−	−	−	∘	−	−	−
Dimitriou [50]	∘	−	−	−	−	∘	−	−
Molnar and Wagner [51]	∘	∘	∘	∘	∘	−	−	−
Dimitriou [53]	∘	∘	∘	∘	∘	−	−	−
Lu et. al [54]	∘	∘	∘	∘	−	−	−	−
Wang et al. [55]	−	−	∘	∘	−	∘	−	−
Akgün et al. [56]	∘	∘	∘	∘	∘	∘	△	∘
Avoine et al. [57, 58]	∘	∘	∘	∘	∘	∘	−	−
Lim and Kwon [16]	∘	−	∘	∘	∘	∘	−	∘
Song and Mitchell [17]	∘	∘	−	−	−	∘	+	△
Cai et al. [60]	∘	∘	∘	∘	∘	∘	+	△
Akgün et al. [61]	∘	∘	∘	∘	∘	∘	△	△

$^{\circ}$ provided

$^{\triangle}$ provided under the assumption in [16, 17]

$^{+}$ partially provided under the assumption in [16, 17]

$^{-}$ not provided

polynomials of degree k where m, b and k are predefined security parameters. Typical values for m and b are given as 16 and 8, respectively in [63]. A server performs 128 polynomial operations to identify a tag, so it has the same calculation burden as a tree-based system with 2^{128} tags [64]. The identification complexity of this protocol is $\mathcal{O}(1)$. However, an adversary repeatedly querying a tag Q_{max} times can trace the tag because the maximum number of queries that the tag will answer correctly is limited to Q_{max}, which means this protocol is not private in the Vaudenay-Model.

Alomair et al. [64] proposed an RFID protocol with constant-time identification. They designed a special database infrastructure on the back-end server. Thus, the reader is able to obtain a tag's data in an extremely short time. In the initialization phase, N pseudonyms are chosen. For each pseudonym p_i, the hash value $h(p_i, c)$ is computed for all c from 0 to the maximum counter value C. These values are stored in the database for tag identification. Each tag stores a counter value and increments the counter after each reader query. If the authentication is successful, the counter value is reset to 0. The identification complexity of this protocol is $\mathcal{O}(1)$. However, an adversary querying a tag C consecutive times can identify its past interactions. In [6], another traceability attack on Alomair et al.'s protocol was presented. This protocol is not private in the Vaudenay-Model.

5.4. Ownership Transfer Protocols

Ownership transfer is an important problem in RFID systems because the owner of an RFID tag can be changed several times during its lifetime. For example, Alice buys a gift from souvenir shop. Therefore, the ownership of a tag attached the gift must be transferred to Alice's reader from the back-end server of souvenir shop. When Alice gives the gift to Bob, the ownership transfer must be carried out between Alice's reader and Bob's reader. After the ownership of a tag is transferred, the authorization to read the tag is transferred to the new owner [16]. The old owner should not interact with the tag and should not identify its interactions anymore. Therefore, we can say that the issue of secure ownership transfer is related with forward untraceability [17].

The first RFID protocol that enables ownership transfer was proposed by Molnar et al. [65]. The authors use tree structure to store keys and a Trusted Center (TC) to provide desired privacy. The proposed protocol also enables time-limited delegation. Thus a reader can perform limited number of interaction with a tag without needing the Trusted Center. The ownership transfer takes place if the old owner of a tag is already delegated to some leaves of the key tree. The authors proposed to perform the ownership transfer in two ways. In the first method, the new owner of a tag learns k leaves that are delegated to the old owner from the Trusted Center. Then it queries the tag $k + 1$ times. Therefore the tag updates its current state $k + 1$ times and it becomes unreachable from Alice. In the second method, the new owner performs a mutual authentication with a tag by using its current leaf. Then it increments the tag counter by sending new counter c' and secrets of the leaf c' to the tag. The tag checks validity of the new owner. The weak point of this scheme is that the old and new owner must trust the same Trusted Center [66]. Furthermore, a reader can interact with a tag for a limited number of times. Thus this scheme provides time-limited delegation rather than ownership transfer [16].

Saito et al. [67] proposed an ownership transfer protocol that uses trusted third party (TTP). The proposed scheme is only used for changing a secret key stored inside a tag. Therefore, it can be used with other RFID privacy and security protection protocols. In this protocol, the old and new owner of a tag use a secure channel to communicate each others. Firstly, the new older takes the secret key from the old owner. Then, the new owner communicates with the tag with the help of TTP to update the secret key. However, this scheme is vulnerable to tracking attacks if the old owner of a tag does not update the secret key before sending it to the new owner. Furthermore, an adversary can learn the secret key shared between TTP and tags by tampering a tag [67]. The authors also proposed another ownership transfer protocol for two-party model. The security of the protocol depends on the difficultly of tapping the backward channel between reader and tag. In the protocol, the nonce created by a tag is used as an encryption key between the tag and the new owner. The new owner changes the key with the help of the encryption key. Moreover, this protocol gives the old owner the opportunity of changing a key which will be sent to the new owner.

Osaka et al. [68] proposed a security protocol that achieves ownership transfer efficiently. In the proposed protocol, the server stores a symmetric key k, ID and $E_k(ID)$ for each tag in its database. $E_k(ID)$ is used as a tag identifier. When ownership transfer takes place, the old owner protects its privacy by updating a symmetric key k. Then the old owner transfers necessary data with the updated symmetric key k' to the new owner in secure channel. The new owner updates the received symmetric key k' to protect its privacy. However, the proposed protocol has no resistance to DoS attacks and tracing attacks [66].

Lim and Kwon [16] proposed an ownership transfer protocol. In the proposed protocol, the back-end server stores the previous and the current state of a tag to prevent DoS attacks. When the new owner of a tag wants to take over the tag, it receives the previous and current state of the tag from the back-end server via secure channel. Then the new owner queries the tag to initiate a legal session in which the tag will updates its secrets with random numbers shared with the new owner. In this protocol, the backward hash chain is used for server validation. Therefore, only the new owner and the back-end can start a legal session in which the tag updates its secrets. However, Song [66] presented an attack in which the privacy of old owner is violated.

In [69], the authors proposed a method for anonymous-ownership transfer. In their method, the ownership transfer is carried out by changing two secrets stored inside the tag. They call their method as anonymous ownership transfer because there is no sign inside the tag that helps to recognize the owner. After the product is sold, two secrets stored inside the tag act as username and password which will be taken from swiping or proximity card key and PIN respectively [69].

In [70, 71], two methods for ownership transfer of an RFID tag were proposed. In the first method, the authors assume that both the new and old owner of a tag rely on the same database. The new owner of a tag receives required values from the tag and sends them to the old owner. The old owner sets the current counter of the tag to its maximum value. At the end, the new owner and the tag updates their secrets

by using a hash function. However, the scheme does not resist replay attacks because an adversary can use the response which is created by hashing the tag secret and a random number to impersonate a tag [66]. In these studies [70, 71], the authors also consider a situation in which the new owner does not trust the database that is trusted by old owner. For example, the customer buying a product does not want to trust the retailers database. For this situation, the authors proposed another method that uses symmetric key cryptography. In this method, the old owner and the tag update the stored secrets before the ownership transfer takes place to protect its privacy from the new owner [66]. After the old owner transfers the stored secrets to the the new owner, the new owner and the tag update the stored secrets with some random values to protect their privacy from the old owner.

In [66], Song described the principals of a secure transfer of ownership. She also proposed ownership transfer protocol by considering these principals. There are two phases in the protocol: transfer phase that is same as the authentication protocol in [17] and updating phase. In updating phase, the stored secrets inside the tag are updated. Cai et al. [60] presented a de-synchronization attack on the updating phase. The same attack is shown in [72].

Dimitriou proposed a protocol that enables secure delegation and ownership transfer of tags [73]. It is assumed that the new owner receives all relevant information about the tag by using a secure channel or all relevant information is written on a receipt which is given to the new owner. After that the new owner updates the secret stored inside the tag. For this purpose, the ownership transfer protocol was proposed. In this protocol, after identification of a tag finishes, the tag and the reader share an additional random number R which is needed to update the secret. The protocol is summarized in Figure 5.7.

5.5. Physically Unclonable Function Based Protocols

Physically Unclonable Functions (PUFs) can be described as physical random functions which are unclonable in a physical sense. Each IC in PUFs has different path

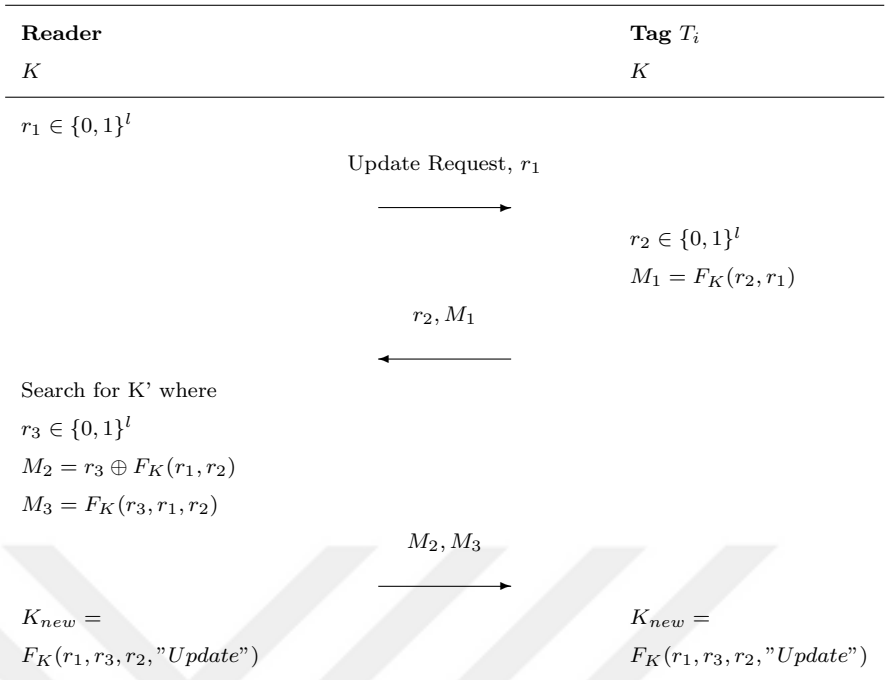

Figure 5.7. Dimitriou's protocol

delays and gate delays because of random manufacturing variations. Although it is very easy to construct and evaluate a random PUF, it seems impossible to have two PUFs with the same challenge-response behavior. PUFs are affected by the environmental noise and their physical properties when they are generating responses. That means a PUF generates different responses when it is queried consecutively with the same challenge. Because of this nature, PUFs can not be used for security protocols. Fuzzy Extractor makes PUFs deterministic by removing noise effects.

Ranasinghe et al. [74] proposed to use PUFs for RFID identification. The reader stores precomputed challenge-response pairs in back-end database. These challenge-response pairs are generated for each tag. The reader queries tags with one of the predefined challenges. After getting the response from a tag, the reader makes linear search in the number of tags for identification. Although this protocol uses PUFs to be secure, it is vulnerable to replay attacks. An adversary can use a challenge-response pair to trace a specific tag. The authors overcome this problem by encrypting challenges and responses.

Pim and Batina [75] used PUFs in their public key cryptography based RFID protocol. They enabled off-line authentication by using PUFs as secure storage. In this solution, tags derive their keys by using PUFs when they are queried.

Lenoid and Gabriel [76] proposed PUF based RFID identification protocol. Their protocol provides security by using one-time pads, pseudo-random functions and PUFs. In the initialization phase, the reader generates an entry $(ID, s(ID), s^2(ID)..., s^k(ID))$ for each tag where s is a PUF varies from tag to tag. Tags respond to the query with their current identifiers ID. After that tags update their identifiers by computing $p(ID)$. If the identification is successful, the reader deletes previous identifiers of the identified tag. This deletion process is important to prevent replay attacks. An attacker can carry out successful DoS attacks by forcing the specific tag to make the update process more than the number k.

Devadas et al. [77] designed and implemented PUF-based ICs (integrated circuits) for RFID tags. They also proposed an RFID identification protocol using the proposed ICs. In this protocol, a trusted party stores randomly chosen challenges and calculated responses in its database for future authentications of a tag. When the tag needs to be authenticated, the trusted party sends one of the challenges that has never been used before and gets the response of the tag. If the response of the tag matches with the stored response, the trusted party authenticates the tag. The authors claim that their protocol is secure and robust against replay attacks.

Bringer et al. used Physical Obfuscated Keys (POKs) to modify tree-based protocols. POKs are strongly related to PUFs. In tree-based protocols, compromising one tag reveals the secret information of other tags because these protocols use correlated keys. They aim to thwart this problem by generating each key via POKs. This protocol provides location and information privacy and prevents tag impersonation attacks.

Sadeghi et. al used PUFs to design the first destructive-private RFID protocol according to the Vaudenay's privacy model [1]. They modified the weak private protocol in [1] by utilizing the secure storage property of PUFs. Tags uses PUFs to generate

their keys with random input s. It is impossible to obtain keys for a strong attacker corrupting tags. Figure 5.8 summarizes Sadeghi et. al' protocol.

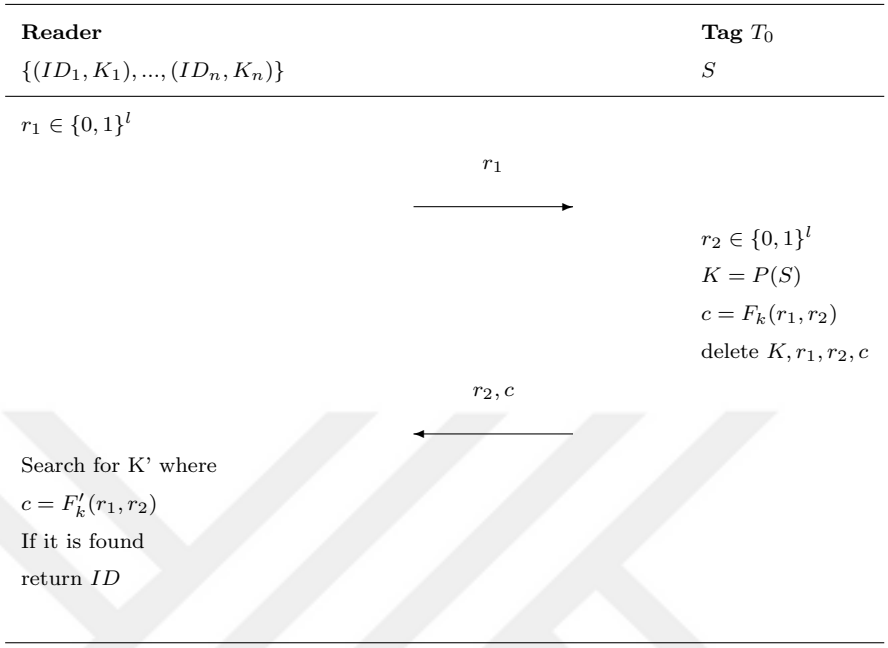

Figure 5.8. Sadeghi et. al's protocol

Kulseng et al. [78] proposed a solution for mutual authentication and ownership transfer for RFID systems. Their solution is based on PUFs and Linear Feedback Shift Registers (LFSRs). Kapoor and Piramuthu [79] presented a de-synchronization attack to the Kulseng et al.'s ownership transfer protocol. In [80], Safkhani et al. presented a secret parameter disclosure attack on the mutual authentication protocol that needs two sequential successive protocol runs. They also showed how the adversary retrieves the PIN value. Kardaş et al. [81] showed that mutual authentication protocol is not resistant the message blocking attack, message injection attack and traceability attack. They also showed that the ownership transfer protocol has privacy leakage because a tag becomes traceable by the old owner after the transfer of ownership.

Choi et al. [82] proposed a low-cost RFID authentication protocol based on a PUF-based encryption processor (PEP). PEP consists of an encryption, a decryption, and an ECC (error correcting code) modules. In the proposed protocol, the reader sends an encrypted challenge to tags. Tags decrypt the challenge and encrypt their response by using PEP. At the end, this unique challenge-response pair is used for

tag authentication. The authors claim that their protocol resists modelling, physical, tracking and spoofing attacks.

Kardaş et al. [83] proposed an RFID mutual authentication protocol. The proposed protocol utilized PUFs to provide unique identities to the tags and to provide resistance against side-channel attacks. The proposed protocol provides destructive privacy in the Vaudenay-Model with $\mathcal{O}(N)$ search complexity.

5.6. Elliptic Curve Based Solutions and Protocols

Recent studies shows that public-key cryptography is applicable on RFID tag. This studies focus on elliptic-curve cryptography (ECC) because it needs more shorter key size that RSA to provide same level of security. In recent years, the researchers have proposed some ECC processor design for RFID tags [84–86] and elliptic curve based protocols for RFID systems [87–94].

5.6.1. Elliptic Curve Processors for RFID Tags

The first ECC processor design for RFID tags was proposed by Batina et al. [95]. They presented an ECC processor architecture based on identification schemes such as Schnorr's [96]. Their design requires between 8500 and 14000 gates.

In [84], a new ECC processor design was proposed. The authors consider different size binary fields and different size area requirements while designing the ECC processor. The authors claim that ECC based systems are suitable for applications which needs lower security. The power requirements of the ECC processor affects the performance of the processor on the constrained devices like RFID tag.

The work of Bock et al. [97] presented an implementation of an integrated authentication module on an RFID tag. In this module, responses for each challenge are the result of elliptic curve point multiplication. The authors claim that their design is resistant against side-channel attacks such as timing, simple power, differential power,

and fault attacks.

In [85], an ECC processor for RFID tags was presented. The proposed processor is implemented on a 180 nm CMOS technology with size of 15K GE and low power consumption. It performs 163-bit ECC point-multiplications.

Lee et al. [98] proposed an ECC processor over $GF(2^{163})$ for RFID tags. Considering the restrictions on the gate area and the number of cycles, the authors introduced some optimization techniques to reduce the number of registers and gate area of register file. They also proposed a method to compute modular operations efficiently. The most compact processor for one point multiplication was implemented by using 10.1 Kgates with 276 Kcycles. The overall processor implementation requires 12.5 Kgates.

Braun et al. [86] implemented an elliptic curve processor by considering side channel attacks. They also proposed an authentication protocol that needs computations in the factor ring Z/qZ where q is the order of the base point on the elliptic curve. Their protocol also provides location privacy and forward security.

The work of Luo et al. [99] showed that ECC processor is implemented by using 16.9 Kgates. The proposed processor performs one elliptic curve point multiplication in 36174 clock cycles and has a power consumption of 6.607 μW at 1.28 MHz using TSMC 0.18 μm low-voltage cell library.

5.6.2. Elliptic Curve Based Protocols

An elliptic curve based zero knowledge authentication protocol for RFID systems was presented in [91, 100]. The proposed protocol provides mutual authentication of a reader and a tag. The authors claim that their protocol is secure even forward secure and scalable on the number of tags.

In [87], Lee et al. showed that ECDLP (Elliptic Curve Discrete Logarithm Problem) based authentication protocols proposed in [75] and [101] are not suitable for

RFID systems. In these protocols, the public key of an RFID tag is considered as its ID. Lee et. al showed the these protocols are vulnerable against tracking attack. Furthermore, Lee et. al proposed an ECDLP based RFID protocol that minimizes the computational workload of a tag. The protocol is summarized in Figure 5.9.

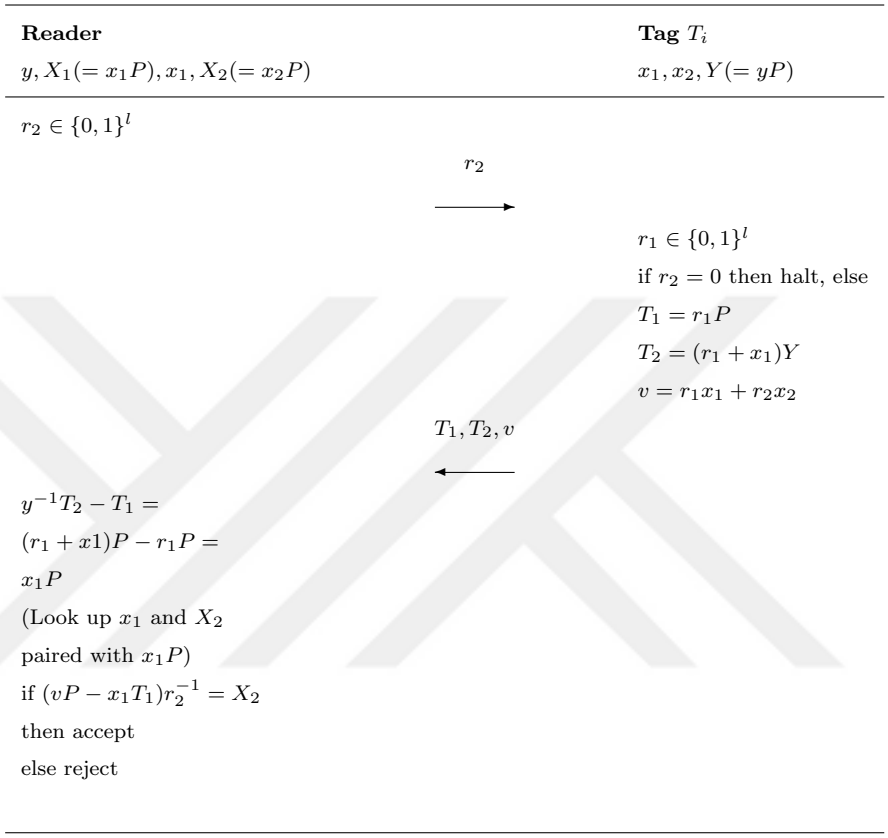

Figure 5.9. Lee et al's protocol

Bringer et al. [102] performed a tracking attack on the protocol in [87]. Thus, an adversary needs two protocol session of the same tag to track it. Bringer et al. also showed the vulnerability of the same protocol against cloning attack. Thus, an adversary can impersonate a tag several times if she obtains the communications of the same tag three times. In the same study, Bringer et al. proposed an efficient identification protocol for RFID systems. The proposed protocol is a modified version of the Schnorr scheme [96]. The authors claim that their protocol is zero-knowledge, Narrow-Strong Private in Vaudenay-Model [1] and scalable.

In [89], Lee et al. proposed six authentication protocols to eliminates the vulnerabilities of the protocol in [87] pointed in [102]. The authors defined 3 different scheme:

ID-transfer scheme for tag authentication, the password-transfer scheme and server's authentication scheme. They combines these schemes in different ways to construct different authentication protocols that meet the security requirements of different applications.

In [103], Lee et al. revised their previous study [89] to eliminate some vulnerabilities and proposed three different ECDLP based authentication protocols: ID-transfer scheme, Pwd-transfer scheme and ID Pwd-transfer scheme. They claim that two of proposed protocols are wide-strong private and the other one is wide-weak private in Vaudenay-Model [1]. They also proposed an efficient search protocol for server or reader. Hermans and Frederik presented man-in-the-middle-attacks on the ID-transfer scheme and ID Pwd-transfer scheme in [104]. They showed that ID Pwd-transfer scheme and ID-transfer scheme is not wide-strong private as claimed. In [105], Deursen and Radomirović showed man-in-the-middle-attacks in which a wide-weak adversary can trace a tag on these protocols.

Lee et al. [94] proposed two authentication protocols for RFID systems that use Elliptic Curve Cryptography (ECC). In this study, the authors revised ID-transfer scheme to eliminate the vulnerability to the man-in-the-middle-attack which is carried by wide attacker. They claim that the new ID-transfer scheme is wide-weak privacy-preserving. They also revised Pwd-transfer scheme and ID Pwd-transfer scheme. The second proposed protocol is constructed by combining ID-transfer scheme and Pwd-transfer scheme. It is vulnerable to man-in-the-middle attacks which are carried by a narrow-strong and a wide-weak attacker. The authors claim that the second proposed protocol is narrow-strong and wide-weak privacy-preserving if it is executed parallelly with the ID-transfer scheme. In [105], Deursen and Radomirović showed that a wide-weak attacker performs man-in-the-middle attack on the proposed protocols.

5.7. Distance Bounding Protocols

An adversary between tag and reader can authenticate itself by relaying exchanged messages to both parties if we use classical security protocols for RFID sys-

tems. In [106], Desmedt et al. describe a new kind of relay attack called "Mafia fraud" that is successful on any authentication protocol. In this attack, an adversary relays messages between a legitimate tag and a legitimate reader for passing authentication. Because the messages are not changed, the legitimate tag and reader are unaware of attack so preventing relay attacks are troublesome. In terrorist fraud, a dishonest tag collaborates with an adversary to convince a legitimate reader. The dishonest tag does not give to the adversary any information that make successful her in future attacks. Therefore, whenever the adversary attacks the system, she always need the help of the legitimate tag.

Although some RFID systems have short reading ranges, they are vulnerable to terrorist and mafia frauds. The adversary is successful to be authenticated by relaying the signals between valid verifier and valid prover. For instance, RFID tags are used to collect highway tolls. The customer can pass the tolling point successfully if the vehicle is very close to RFID reader. Assume that there is a waiting queue for passing. An attacker in the waiting queue can relay messages between the legitimate reader in the tolling point and the tag on another vehicle. Therefore, the attacker can pass the point without paying the toll. Hancke [107] presented the first successful mafia fraud attack. The vulnerabilities of contactless smart cards are defined and low cost and practical relay attacks on these cards are presented in [107, 108]. Outcomes of some live experiments (UK's EMV implementation, Chip and Pin) are also revealed [109].

One way to prevent relay attacks is to detect delay in prover's expected response. The pre-calculated the round trip time of challenge-response pairs can be used to determine the physical distance between prover and verifier [110]. Therefore, delays in the prover's response can be detected by using distance bounding protocols.

Brands and Chaum [111] proposed the first known distance bounding protocol in the literature. This protocol was designed to prevent mafia fraud by the help of rapid bit exchange phase. In the rapid bit exchange phase, the verifier creates a random bit and sends it to the prover. After getting the challenge bit, the prover creates a random bit and sends it to the verifier. Both parties repeat this phase n times. Mafia fraud

resistance is provided with the help of the signature formed by using all bits at the end of the rapid bit exchange phase. The success probability of mafia attacker is $(1/2)^n$. However, this protocol is vulnerable to distance fraud. Distance fraud can be realized by sending response bits without waiting for challenge bits. Brands and Chaum make the success probability of an attacker $(1/2)^n$ by adding dependency between challenge bits and response bits.

Capkun et al. [112] modified Brands and Chaum's protocol [111] to enable mutual authentication. The prover and verifier concatenate their identities with challenge and response bits from the rapid phase and compute the MAC of this bit string. Both parties authenticate other party by checking the correctness of MAC value.

Another RFID distance bounding protocol was proposed by Hancke and Kuhn [113]. In the first phase of the protocol, the verifier sends a random number to the prover. The verifier and prover calculate the MAC value of random number. In the fast bit exchange phase, the prover calculates response bits by using the corresponding bit of MAC value and the challenge bit. An attacker can realize distance fraud with the success probability $(3/4)^n$. Moreover, this protocol is vulnerable to terrorist and mafia frauds. The protocol is summarized in Figure 5.10.

Singelée and Preneel [114] proposed the idea of using the temporary secret key in the fast bit exchange phase. This idea was proposed by Bussard [115] to prevent terrorist fraud attack. They also presented another way to prevent terrorist attack by using trusted hardware. An attacker can not extract the secret key from the trusted hardware.

Reid et al. [116] modified the Hancke and Kuhn's protocol in [113] by taking advantage of the idea of using the secret key in the fast bit exchange phase. Mafia, distance and terrorist fraud can be realized with the success probability $(3/4)^n$. Piramuthu [117] presented a mafia fraud attack with the success probability $(7/8)^n$. Mitrokotsa et al. [118] showed the success probability of the mafia attack in [117] is $(3/4)^n$. While the noise increases the success probability decreases.

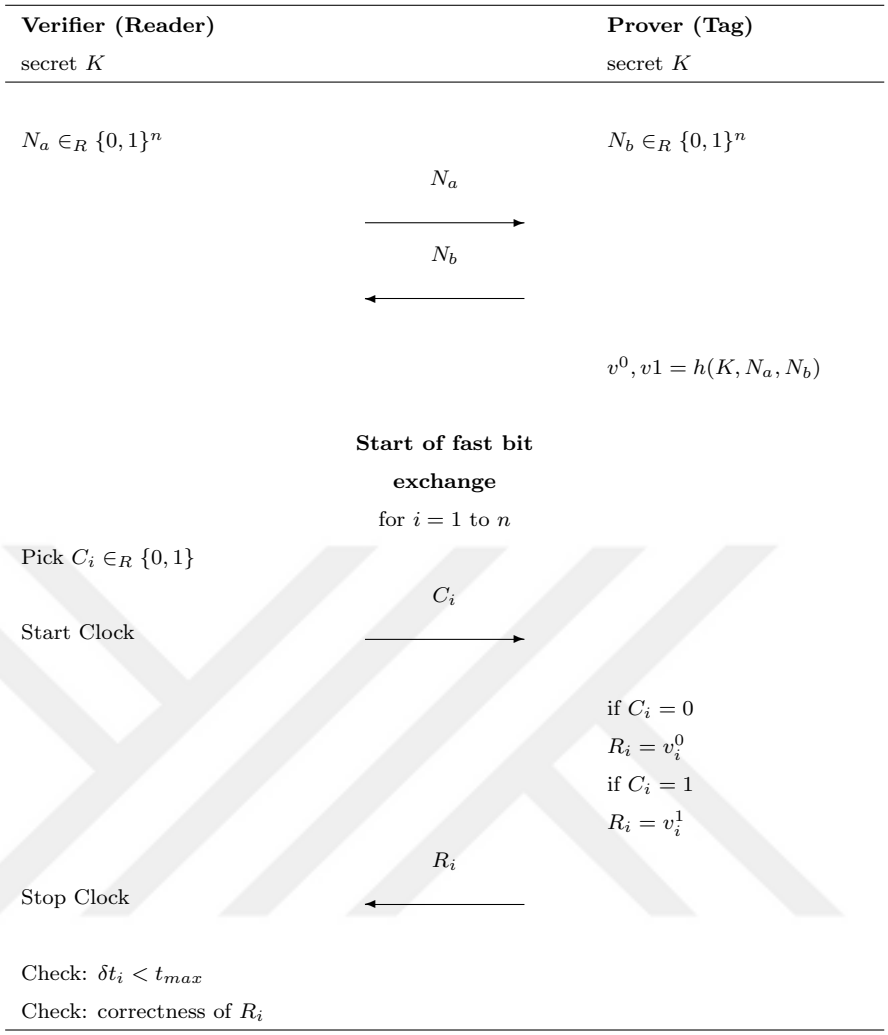

Figure 5.10. Hancke and Kuhn's protocol

Singelée and Preneel [119] used the noise-resilience property in [113] to modify Capkun's protocol in [112]. Relay attacks with the higher success probability were presented to this protocol in [120].

Tu and Piramithu [121] used the method in [116] to provide resistance against terrorist attack. In this protocol, there are four sections in fast bit exchange phase. The verifier sends a MAC value to the prover for authentication at the end of each section. While the number of sections in rapid bit exchange phase increases the success probability of attacker decreases. A key recovery attack for this protocol was presented in [122].

In 2008, Munilla and Peinado [123] proposed the idea of using void challenges to increase the security level aganist mafia fraud. The mafia attacker can not form the MAC value without knowing locations of void challenges. The success probability of the mafia attacker is calculated as $(3/5)^n$. Kim et al [122, 124] state that implementing three states physically (void, 0 and 1) is very difficult.

Kim et al. [122] made some changes on the protocol in [116]. These changes are that: the prover computes MAC value with the nonce generated by itself, both parties send their IDs in hidden format and at the end of the fast bit exchange phase, signatures are computed and sent to another party. The prover considers late responses, differences between challenges received and challenges sent and differences between given responses and expected responses to calculate an error value. If the error value is greater than the threshold value, the prover terminates the authentication session. The success probability of the distance, mafia and terrorist fraud are (3/4), (1/2) and (3/4) respectively. Lopez et al. [125] presented a disclosure attack in which an adversary can extract the prover's secret key.

In [124], Kim and Avoine increased the security level of the protocol in [113] against mafia fraud by using mixed challenges, random challenges and predefined challenges. The prover and verifier know predefined challenges so the content of MAC value is not disclosed by the mafia attacker. An attacker can realize mafia fraud, distance fraud and terrorist fraud with the success probabilities $(1/2)^n$, $(7/8)^n$ and 1.

Rasua et al. [126] recomputed mafia and distance fraud success probability of RFID distance bounding protocols proposed in [127] and [124]. The authors introduced the graph based concept and proposed an RFID distance bounding protocol based on a particular graph. Their protocol gives the best security protection against mafia and distance fraud attacks.

Peris-Lopez et al. [128, 129] proposed an RFID protocol that uses Weakly Secret Bit Commitment (WSBC). Weakly Secret Bit Commitment has the problem of key delegation. Their protocol needs the solution of cryptographic puzzles. They solved

the key delegation problem by merging distance bounding protocols and cryptographic puzzles.

Gürel et al. [130] presented the idea of traversing registers with non-uniform steps for producing responses. This protocol uses challenges and the secret key to traverse the register with non-uniform steps. The success probability of the mafia, terrorist and distance attacker is converged to $(1/2)^n$ without using the final signature. In [131], a distance fraud attack to this protocol is presented. The authors show that the success probability of an attacker could be $(3/4)^n$, if the distance between the attacker and the reader is very short.

Avoine et al. proposed a framework to design or analyse RFID distance bounding protocols [132]. They analysed and distinguished the terminology in the distance bounding domain. They defined a generic model for the adversary by considering capabilities and strategies of the adversary. They show that some equivalences exist between distance, mafia and terrorist frauds by considering the white-box and black-box models. Therefore, their analysis reduces the number of cases that should be considered when designing a new protocol. The authors also analysed the distance bounding protocol in [123] by using their framework and show that it has lower security level.

Kara et al. [133] introduced the notion of k-previous challenge dependent (k-PCD) protocol in which the current and k-previous challenges are used to compute the current response bit. They also defined the notion of current challenge dependent (CCD) protocol that is the special case $k = 0$ the current response bit is computed by using the current challenge bit. They showed that security levels of both protocol against distance and mafia fraud attacks are interrelated by presenting two attack scenarios. Furthermore, they tried to decrease the success probabilities of distance and mafia attacker by making some improvements.

Kardaş et al. [134] proposed two PUF based distance bounding protocol. They utilized PUFs as a secure storage to increase the security level of their protocol against all frauds. Security level of the first protocol against distance, mafia and terrorist

frauds is $(3/4)^n$ without final signature. The second protocol that uses final signature increases the security level to $(1/2)^n$.

Avoine et al. [135] showed that using a secret-sharing scheme based on threshold cryptography prevent terrorist fraud. They modified the distance bounding protocol in [113] and yielded two variants: the threshold distance bounding (tdb) protocol and the thrifty threshold distance bounding (ttdb) protocol. They defined three classes of adversaries to analyse security level of protocols. Their results show that, at least, a (3, 3) threshold scheme should be used to resist to terrorist fraud with powerful adversaries.

5.8. Lightweight Protocols

Low-cost RFID tags have 250-4K gates for security processing [12]. This makes them very limited in terms of computation because it is difficult to implement hashing functions with only 250-4000 gates. RFID authentication protocols that run on low-cost RFID tags and do not need hashing function are named as "Lightweight RFID Authentication Protocol". The protocols in [136–138] are considered as lightweight RFID authentication protocols. HB-based protocols [139–141] are also lightweight protocols because they use only dot products of binary vectors and a random noise bit and do not need hashing functions.

5.8.1. Protocols Conforming to EPC Class 1 Generation 2 Standards

EPCglobal Class 1 Generation 2 standards was adopted as 18000-6 international Standard in 2004 and published by ISO/IEC in 2006. RFID tag has the following properties according to EPCC1G2 standards [142]:

(i) Tags are passive.
(ii) Tags operate on the UHF band (860-960 MHz). Their communication range is 2-10m.

(iii) Tags have a 16-bit Pseudo-Random Number Generator (PRNG) and a 16-bit Cyclic Redundancy Code (CRC).

(iv) Tags have two 32-bit PINs. One of them is used to kill the tag and the other is used to read or write in the password fields.

A synchronization-based communication protocol for the EPCC1G2 tag was proposed in [137]. The proposed solution provides some level of security by using PRNG and CRC. It can not resist DoS attack and key guessing attack. Chien et al. [138] showed that a strong attacker that is able to corrupt tags can break forward security.

Chien et al. [138] proposed a mutual authentication protocol for RFID systems. Their protocol uses challenge-response methodology to prevent replay attacks. There are two authentication and access keys for each tag. These keys are used for authentication and resynchronization processes. If a tag authenticates the reader successfully, it updates its keys in order to provide forward security. Song and Mitchell [17] showed that a strong attacker knowing secrets can break forward and backward security. This protocol is also vulnerable to server impersonation attacks.

Burmester and Medeiros [143] analysed the protocols in [137, 138]. They showed that the protocol in [137] is vulnerable to tag impersonation attack resulting from the linearity of CRC-16 and the protocol in [138] is vulnerable to desynchronization attack and tag impersonation attack. Furthermore, they proposed three trivial RFID authentication protocols (TRAPs) conforming EPCGen2 standards. Yeh and Lo [144] presented a desynchronization attack for the TRAP-3 protocol.

5.8.2. Ultralightweight Protocols

In [136], a protocol based XOR operations and a simple matrix multiplication was proposed. This protocol does not need extensive cryptographic operations. It is very difficult to recover the multiplier from the multiplication of two matrices. This protocol uses this difficulty to provide some security level. In [138], DoS, replay and tracing attacks to this protocol were presented.

Peris et al. proposed first proposal M^{2}AP [145] of the family of ultralightweight mutual authentication protocols. The protocol is summarized in Figure 5.11. M^{2}AP is based on the use of index-pseudonym (IDS) which is index on table where all the information about a tag is stored. Each tag has a key $K = (K1||K2||K3||K4)$ where each part is 96 bits. Therefore, each tag needs rewritable memory for IDS and K that are 480 bits in sum and ROM memory for 96 bits static tag ID. Tags needs only 1000 logical gates to perform security operations: $\oplus, \wedge, \vee$ and sum mod(2^m). M^{2}AP has four stages: tag singulation in which the reader identifies the tag, mutual authentication, index-pseudonym updating and key updating. The authors claim that M^{2}AP have resistance to main security problems of RFID systems (privacy,tracking,etc.). In [146], how passive attacker can learn the identification number ID and secrets shared by the tag and the reader by eavesdropping some rounds of the protocol was shown.

Peris et al. proposed an another ultralightweight mutual authentication protocol EMAP [147]. The protocol has the same structure and assumptions with M^{2}AP. The differences are the way of generating exchanged messages and updating index-pseudonym and keys. In [148], the de-synchronization attack and the full-disclosure attack on EMAP were presented. Furthermore, Alomair et al. [149] presented an attack on EMAP and M^{2}AP in which a passive adversary can extract the tag's unique ID by eavesdropping a logarithmic number of protocol rounds.

LMAP that is modified version of the M^{2}AP protocol was proposed by Peris et al. [150]. They modified the index-pseudonym updating phase of the M^{2}AP protocol. In [151], the authors presented a passive attack against the protocol in which an attacker can learn the identification number ID and shared secrets by eavesdropping several rounds of protocol.

Li and Wang [152,153] presented the full-disclosure attacks and the de-synchronization attacks for the protocols M^{2}AP, EMAP and LMAP. An attacker wishing to perform the full disclosure attack must carry out the de-synchronization attack as a priority. The attacker also needs several sessions of the protocol to carry out the full disclosure attack. The authors also proposed an improved protocol that eliminates the vulnera-

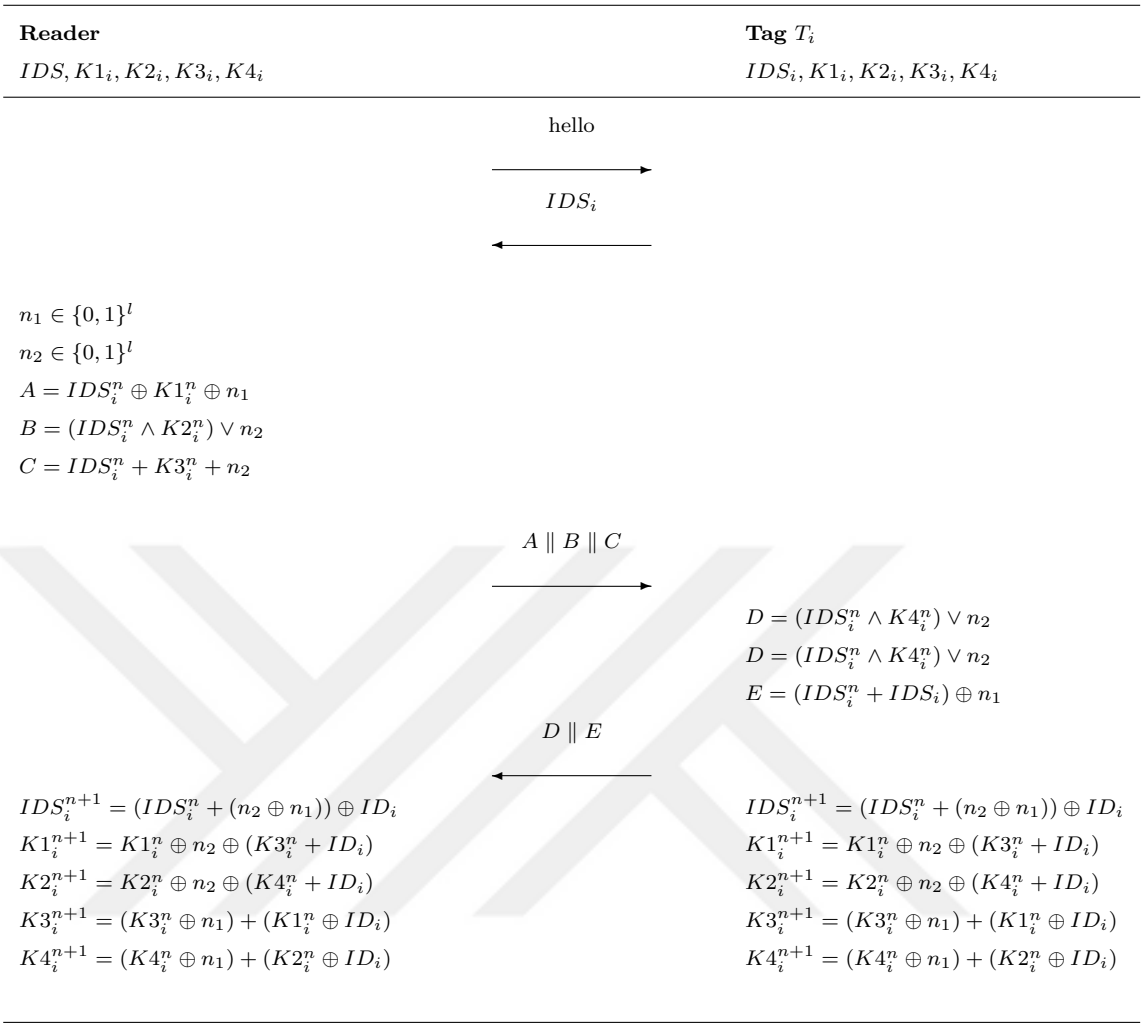

Figure 5.11. Peris et al's protocol - M^{2}AP

bilities of M^{2}AP, EMAP and LMAP. However, Chien and Huang [154] presented the de-synchronization attack and the full-disclosure attack for the improved protocol.

Chien [155] proposed an ultralightweight RFID mutual authentication protocol called as SASI. They claim that SASI enables strong authentication and integrity for RFID systems. The back-end server stores a pseudonym IDS and keys $K1, K2$ for each tag. The protocol resists de-synchronization attacks by keeping the previous and the current values of $(IDS, K1, K2)$. At the end of the each successful authentication session, synchronization values are confirmed to make the protocol robust to the possible de-synchronization attacks. Authors claim that their protocol can resist all possible attacks. However, D'Arco and De Santis [156] presented three attacks against SASI:

a de-synchronization attack, an identity disclosure attack and a full disclosure attack. In [157], the first passive attack to fully recover the secret ID of the RFID tag was proposed. The proposed attack is successful against variant of SASI that use modular rotations. In [158], Cao and Bertino presented two attacks against SASI: DoS attack in which a man-in-the-middle adversary breaks the synchronization between the reader and the tag, and tracing attack in which an adversary compromising a tag can trace the past communications and violate the backward untraceability. Phan [159] presented a passive attack in which a passive attacker can trace the tag with non-negligible probability. In [160], two de-synchronization attacks against SASI were presented.

In 2007, Li and Wang [161] proposed an ultralightweight RFID mutual authentication protocol called SLMAP. SLMAP uses only very efficient operations like bitwise XOR and modular addition. In SLMAP, random numbers are generated by the reader. Castro et al. [162] showed that SLMAP does not achieve untraceability. They presented black-box attack which is implemented by a non-standard cryptanalytic technique based on the use of a Simulated Annealing algorithm.

Peris et al. [163] proposed a new ultralightweight RFID mutual authentication protocol Gossamer that is inspired by SASI [155] so the design of Gossamer is very similar to SASI. In this proposal the authors use dual rotation and the MixBits function to make the protocol more secure. The authors claim that Gossamer is more secure than SASI because it eliminates the vulnerabilities of SASI. However, Bilal et al. [164] presented a de-synchronization attack on Gossamer. In this attack, an attacker replays eavesdropped messages to break key synchronization between reader and tag. The same attack is presented in [144] by Yeh and Lo.

Billet et al. [165] proposed a lightweight privacy preserving authentication protocol for RFID systems. It only uses a lightweight stream cipher which can be constructed on the tag. The authors introduce the notion of almost forward privacy by relaxing the unlinkability requirements in the definition of a forward private protocol. The authors claim that their protocol achieves both DoS-resistance and a very strong form of privacy which is close to the notion of forward privacy.

5.8.3. HB Based Protocols

Hopper and Blum [166] proposed the use of the Learning Parity with Noise (LPN) problem for secure authentication and identification for unassisted humans. This protocol is very lightweight because it requires random noise bits and dot products of binary vectors. It provides security by using the computational hardness of LNP. In the literature, this protocol is called as the HB protocol. The protocol is summarized in Figure 5.12.

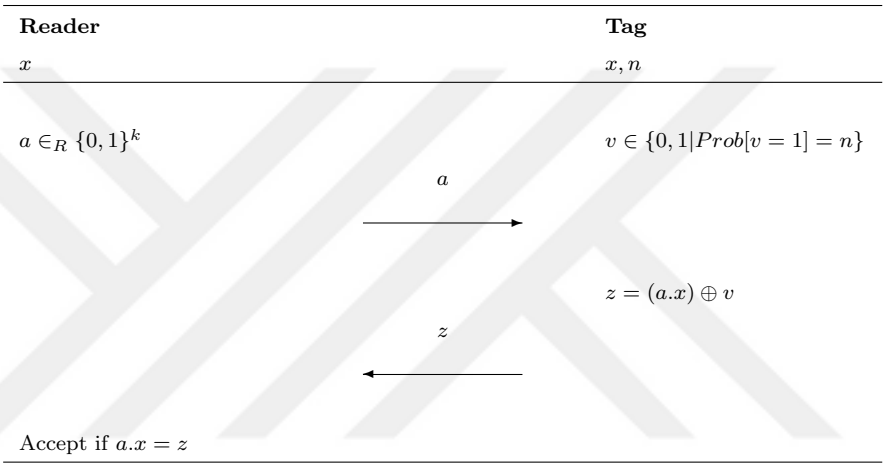

Figure 5.12. HB protocol round

Juels and Weis [139] investigated that RFID tags are similar to humans in terms of having limited capabilities. The protocol is summarized in Figure 5.13. They adopted HB protocol to the pervasive computing settings and proposed a new low-cost authentication protocol called HB$^+$ protocol. The HB$^+$ protocol was proved by considering active adversaries. An efficient man-in-the-middle attack with linear computational and communication complexity was presented in [167].

Bringer et al. [140] proposed the HB^{++} protocol that is modified version of the HB$^+$ protocol. The HB^{++} protocol removes the vulnerabilities of HB$^+$ against active attacks.

Piramuthu [141] analysed HB based RFID authentication protocols. He presented an attack to the HB^{++} protocol in which an attacker tries to identify secrets of a tag by

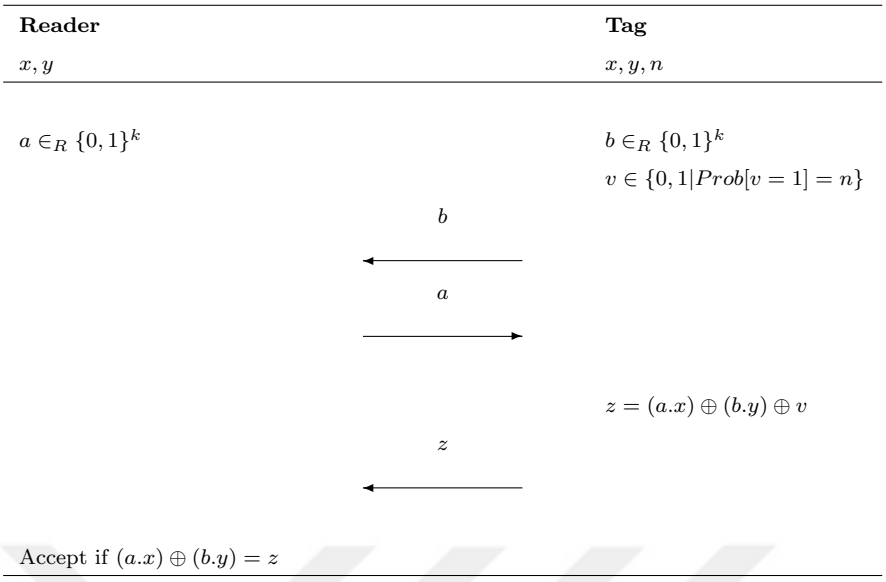

Figure 5.13. HB$^+$ protocol round

using man-in-the-middle methodology. He proposed some modifications to the HB^{++} protocol. These modifications does not provide security against all attack types.

Duc and Kim [168] proposed some modifications to the HB$^+$ protocol. The new protocol was called as HB*. Modifications on HB* removes the vulnerabilities of HB$^+$ presented by Gilbert et al. in [167]. In [169], Piramuthu showed that an attacker can learn some information by impersonating the reader to a tag and can track the tag by using this information.

Munilla and Peinado [170] derived a new HB based protocol from the protocol HB$^+$ and called it as HB-MP. Their first modification is to use of two messages instead of three messages. They also made some modifications to provide resistance to attack by Gilbert et al. in [167].

Gilbert et al. [171] proposed two HB based authentication protocols: random-HB$^\#$ and HB$^\#$. These protocols are the variants of HB$^+$. Random- HB$^\#$ is capable of the same calculation with HB$^+$ and has a single communication round. It is provably secure in the detection-based model and provides resistance to the attack in [167]. However, storage is the single drawback of random-HB$^\#$. Gilbert et al. solved the

storage problem of random-HB$^{\#}$ in the HB$^{\#}$ protocol.

Quafi et al. [172] presented a man-in-the-middle attack against HB$^{\#}$ and random-HB$^{\#}$. Their attack is also applicable to other HB based protocols. It recovers a shared secret in 2^{25} or 2^{20} authentication rounds for HB$^{\#}$ and 2^{34} or 2^{28} for random-HB$^{\#}$.

Hammori and Sunar [173] proposed a lightweight authentication protocol called PUF-HB. The PUF-HB protocol combines the strength of physically unclonable functions (PUFs) and the working principals of HB protocol to provide tamper resistant and secure authentication. The PUF-HB protocol has been proven against active attacks except man-in-the-middle attacks.

Leng et al. [174] improved the security of the HB-MP protocol against man-in-the-middle attacks. They called their new protocol HB-MP^{+}.

Table 5.2. Comparison of distance bounding protocols

Protocol	Distance Fraud	Mafia Fraud	Terrorist Fraud	Final Signature	Noise Resilience
Brands Chaum [111]	$(1/2)^n$	$(1/2)^n$	1	Yes	No
Capkun et al. [112]	$(1/2)^n$	$(1/2)^n$	1	Yes	No
Hancke Kuhn [113]	$(3/4)^n$	$(3/4)^n$	1	No	Yes
Reid et al. [116]	$(3/4)^n$	$(3/4)^n$	$(3/4)^n$	No	Yes
Singelée and Preneel [119]	$(1/2)^n$	$(1/2)^n$	1	Yes	Yes
Tu and Piramithu [121]	$(3/4)^n$	$(9/16)^n$	$(3/4)^n$	Yes	No
Munilla and Peinado [123]	$(3/4)^n$	$(3/5)^n$	1	No	Yes
Kim et al. [122]	$(3/4)^n$	$(1/2)^n$	$(3/4)^n$	Yes	Yes
Kim and Avoine [124]	$(7/8)^n$	$(1/2)^n$	1	No	Yes
Gürel et al. [130]	$> (3/4)^n$	$> (1/2)^n$	$> (1/2)^n$	No	Yes
Kardaş et al. [134]	$(1/2)^n$	$(1/2)^n$	$(1/2)^n$	Yes	No

Table 5.3. Comparison of protocols conforming to EPC class 1 generation 2 standards and ultralightweight protocols

Protocol	Information Privacy	Location Privacy	Tag Imp.	Replay Attack	DoS Attack	Backward Tracea.	Forward Tracea.	Server Imper.
Duc et al. [137]	○	○	–	–	–	–	–	–
Chien et al. [138]	○	○	–	–	–	–	–	–
KarthikeyanN-2005-sasnN-2005-sasn et al. [136]	○	–	○	–	–	–	–	–
Burmester et al. [143] TRAP-3*	○	○	○	○	–	○	–	–
Peris et al. [145] - M^2AP	–	–	–	–	–	–	–	–
Peris et al. [147] - EMAP	–	–	–	–	–	–	–	–
Peris et al. [150] - LMAP	–	–	–	–	–	–	–	–
Chien [155] SASI	–	–	–	–	–	–	–	–
Li and Wang [161] - SLMAP	○	–	○	○	○	–	–	–
Peris et al. [163] - Gossamer	○	–	○	–	–	–	–	–
Billet et al. [165]	○	○	○	○	○	○	–	–

* EPC Class 2 Generation 2 Compliant

6. NEW ATTACKS AND IMPROVEMENTS TO RECENT RFID PROTOCOLS

In this chapter, we exploit security flaws of some recently proposed RFID authentication protocols that have received no attacks yet. Each section involves a description of the target protocol and details of proposed attack.

6.1. New Attacks and Improvements to Gódor et. al's Protocol

In this section, we analyze the security of an RFID mutual authentication protocol called SLAP (Simplest Lightweight Authentication Protocol) [175] proposed by Gódor et al. SLAP meets the security requirements of small computational capacity RFID environments. Authors claim that their protocol provides data security and integrity of messages, mutual authentication, prevention of tracking, forward privacy, anonymity of a tag, undistinguishable messages from a random bit string, and defenses against the de-synchronization attack. Gódor and Imre [176] give a theoretical analysis of the SLAP protocol from the point of view of security and performance. They define the attacker model and give a proof of the correctness of the SLAP protocol using GNY logic [177]. We first show that the SLAP has no resistance to server impersonation attack introduced in [17]. The attacker can easily impersonate the valid back-end to a valid tag by only querying tag with special nonce. Furthermore as a result of the server impersonation attack, the attacker can break the synchronization between the tag and the back-end. Thus, the tag cannot be further authenticated by the back-end. We first show server impersonation attacks to SLAP. We propose our revised mutual authentication protocol that eliminates the vulnerabilities of the SLAP and provides more security with the same storage and computation requirements.

6.1.1. Protocol Description

In 2008, Godor et. al proposed a lightweight mutual authentication protocol for RFID systems [175]. They called their protocol SLAP (Simple Lightweight Authentication Protocol). SLAP needs only three hash computations on the tag side. In SLAP, the tag sends first log(Number of Tags) bits of its identifier to the back-end. $\log(x)$ implies $\log_2(x)$ in this context. The back-end does not compute a hash value for each entry in the database. The back-end computes a hash value for database entries whose first log(Number of Tags) bits of its identifier is equal to tag's response. Since the back-end performs database search fast, that is the reason SLAP is named lightweight.

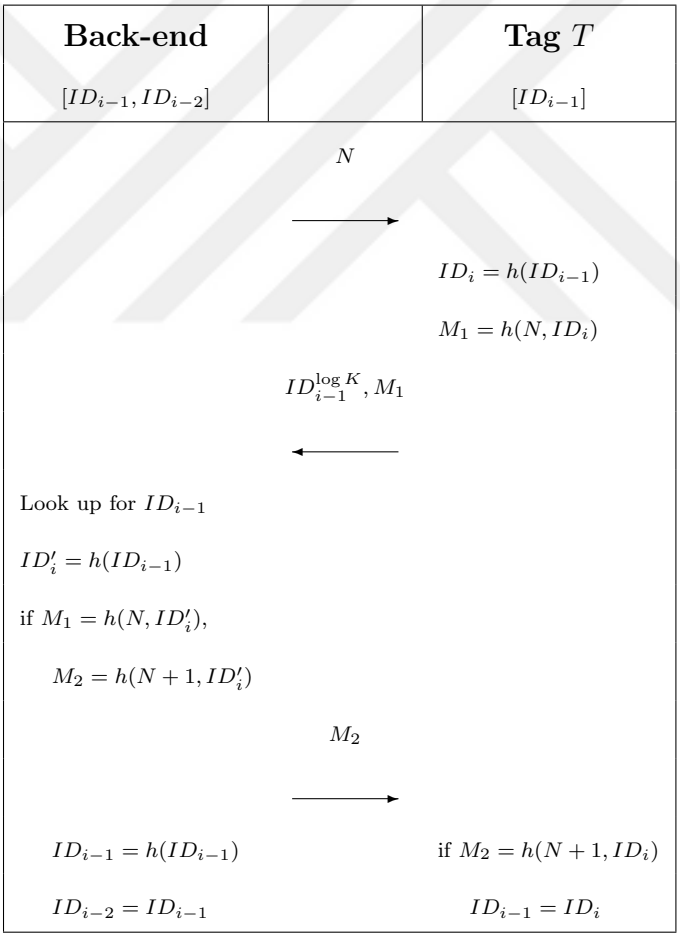

Figure 6.1. Gódor et al.'s protocol (SLAP)

We summarize SLAP protocol in Figure 6.1 and notations used in SLAP are listed in Table 6.1. For each tag T, the back-end server stores the current ID_{i-1} and previous ID_{i-2} identification number. A tag T only stores its current identification

Table 6.1. Notations of Gódor et. al's protocol

N	:	Nonce.
K	:	The number of tags.
$h()$	:	$\{0,1\}^l \times \{0,1\}^l \rightarrow \{0,1\}^l$ hash function
T	:	The tag.
i	:	The transaction identifier.
ID_{i-1}	:	The current identification number of T.
ID_{i-2}	:	The previous identification number of T.

number ID_{i-1}. In SLAP the initial transaction identifier is assumed to be $i = 1$. In initialization, ID_0 is randomly generated for every tag and the back-end server stores ID_0 and $ID_{-1} = 0$. A step by step description of the SLAP is given below:

(i) Back-end sends a random challenge N to the tag T.

(ii) After receiving nonce N, T computes i^{th} identification number as $ID_i = h(ID_{i-1})$. At this point, T does not write ID_i into its memory. Next, T computes the message $M_1 = h(N, ID_i)$ and send it to the back-end with the first $\log K$ bits of ID_{i-1} (current identification number).

(iii) After receiving T's reply, the back-end finds database records whose first $\log K$ bits equal to receiving $ID_{i-1}^{\log K}$. For each entry the back-end computes $ID_i = h(ID_{i-1})$ and compares M_1 with $h(N, ID_i)$. If they are equal, T is authenticated. The back-end updates the previous identification number as $ID_{i-2} = ID_{i-1}$ and current deification number as $ID_{i-1} = ID_i$.

(iv) The back-end computes $M_2 = h(N + 1, ID_i)$ and send it to the tag T.

(v) After receiving M_2, T computes its $h(N + 1, ID_i)$ and compares it with M_2. If they are equal, the back-end is authenticated and T updates its current identifier as $ID_{i-1} = ID_i$.

To the best of our knowledge, there are no attacks that were presented on SLAP. Lightweight authentication protocols are designed for resource constrained environ-

ments. They are intended to be at least secure against impersonation attacks. We will first present a server impersonation attack to SLAP in which adversary must prevent the reader from receiving a message. Furthermore, we will present another server impersonation attack to SLAP in which adversary only collects messages from one or more runs without interfering with the communication between the back-end and the tag.

6.1.2. Server Impersonation Attack I

We present that an adversary can impersonate a valid back-end in the SLAP mutual authentication protocol without using the internal state knowledge of the tag. This attack breaks synchronization of the secret information between tag and back-end. The details of this attack are given below.

- Phase I:
  - In a valid session denoted as $session_1$ between the back-end and the tag, the back-end sends a random challenge N to the tag. After receiving N, the tag computes $ID_i = h(ID_{i-1})$ and $M_1 = h(N, ID_i)$ and sends a reply $ID_{i-1}^{\log K}, M_1$ to the back-end. The adversary gets the values of N, M_1 and prevents the back-end from receiving $ID_{i-1}^{\log K}, M_1$.

- Phase II:
  - The adversary starts a new session denoted as $session_2$ with the tag. It sends $N' = N - 1$ as a challenge to the tag. After receiving N', the tag computes $ID_i = h(ID_{i-1})$ and $M_1' = h(N', ID_i)$ and sends a reply $ID_{i-1}^{\log K}, M_1'$ to the adversary.
  - After receiving $ID_{i-1}^{\log K}, M_1'$, the adversary sends $M_2' = M_1 = h(N' + 1, ID_i)$ to the tag. The tag will checks the validity of M_2' and accept it. As a result the tag updates its current identification number as $ID_{i-1} = ID_i$.

In the last step, the tag will accept M_2' and updates its current identification number. In $session_2$, the adversary queries the tag with a challenge N' and it needs a $h(N'+1, ID_i)$ value to impersonate the valid back-end. In $session_1$, the adversary eavesdrops a $M_1 = h(N, ID_i)$ value. Since $N'+1 = N$ it already knows $h(N'+1, ID_i) = h(N, ID_i) \Rightarrow M_2' = M_1$. Therefore the tag accepts M_2'. The adversary can repeat the above attack continuously.

6.1.3. Server Impersonation Attack II

In addition to the above attack, we present another server impersonation attack in which the adversary does not need to the legal session between the back-end and the tag and does not need to prevent the back-end from receiving the message M_2 but starts communication with the tag directly. The details of this attack are given below.

- Phase I:
  - The adversary starts a new session denoted as $session_1$ with the tag. It sends N as a challenge to the tag. After receiving N, the tag computes $ID_i = h(ID_{i-1})$ and $M_1 = h(N, ID_i)$ and sends a reply $ID_{i-1}^{\log K}, M_1$ to the adversary.

- Phase II:
  - The adversary starts a new session denoted as $session_2$ with the tag. It sends $N' = N - 1$ as a challenge to the tag. After receiving N', the tag computes $ID_i = h(ID_{i-1})$ and $M_1' = h(N', ID_i)$ and sends a reply $ID_{i-1}^{\log K}, M_1'$ to the adversary.
  - After receiving $ID_{i-1}^{\log K}, M_1'$, the adversary sends $M_2' = M_1 = h(N'+1, ID_i)$ to the tag. The tag will checks the validity of M_2' and accept it. As a result the tag updates its current identification number as $ID_{i-1} = ID_i$.

In the last step, the tag will accept M_2' and updates its current identification number. In $session_2$, the adversary queries the tag with a challenge $N' = N - 1$ and it needs a $h(N' + 1, ID_i)$ value to impersonate the valid back-end. In $session_1$, the adversary gets a $M_1 = h(N, ID_i)$ value from the tag. Since $N' + 1 = N$ the adversary has $h(N' + 1, ID_i) = h(N, ID_i) \Rightarrow M_2' = M_1$. Therefore, the tag accepts M_2'. The adversary can repeat the above attack continuously like the first attack.

6.1.4. Revised RFID Mutual Authentication Protocol

In SLAP, an adversary can get the messages that it needs to impersonate the valid back-end by querying the tag. An adversary can use these messages to impersonate the valid back-end and break the synchronization between the tag and the back-end. In this section, we propose a revised mutual authentication protocol that eliminates the vulnerabilities of the SLAP.

Our revised mutual authentication protocol is the same as SLAP except the way the message M_2 is created. In SLAP, M_2 is created by $h(N + 1, ID_i)$. In our revised mutual authentication protocol, we set M_2 to $h(ID_i, N+1)$. Implications of reordering M_2 contents to prevent the attack are detailed in Section 6.1.5.

We summarize our revised mutual authentication protocol in Figure 6.2 and we use the same notations with SLAP are listed in Table 6.1.

6.1.5. Resistance of Revised Protocol to Server Impersonation Attack

In SLAP, the tag uses the message M_2 to authenticate the back-end. The back-end can create the valid M_2 if it has N and the valid ID_i. The back-end can create the valid ID_i with the knowledge of ID_{i-1}. The weakness of SLAP is that an adversary can create the valid M_2 without knowledge of ID_{i-1} and ID_i. The message M_1 created by the tag and the message M_2 created by the back-end has same input structure (random nonce$\|ID_i$) and the back-end use $N + 1$ as random nonce so the adversary can get valid $M_2 = h(N + 1, ID_i)$ by querying the tag with $N + 1$.

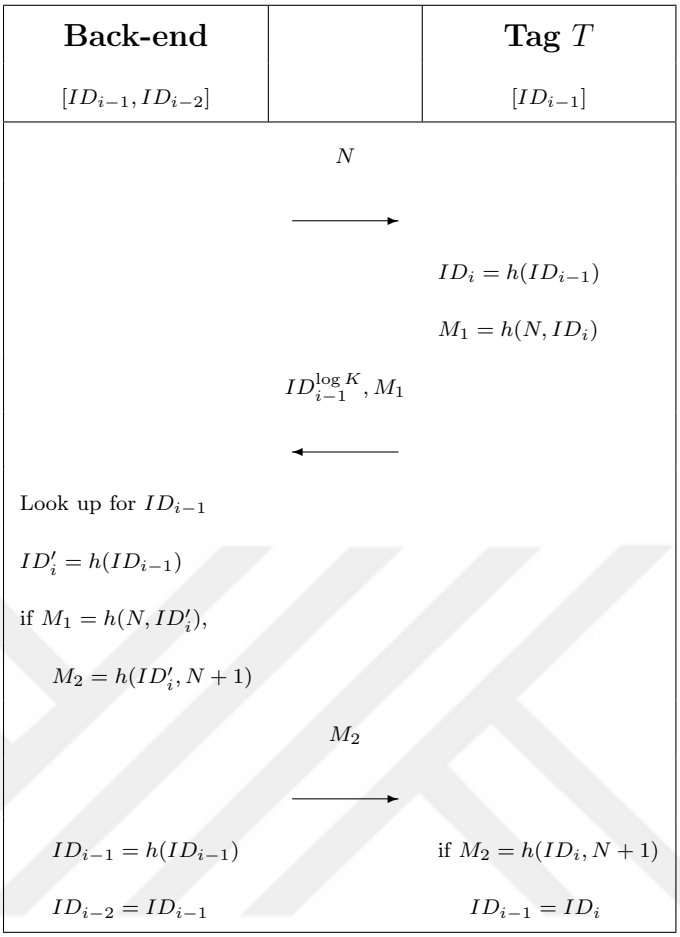

Figure 6.2. Revised SLAP protocol

In our revised mutual authentication protocol, we change the input structure of the message M_2. The back-end creates M_2 by computing $h(ID_i, N+1)$. Therefore an adversary can not query the tag to create the valid message $M_2 = h(ID_i, N+1)$ and can not create the valid M_2 without knowledge of ID_i. Thus, our revised mutual authentication protocol can prevent the server impersonation attack detailed in Section 6.1.2 and 6.1.3.

6.2. New Attacks to Gódor and Antal's Protocol

In this section, we analyze the security of an RFID lightweight mutual authentication protocol proposed by Gódor and Antal. Authors claim that this protocol provides data confidentiality, integrity, tag anonymity and untraceability of tags. They also explain that it prevents replay attacks, man-in-the-middle attacks as well as impersonating the parties. Furthermore, they give a theoretical analysis of their protocol

from the point of view of security and give a proof of the correctness of it using GNY logic [177]. However, some attacks can be handled to the GA protocol. We first show that the GA protocol is vulnerable to tag impersonation attack. This breaks the mutual authentication between the back-end and tags. Furthermore we show that the GA protocol has no resistance to server impersonation attack introduced in [17]. The attacker can easily impersonate the valid back-end to a valid tag by only querying tag with special value. Furthermore as a result of the server impersonation attack, the attacker can break the synchronization between the tag and the back-end. Thus, the tag cannot be further authenticated by the back-end.

6.2.1. Description of Luo et al.'s Protocol

In 2005, Luo et al. in [178] designed a challenge-response protocol based on Ohkubo et al.'s scheme in [23]. Their protocol is illustrated in Figure 6.3. At the beginning, each tag has an initial secret s_0 and a transaction counter c. On the other hand, the back-end database contains initial secret s_0 and workable region W_k('s) for each tag.

In this context, b is used for values calculated or stored in the back-end. s_i^b is equal to s_i stored in the tag side. The back-end calculates s_i^b by hashing s_0^b for c_i times. In the i^{th} transaction, the protocol steps are defined as follows:

(i) Upon receiving the reader's query, the tag sends c_i and $m_{1_i} = G(s_i \oplus c_i)$ to the reader.

(ii) For each tag, the back-end database calculates s_i^b by hashing s_0^b for c_i times and it also calculates $m_{1_i}^b = G(s_i^b \oplus c_i)$ to find the matching tag.

(iii) The back-end database generates a random session number R and sends $m_{2_i} = R \oplus G(s_i^b)$, $m_{3_i} = G(s_i^b \oplus R \oplus W_k)$ and W_k to the tag.

(iv) The tag extracts R^t from m_{2_i} and checks whether $m_{3_i} = G(s_i^b \oplus R \oplus W_k) = G(s_i \oplus R^t \oplus W_k)$ or not.

(v) The mutual authentication is completed and a common random session number, R is agreed. The authorization of W_k is also achieved.

(vi) The tag computes $G(R) \oplus data$ and sends it to the reader. For integrity protection, $G(data)$ is also transferred.

(vii) The tag increments the transaction counter, c_i, by one and renews secret $s_{i+1} = H(s_i)$.

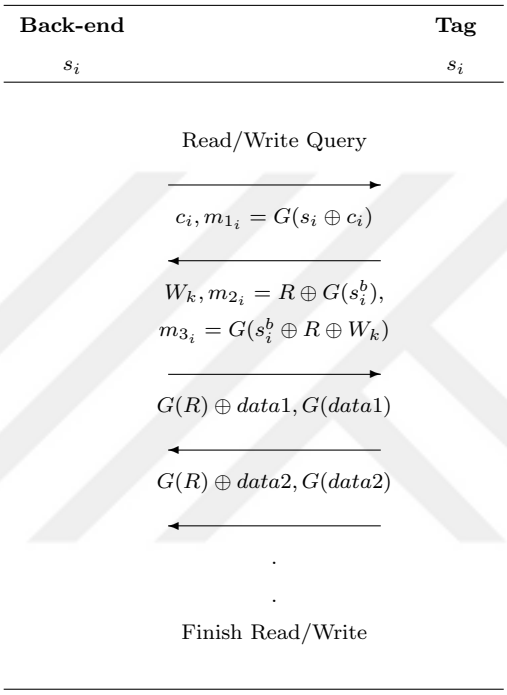

Figure 6.3. Luo et al.'s protocol

In 2008, Gódor and Antal in [179] claimed that the LCL protocol in [178] has two drawbacks:

- The back-end database needs to calculate hash c_i times for every tag which causes a heavy computation overhead. To obtain a better performance, the number of hash operations should be decreased.
- An adversary can easily replay the second message to impersonate the tag. This problem stems from the fact that the back-end database does not use any challenge to ensure the freshness of this message.

6.2.2. Description of Gódor and Antal's Protocol

In 2008, Godor et al. proposed a lightweight mutual authentication protocol for RFID systems [179]. Their protocol is the improved version of the protocol in [178]. The protocol is illustrated in Figure 6.4 and notations are listed in Table 6.2. This protocol is different than the LCL protocol in [178] in two ways:

- At the beginning of every transaction, a challenge R_0 is produced and sent by the back-end database. The tag uses this challenge to compute the hash value, thus freshness of the tag's response could be verified.
- A simple iterated hash chain is used in every transaction. That is, s_i, the secret value for the present transaction, is derived by hashing s_{i-1}, the secret value for the previous transaction only. In the GA protocol the back-end stores the secret value for the previous transaction but in the LCL protocol, the back-end stores the initial secret value. Thus, the number of hash operations compared to the LCL protocol in [178] is decreased significantly.

The rest of the protocol steps are same as those of the LCL protocol in [178].

Table 6.2. Notations of Gódor and Antal's protocol

i	:	Transaction identifier.
s_i	:	The tag secret.
c_i	:	Transaction counter.
R	:	Common random session number.
R_0	:	Random Challenge.
W_k	:	Workable region.
$H()$	:	$\{0,1\}^l \rightarrow \{0,1\}^l$ hash function
$G()$	:	$\{0,1\}^l \rightarrow \{0,1\}^l$ hash function

The authors claim that this protocol provides data confidentiality, integrity, tag anonymity and untraceability of tags. They also explain that it prevents replay attacks, man-in-the-middle attacks as well as impersonating the parties. However, as will be

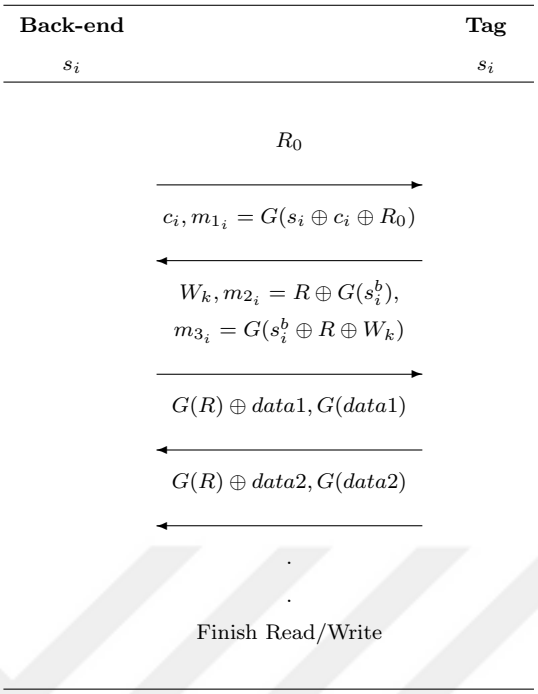

Figure 6.4. Gódor and Antal's protocol

explained in the next section, some attacks can be handled to eliminate some of these properties.

To the best of our knowledge, there are no attacks that were presented on GA protocol. RFID authentication protocols designed for resource constrained environments have to be at least secure against impersonation attacks. We show that an adversary can impersonate a tag to the server and the server to a tag. In our attacks, we assume that an adversary can control the network by blocking, modifying and injecting messages in any communication between a reader and a tag.

6.2.3. Tag Impersonation Attack

We present an attack in which an adversary can impersonate a tag to a legitimate server. In this attack, the adversary has no knowledge about the internal state of the tag. This attack breaks synchronization of the secret information between tag and the back-end. The details of this attack are given below:

- Phase I:
  - The adversary sends a random challenge R_0 to the tag. After receiving R_0, the tag computes $m_{1_i} = G(s_i \oplus c_i \oplus R_0)$ and sends a reply c_i, m_{1_i} to the adversary.

- Phase II:
  - The valid back-end queries tags with a random challenge R'_0.
  - After receiving R'_0, the adversary sends c'_i, m_{1_i} to the back-end, where $c'_i = c_i \oplus R_0 \oplus R'_0$.

Now, we show that c'_i, m_{1_i} is a valid reply for the tag. Since $c'_i = c_i \oplus R_0 \oplus R'_0$, the back-end calculates $G(s_i \oplus c_i \oplus R_0 \oplus R'_0 \oplus R'_0) = G(s_i \oplus c_i \oplus R_0) = m_{1_i}$. Therefore, the adversary fools the back-end by impersonating the valid tag. In the LCL protocol, the back-end database calculates hash c_i times for every tag to obtain s_i^b. In the GA protocol, c_i value stored by tags is only used as input to $G()$ function and the back-end does not store c_i values related to tags. The adversary can use c_i value to generate correct replies to the back-end queries. At the end of the attack, the adversary can not send some valid data to the back-end server because the adversary does not know the common random session number R.

6.2.4. Server Impersonation Attack

We present an attack in which an adversary can impersonate a legitimate server to a tag. In this attack, the adversary has no knowledge about the internal state of the tag. The details of this attack are given below:

- Phase I:
  - In a valid session between the back-end and the tag, the back-end sends a random challenge R_0 to the tag.
  - After receiving R_0, the tag computes $m_{1_i} = G(s_i \oplus c_i \oplus R_0)$ and sends a reply c_i, m_{1_i} to the back-end.

  - After receiving the tag's reply, the back-end creates a random number R, computes $s_i^b = H(s_{i-1}^b)$, $m_{2_i} = R \oplus G(s_i^b)$ and $m_{3_i} = G(s_i^b \oplus R \oplus W_k)$. and sends m_{2_i}, m_{3_i}, W_k to the tag.
  - The adversary gets the values of c_i, W_k.

- Phase II:
  - The adversary starts a new session with the tag. It sends $R'_0 = c_i + 1$ as a challenge to the tag.
  - After receiving R'_0, the tag computes $m'_{1_i} = G(s_i \oplus (c_i + 1) \oplus R'_0)$ and sends a reply c_i, m'_{1_i} to the adversary.
  - After receiving c_i, m'_{1_i}, the adversary creates a number $R' = W_k$, computes $m'_{2_i} = R' \oplus m'_{1_i}$ and $m'_{3_i} = m'_{1_i}$ and sends m'_{2_i}, m'_{3_i}, W_k to the tag.
  - The tag learns the number R' from m'_{2_i} and checks the validity of m'_{3_i}.

Now, we show that the tag will accept m'_{2_i}, m'_{3_i}, W_k to authenticate the back-end. In the second round, the tag computes $m'_{1_i} = G(s_i \oplus (c_i + 1) \oplus R'_0)$ where $R'_0 = c_i + 1$. Therefore, $m'_{1_i} = G(s_i \oplus (c_i + 1) \oplus (c_i + 1)) = G(s_i)$, $m'_{2_i} = R' \oplus m'_{1_i} = R' \oplus G(s_i = s_i^b)$ and $m'_{3_i} = m'_{1_i} = G(s_i = s_i^b)$. The tag learns the number R'' from $m'_{2_i} = R' \oplus G(s_i)$ and then computes $m''_{3_i} = G(s_i = s_i^b \oplus R'' = W_k \oplus W_k) = G(s_i = s_i^b) = m'_{1_i}$. We can see that the adversary passes the check in the tag side. After checking the validity of messages, the tag sends data stored in workable regions to the adversary. At the end of data transfer, the tag updates its secret to $s_{i+1} = H(s_i)$, which is different from the current secret s_i stored in the back-end. Therefore, the back-end has no knowledge on the updated tag secret after this attack and thus the server will not able to identify or authenticate the tag in the future sessions. At the end of the attack, the adversary can decrypt data that is sent to her by the victim tag because the adversary knows the common random session number R.

The GA protocol is vulnerable to the tag impersonation attack and the server impersonation attack like the protocol in [17]. Cai et al. [60] show the vulnerabilities of the protocol in [17]. Vulnerabilities of the GA protocol stem from the use of $\oplus$ operation. Another weak point is the use of transaction counter c_i which is incremented

by 1 after successful authentication as an input to $G()$ function. An adversary can use these weak points to impersonate a valid tag to a back-end and impersonate a back-end to a valid tag.

6.3. New Attacks and Improvements to Gao et. al's Protocol

Gao et al. [180] proposed an ultra lightweight RFID authentication protocol that utilizes CRC-16 and permutation (LPCP) functions. The authors formally verify the security of their protocol by using Simple Promela Interpreter (SPIN). They claim that their protocol provides resistance to the following attacks: de-synchronization attacks, tracing attacks, replay attack and secret disclosure attack.

6.3.1. Protocol Description

In Gao et al.'s protocol, each tag $\mathcal{T}_i$ is assigned with four parameters (TID_i, $KeyH_i$, $KeyL_i$, $KeyM_i$). The server stores two entry for the tag $\mathcal{T}_i$: (TID_i^{old}, $KeyH_i^{old}$, $KeyL_i^{old}$, $KeyM_i^{old}$) and (TID_i^{new}, $KeyH_i^{old}$, $KeyL_i^{old}$, $KeyM_i^{old}$). Parameters represented with $_{new}$ substring are the current secrets of $\mathcal{T}_i$ and parameters represented with $_{old}$ substring are the last successfully verified secrets of $\mathcal{T}_i$. At the system initialization, these two entry equals each other. Table 6.3 gives the notations used in describing the proposed protocol. The details of the protocol are given below:

(i) The reader $\mathcal{R}$ sends a $Hello$ message to the tag $\mathcal{T}_i$.

(ii) Upon receiving the $Hello$ message, $\mathcal{T}_i$ sends TID_i to $\mathcal{R}$.

(iii) $\mathcal{R}$ gets the entry in the index $\mathcal{T}_i$ and generates a random number R_1. $\mathcal{R}$ computes α and β and sends them to $\mathcal{T}_i$.

(iv) $\mathcal{T}_i$ extracts R_1 from α and checks the validity of β in order to authenticate $\mathcal{R}$. If it is valid, $\mathcal{T}_i$ computes γ and sends it to $\mathcal{R}$.

(v) $\mathcal{R}$ checks the validity of γ in order to authenticate $\mathcal{T}_i$. If $\mathcal{T}_i$ is authenticated, $\mathcal{R}$ generates a random number R_2 and computes δ and ζ. Then δ and ζ are sent to $\mathcal{T}_i$. If $TID_i = TID_i^{new}$, $\mathcal{R}$ updates the old secrets of $\mathcal{T}_i$ otherwise old secrets of $\mathcal{T}_i$ remains unchanged. Then $\mathcal{R}$ updates the new secrets of $\mathcal{T}_i$.

Table 6.3. Notations of Gao et al.'s protocol

Notation	Description
TID_i	The secure identity of the tag $\mathcal{T}_i$
$KeyX_i$	The secret keys of the tag $\mathcal{T}_i$
x_{new}	The current value of x
x_{old}	The previous value of x
CRC	The cyclic redundancy check operation
Per	The permutation operation
$\oplus$	The bit-wise XOR operation
$\in$	Random choice operator
$\leftarrow$	The substitution operation

(vi) Upon receiving δ and ζ, $\mathcal{T}_i$ extracts R_2 from β and checks the validity of ζ. If it is valid, $\mathcal{T}_i$ updates it secrets.

6.3.2. De-synchronization Attack

An adversary $\mathcal{A}$ performs the following attack in order to de-synchronize the secrets shared between $\mathcal{R}$ and $\mathcal{T}_i$. For simplicity, we assume that the attack begins after the last successful authentication session $s-1$. At the beginning of the attack, the states of the reader $\mathcal{R}$ and the tag $\mathcal{T}_i$ are shown in Table 6.4.

Table 6.4. De-synchronization attack on Gao et al.'s protocol: state of $\mathcal{R}$ and $\mathcal{T}_i$ at the beginning of the attack

Reader $\mathcal{R}_{new}$	$[TID_i, KeyH_i, KeyL_i, KeyM_i]^s$
Reader $\mathcal{R}_{old}$	$[TID_i, KeyH_i, KeyL_i, KeyM_i]^{s-1}$
Tag $\mathcal{T}_i$	$[TID_i, KeyH_i, KeyL_i, KeyM_i]^s$

(i) In a protocol session s between $\mathcal{R}$ and $\mathcal{T}_i$, $\mathcal{A}$ prevents $\mathcal{T}_i$ from taking last message flow and eavesdrops α^s, β^s, δ^s and ζ^s.

Reader		**Tag** T_i
$[TID_i^{old}, KeyH_i^{old}, KeyL_i^{old}, KeyM_i^{old}]$		$[TID_i, KeyH_i, KeyL_i, KeyM_i]$
$[TID_i^{new}, KeyH_i^{new}, KeyL_i^{new}, KeyM_i^{new}]$		
	$\xrightarrow{Hello}$	
	$\xleftarrow{TID_i}$	
$R_1 \in \{0,1\}^l$ $\alpha = CRC(Per(KeyM_i, KeyH_i)) \oplus R_1$ $\beta = CRC(Per(KeyM_i \oplus$ $KeyH_i, CRC(KeyM_i \oplus R_1)) \oplus$ $Per(KeyL_i, CRC(KeyM_i \oplus R_1)))$		
	$\xrightarrow{\alpha, \beta}$	
		$\gamma =$ $CRC(Per(CRC(KeyH_i \oplus R_1), CRC(R_1 \oplus$ $KeyM_i)) \oplus Per(CRC(KeyM_i \oplus$ $KeyL_i), CRC(R_1 \oplus KeyL_i)))$
	$\xleftarrow{\gamma}$	
$R_2 \in \{0,1\}^l$ $\delta = CRC(Per(KeyM_i, KeyL_i)) \oplus R_2$ $\zeta =$ $CRC(Per(CRC(KeyH_i \oplus R_1), CRC(R_2 \oplus$ $KeyM_i)) \oplus Per(CRC(KeyM_i \oplus$ $R_2), CRC(R_1 \oplus KeyL_i)))$		
	$\xrightarrow{\delta, \zeta}$	
if TID_i^{new} is received $TID_i^{old} = TID_i^{new}$, $KeyM_i^{old} = KeyM_i^{new}$ $KeyH_i^{old} = KeyH_i^{new}$, $KeyL_i^{old} = KeyL_i^{new}$ $TID_i^{new} = CRC(Per(TID_i^{new}, R1 \oplus$ $R2) \oplus KeyH_i^{old} \oplus KeyM_i^{old} \oplus KeyL_i^{old})$ $KeyH_i^{new} =$ $CRC(Per(KeyH_i^{old}, R_1) \oplus KeyM_i^{old})$ $KeyM_i^{new} =$ $CRC(Per(KeyM_i^{old}, R_2) \oplus KeyH_i^{old})$ $KeyL_i^{new} =$ $CRC(Per(KeyL_i^{old}, R_1 \oplus R_2) \oplus TID_i^{old})$		$TID_i = CRC(Per(TID_i, R1 \oplus R2) \oplus$ $KeyH_i \oplus KeyM_i \oplus KeyL_i)$ $KeyH_i = CRC(Per(KeyH_i, R_1) \oplus KeyM_i)$ $KeyM_i =$ $CRC(Per(KeyM_i, R_2) \oplus KeyH_i)$ $KeyL_i =$ $CRC(Per(KeyL_i, R_1 \oplus R_2) \oplus TID_i)$

Figure 6.5. Gao et al.'s protocol

(ii) At the end of the protocol session s, $\mathcal{R}$ updated the secrets related with $\mathcal{T}_i$. However, $\mathcal{T}_i$ did not update its secrets because it did not receive the last message flow. The states of the reader $\mathcal{R}$ and the tag $\mathcal{T}_i$ are shown in Table 6.5.

Table 6.5. De-synchronization attack on Gao et al.'s protocol: state of $\mathcal{R}$ and $\mathcal{T}_i$ at the end of the session s

Reader $\mathcal{R}_{new}$	$[TID_i, KeyH_i, KeyL_i, KeyM_i]^{s+1}$
Reader $\mathcal{R}_{old}$	$[TID_i, KeyH_i, KeyL_i, KeyM_i]^{s}$
Tag $\mathcal{T}_i$	$[TID_i, KeyH_i, KeyL_i, KeyM_i]^{s}$

(iii) In a protocol session $s+1$ between $\mathcal{R}$ and $\mathcal{T}_i$, $\mathcal{A}$ prevents $\mathcal{T}_i$ from taking last message flow.

(iv) At the end of the protocol session $s+1$, $\mathcal{R}$ updated the secrets related with $\mathcal{T}_i$. However, $\mathcal{T}_i$ did not update its secrets because it did not receive the last message flow. The states of the reader $\mathcal{R}$ and the tag $\mathcal{T}_i$ are shown in Table 6.6.

Table 6.6. De-synchronization attack on Gao et al.'s protocol: state of $\mathcal{R}$ and $\mathcal{T}_i$ at the end of the session $s+1$

Reader $\mathcal{R}_{new}$	$[TID_i, KeyH_i, KeyL_i, KeyM_i]^{s+2}$
Reader $\mathcal{R}_{old}$	$[TID_i, KeyH_i, KeyL_i, KeyM_i]^{s}$
Tag $\mathcal{T}_i$	$[TID_i, KeyH_i, KeyL_i, KeyM_i]^{s}$

(v) $\mathcal{A}$ broadcasts a *Hello* message and $\mathcal{T}_i$ sends its TID_i^s to $\mathcal{A}$.

(vi) $\mathcal{A}$ sends α^s and β^s to $\mathcal{T}_i$. $\mathcal{A}$ passes the check by the tag $\mathcal{T}_i$ because α^s and β^s are generated with the current secrets of $\mathcal{T}_i$ and they are not generated with random values created by $\mathcal{T}_i$. Therefore, $\mathcal{T}_i$ sends γ^s to $\mathcal{A}$.

(vii) $\mathcal{A}$ sends δ^s and ζ^s to $\mathcal{T}_i$. $\mathcal{A}$ passes the check by the tag $\mathcal{T}_i$ because δ^s and ζ^s are generated with the current secrets of $\mathcal{T}_i$ and they are not generated with random values created by $\mathcal{T}_i$. $\mathcal{T}_i$ updates its secrets by using TID_i^s, $KeyH_i^s$, $KeyL_i^s$, $KeyM_i^s$, R_1^s and R_2^s values from the session s. The states of the reader $\mathcal{R}$ and the tag $\mathcal{T}_i$ are shown in Table 6.7. In the next session, $\mathcal{R}$ will not be able to authenticate the tag.

Table 6.7. De-synchronization attack on Gao et al.'s protocol: state of $\mathcal{R}$ and $\mathcal{T}_i$ at the end of the attack

Reader $\mathcal{R}_{new}$	$[TID_i, KeyH_i, KeyL_i, KeyM_i]^{s+2}$
Reader $\mathcal{R}_{old}$	$[TID_i, KeyH_i, KeyL_i, KeyM_i]^{s}$
Tag $\mathcal{T}_i$	$[TID_i, KeyH_i, KeyL_i, KeyM_i]^{s+1}$

In the above, we show that the adversary $\mathcal{A}$ breaks synchronization between $\mathcal{R}$ and $\mathcal{T}_i$. $\mathcal{A}$ impersonates the valid reader by using the messages eavesdropped in the session s and makes the tag to update its keys. At the end of the attack, the secret values at server side and the tag side are shown in Table 6.7.

6.3.3. Countermeasure

To defend against de-synchronization, we recommend the following modifications. The tag does not generate a random number after receiving messages α and β. The tag should generate a random number $R_3 \in 0, 1^l$ and use R_3 in the computation of γ and κ as described below.

$$\begin{aligned} \kappa &= CRC(Per(KeyL_i, KeyH_i)) \oplus R_3 \\ \gamma &= CRC(Per(CRC(KeyH_i \oplus R_1), CRC(R_1 \oplus KeyM_i)) \\ &\quad \oplus Per(CRC(KeyM_i \oplus KeyL_i), CRC(R_3 \oplus KeyL_i))) \end{aligned} \tag{6.1}$$

The reader extracts R_3 from κ and checks the validity of γ in order to authenticate the tag. The reader should use R_3 in the calculation of ζ as described below.

$$\begin{aligned} \zeta &= CRC(Per(CRC(KeyH_i \oplus R_1), CRC(R_2 \oplus KeyM_i)) \\ &\quad \oplus Per(CRC(KeyM_i \oplus R_3), CRC(R_1 \oplus KeyL_i))) \end{aligned} \tag{6.2}$$

Let's consider the following attack scenario.

(i) In a protocol session s between $\mathcal{R}$ and $\mathcal{T}_i$, $\mathcal{A}$ prevents $\mathcal{T}_i$ from taking last message flow and eavesdrops α^s, β^s, δ^s and ζ^s.

(ii) At the end of the protocol session s, $\mathcal{R}$ updated the secrets related with $\mathcal{T}_i$. However, $\mathcal{T}_i$ did not update its secrets because it did not receive the last message flow. The states of the reader $\mathcal{R}$ and the tag $\mathcal{T}_i$ are shown in Table 6.5.

(iii) In a protocol session $s+1$ between $\mathcal{R}$ and $\mathcal{T}_i$, $\mathcal{A}$ prevents $\mathcal{T}_i$ from taking last message flow.

(iv) At the end of the protocol session $s+1$, $\mathcal{R}$ updated the secrets related with $\mathcal{T}_i$. However, $\mathcal{T}_i$ did not update its secrets because it did not receive the last message flow. The states of the reader $\mathcal{R}$ and the tag $\mathcal{T}_i$ are shown in Table 6.6.

(v) $\mathcal{A}$ broadcasts a $Hello$ message and $\mathcal{T}_i$ sends its TID_i^s to $\mathcal{A}$.

(vi) $\mathcal{A}$ sends $\alpha^{s+2} = \alpha^s$ and $\beta^{s+2} = \beta^s$ to $\mathcal{T}_i$. $\mathcal{A}$ passes the check by the tag $\mathcal{T}_i$ because α^s and β^s are generated with the current secrets of $\mathcal{T}_i$ and they are not generated with random values created by $\mathcal{T}_i$. Therefore, $\mathcal{T}_i$ sends γ^{s+2} and κ^{s+2} to $\mathcal{A}$.

(vii) $\mathcal{A}$ sends $\delta^{s+2} = \delta^s$ and $\zeta^{s+2} = \zeta^s$ to $\mathcal{T}_i$. $\mathcal{A}$ can not pass the check by the tag $\mathcal{T}_i$ because ζ^{s+2} has to be generated with the current secrets of $\mathcal{T}_i$ and the random number R_3^{s+2} created by $\mathcal{T}_i$. Therefore, ζ^s that are generated with the random number R_3^s can not be used in place of ζ^{s+2}.

6.4. New Attacks and Improvements to Pang et. al's Protocol

Pang et al. [181] proposed a lightweight RFID authentication protocol. This protocol utilized cyclic redundancy check (CRC) and PRNG to create a new tag indexing method, called the two-layer tag indexing mechanism. However, Safkhani and Bagheri [182] presented de-synchronization attack and traceability attack against this protocol by using the following linear property of CRC function [183, 184]:

$$CRC(A||B) = CRC(A \ll n) \oplus CRC(B) \tag{6.3}$$

Safkhani and Bagheri [182] also strengthened Pang et al.'s protocol by using PRNG instead of CRC. The de-synchronization attack presented in Section 6.4.2 can be applied both Pang et al.' protocol and its revised version by Safkhani and Bagheri.

6.4.1. Protocol Description

In Pang et al.'s protocol, each tag $\mathcal{T}_i$ is assigned with two parameters (K_i, SID_i). The server stores an entry for the tag $\mathcal{T}_i$: (K_i^{old}, K_i^{new}, SID_i, D_i). K_i^{new} is the current secret of $\mathcal{T}_i$ and K_i^{old} is the last successfully verified secret of $\mathcal{T}_i$. At the system initialization, $K_i^{old} = K_i^{new}$. Table 6.8 gives the notations used in describing the proposed protocol. The details of the authentication process are composed of the following six steps:

(i) The reader $\mathcal{R}$ generates a random number r and sends it to the tag $\mathcal{T}_i$.

(ii) Upon receiving r, $\mathcal{T}_i$ generates a random number r and computes m_1 and m_2. $\mathcal{T}_i$ sends r_1, m_1 and m_2 to $\mathcal{R}$.

(iii) $\mathcal{R}$ forwards r, r_1, m_1 and m_2 to the back-end server $\mathcal{S}$.

(iv) $\mathcal{S}$ searches its database in order to identify $\mathcal{T}_i$ by checking the validity of m_1 and m_2. If $\mathcal{S}$ identifies $\mathcal{T}_i$, it generates a random number R and computes $n_{right} = CRC_{right}(K_i||SID_i||R||r)$ and sends R, the detailed information D_i and n_{right} to $\mathcal{R}$. $\mathcal{S}$ updates K_i^{old} with K_i^{new} and K_i^{new} with $K_i^{new} \oplus n_{left}$.

(v) $\mathcal{R}$ forwards R and n_{right} to $\mathcal{T}_i$.

(vi) $\mathcal{T}_i$ checks the validity of n_{right} in order to authenticate $\mathcal{R}$. If it is valid, $\mathcal{T}_i$ updates K_i with $K_i \oplus n_{left}$.

6.4.2. De-synchronization Attack

An adversary $\mathcal{A}$ performs the following attack in order to de-synchronize the secrets shared between $\mathcal{R}$ and $\mathcal{T}_i$. For simplicity, we assume that the attack begins after the last successful authentication session $s-1$. $\mathcal{R}$ stores $K_i^{old} = K_i^{s-1}$ and $K_i^{new} = K_i^s$. $\mathcal{T}_i$ stores $K_i = K_i^s$.

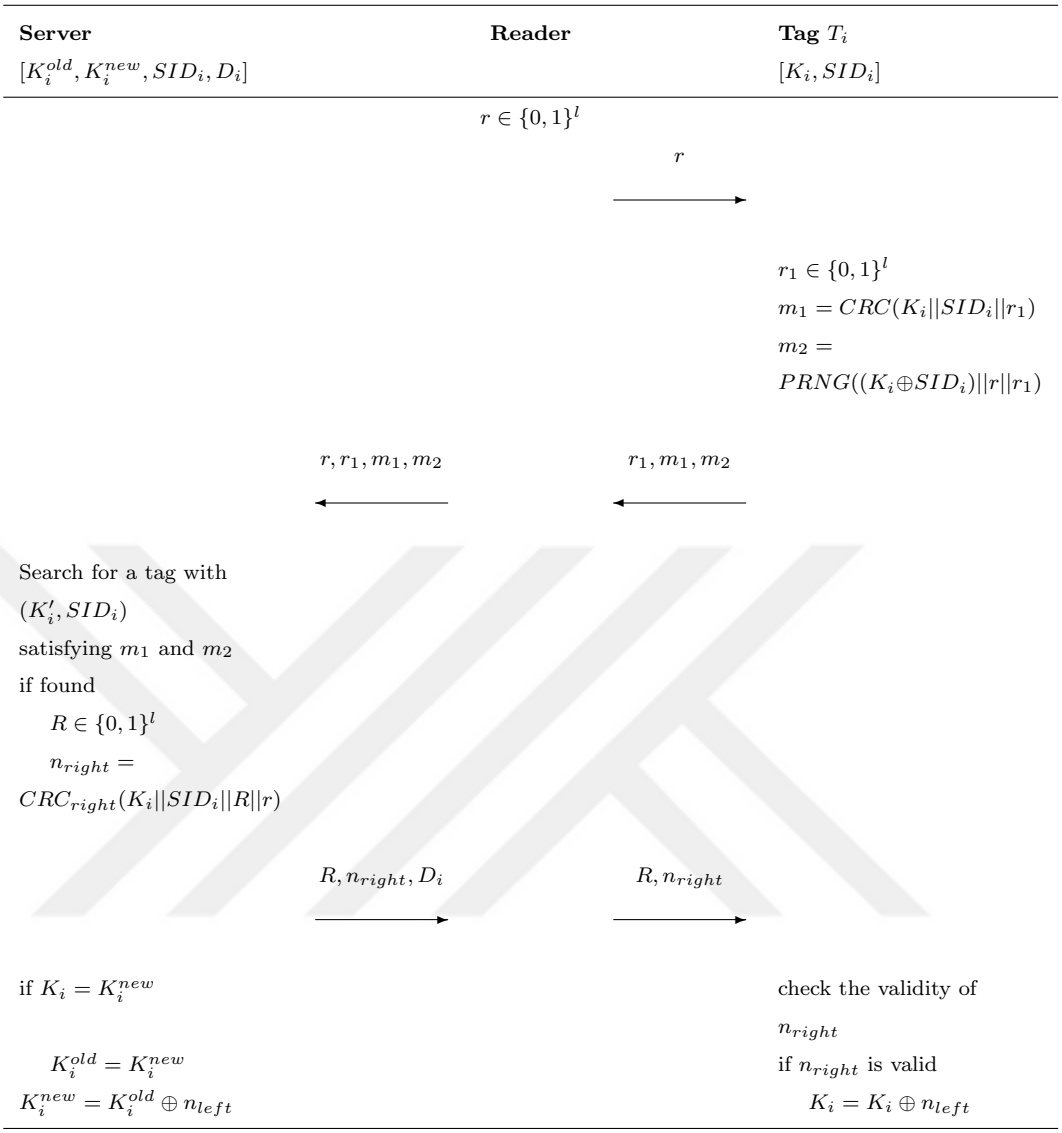

Figure 6.6. Pang et al.'s protocol

(i) In a protocol session s between $\mathcal{R}$ and $\mathcal{T}_i$, $\mathcal{A}$ prevents $\mathcal{T}_i$ from taking last message flow and eavesdrops r^s, R^s and n^s_{right}.

(ii) At the end of the protocol session s, $\mathcal{R}$ updated the secrets related with $\mathcal{T}_i$. However, $\mathcal{T}_i$ did not update its secrets because it did not receive the last message flow. $\mathcal{R}$ stores K_i^s and K_i^{s+1}. $\mathcal{T}_i$ stores K_i^s.

(iii) In a protocol session $s+1$ between $\mathcal{R}$ and $\mathcal{T}_i$, $\mathcal{A}$ prevents $\mathcal{T}_i$ from taking last message flow.

(iv) At the end of the protocol session $s+1$, $\mathcal{R}$ updated the secrets related with $\mathcal{T}_i$. However, $\mathcal{T}_i$ did not update its secrets because it did not receive the last message flow. $\mathcal{R}$ stores K_i^s and K_i^{s+2}. $\mathcal{T}_i$ stores K_i^s.

Table 6.8. Notations for Pang et al.'s protocol

Notation	Description
SID_i	The secure identity of the tag $\mathcal{T}_i$
D_i	The detailed information of the tag $\mathcal{T}_i$
K_i	The secret key of the tag $\mathcal{T}_i$
x_{new}	The current value of x
x_{old}	The previous value of x
x_{left}	The left part of the massage x
x_{right}	The right part of the massage x
CRC	The cyclic redundancy check operation
$PRNG$	The pseudorandom number generator
$\oplus$	The bit-wise XOR operation
$\|\|$	The concatenation operator
$\in$	Random choice operator
$\leftarrow$	The substitution operation

(v) $\mathcal{A}$ sends r^s to $\mathcal{T}_i$ and $\mathcal{T}_i$ sends r_1,m_1 and m_2 to $\mathcal{A}$.

(vi) $\mathcal{A}$ sends R^s and n^s_{right} to $\mathcal{T}_i$. $\mathcal{A}$ passes the check by the tag $\mathcal{T}_i$ because n^s_{right} was generated with the current secret K^s_i of $\mathcal{T}_i$ and random values r^s and R^s. It is not generated with the random value r_1 created by $\mathcal{T}_i$. Therefore, $\mathcal{T}_i$ updates its secret K^s_i by using n^s_{right}. $\mathcal{R}$ stores K^s_i and K^{s+2}_i. $\mathcal{T}_i$ stores K^{s+1}_i. In the next session, $\mathcal{R}$ will not be able to authenticate the tag.

In the above, we show that the adversary $\mathcal{A}$ breaks synchronization between $\mathcal{R}$ and $\mathcal{T}_i$. $\mathcal{A}$ impersonates the valid reader by using the messages eavesdropped in the session s and makes the tag to update its keys. At the end of the attack, the secret values at server side are $K^{new}_i = K^{s+2}_i$ and $K^{old}_i = K^s_i$, and the secret value at the tag side is $K_i = K^{s+1}_i$.

6.4.3. Countermeasure

To defend against de-synchronization, we recommend the following modifications.

In this protocol, the tag generates a random number r_1 after receiving the reader's query. However, the random number r_1 is not used in the computation of the last protocol message. The back-end server should use the random number r_1 in the computation of n_{right} as described below.

$$n_{right} = CRC_{right}(K_i||SID_i||R||r||r_1) \tag{6.4}$$

Let's consider the following attack scenario.

(i) In a protocol session s between $\mathcal{R}$ and $\mathcal{T}_i$, $\mathcal{A}$ prevents $\mathcal{T}_i$ from taking last message flow and eavesdrops r^s, R^s and n^s_{right}.

(ii) At the end of the protocol session s, $\mathcal{R}$ updated the secrets related with $\mathcal{T}_i$. However, $\mathcal{T}_i$ did not update its secrets because it did not receive the last message flow. $\mathcal{R}$ stores K_i^s and K_i^{s+1}. $\mathcal{T}_i$ stores K_i^s.

(iii) In a protocol session $s+1$ between $\mathcal{R}$ and $\mathcal{T}_i$, $\mathcal{A}$ prevents $\mathcal{T}_i$ from taking last message flow.

(iv) At the end of the protocol session $s+1$, $\mathcal{R}$ updated the secrets related with $\mathcal{T}_i$. However, $\mathcal{T}_i$ did not update its secrets because it did not receive the last message flow. $\mathcal{R}$ stores K_i^s and K_i^{s+2}. $\mathcal{T}_i$ stores K_i^s.

(v) $\mathcal{A}$ sends $r^{s+2} = r^s$ to $\mathcal{T}_i$ and $\mathcal{T}_i$ sends r_1^{s+2},m_1^{s+2} and m_2^{s+2} to $\mathcal{A}$.

(vi) $\mathcal{A}$ sends $R^{s+2} = R^s$ and $n^{s+2}_{right} = n^s_{right}$ to $\mathcal{T}_i$. $\mathcal{A}$ cannot pass the check by the tag $\mathcal{T}_i$ because n^{s+2}_{right} has to be generated with the current secret K_i^s of $\mathcal{T}_i$ and random values r^s, R^s and r_1^{s+2}. Therefore, n^s_{right} that are generated with the random number r_1^s can not be used in place of n^{s+2}_{right}.

7. NEW ATTACKS AND IMPROVEMENTS TO CHAOTIC-MAP BASED RFID PROTOCOLS

There are several attempts to provide secure authentication protocol for EPC C1-G2 standard tags. Chien et al. [138] proposed a solution to authentication problem for passive tags. The protocol uses a CRC function, which is considered to be highly linear, that introduces vulnerability against some attacks including backward security, tag forgery and denial of service. In addition to its vulnerabilities, identification process takes long time as implementation of exhaustive search method for the database. Another effort is made by Yeh et al. [185] to provide a secure authentication protocol in which pseudo random number generation is used instead of CRC function. It has been showed that this protocol has security flaws such as forward security, and data integrity by Yoon [186] in which some modifications are proposed to overcome security vulnerabilities. However, in [187], despite Yoon's improvements, the resulting protocol is still considered to have security weaknesses against impersonation, tracking and secret disclosure. Another solution for improving inpatient medication safety using RFID is proposed by Peris-Lopes et al [188]. This solution is also showed to be insecure in secret disclosure aspect by [189].

Cheng et al. [190] utilized Chebyshev chaotic maps in order to solve RFID authentication problem. Benssalah et al. [191] showed that the proposed solution has weaknesses on message generation and shared secret updating. They propose some improvements on the protocol to overcome these weaknesses. In this chapter, we show that despite the proposed improvements, these protocols have fundamental weaknesses stem from message generation. They are vulnerable to tracking, tag impersonation and de-synchronization attacks. The success probabilities of the proposed attacks are significant and their complexities are polynomial. Furthermore, we propose improved RFID authentication protocols. Our protocols utilize the Chebyshev chaotic map hard problem. They eliminate the weaknesses of previous protocols.

7.1. CHEBYSHEV CHAOTIC MAP

In this section, we present some definitions about Chebyshev chaotic maps. These definitions were proposed by Wang and Zhao [192].

Definition 7.1 (Chebyshev polynomials [190]). *Let x be a variable value over the interval [-1,1] and n is an integer. The following function defines a Chebyshev polynomial map* $T_n : R \rightarrow R$ *of degree n:*

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \tag{7.1}$$

where the integer $n \geq 2$, $T_0(x) = 1$, *and* $T_1(x) = x$.

Definition 7.2. *Let x be a variable value over the interval [-1,1] and n is an integer. The following function defines a Chebyshev polynomial map* $T_n : R \rightarrow R$ *of degree n:*

$$T_n(x) = cos(n.arccos(x)) \tag{7.2}$$

Definition 7.3 (Semi-group property). *The Chebyshev polynomial has the following semi-group property:*

$$T_r(T_s(x)) = T_{r.s}(x) \tag{7.3}$$

Definition 7.4 (Commutativity). *The Chebyshev polynomial has the following commutativity property:*

$$T_r(T_s(x)) = T_s(T_r(x)) \tag{7.4}$$

Definition 7.5 (Enhanced Chebyshev polynomials). *Let x be a variable value over the interval* $[-\infty, +\infty]$, $n \geq 2$ *and N is a large prime number. The following function defines a enhanced Chebyshev polynomial:*

$$T_n(x) = (2xT_{n-1}(x) - T_{n-2}(x)) \bmod N \tag{7.5}$$

Definition 7.6 (Discrete logarithm Problem (DLP)). *Given x and y, The discrete logarithm problem is defined as follows: given x and y, find n such that $T_n(x) = y$ is a DLP.*

Definition 7.7 (Diffie - Hellman Problem (DHP)). *The Diffie - Hellman problem is defined as follows: given x, $T_s(x)$ and $T_t(x)$, find n such that $T_n(x)$ where $n = s.t$ is a DHP.*

7.2. New Attacks and Improvements to Cheng et al.'s Protocol

In 2013, Cheng et al. proposed an RFID mutual authentication protocol based on chaotic maps [190]. They utilized enhanced Chebyshev polynomials in the proposed protocol (Definition 7.5). The proposed protocol needs seven exclusive-or and two chaotic cryptographic operations on the tag side. The authors presented the authentication proof of the proposed protocol based on Burrows-Abadi-Needham logic [193]. They also claim that their protocol provides the following security requirements: resistance to replay attacks, resistance to impersonation attacks, resistance to denial-of-service attacks, location privacy and forward secrecy.

7.2.1. Protocol Description

We give the overview of Cheng et al. protocol in Figure 7.1 and list notations in Table 7.1.

For each tag T, the back-end server stores the following entry: $[H(ID) \oplus x_{old}$, $H(ID) \oplus x$, $H(ID)$, ID, x, $x_{old}]$. The tag T stores the current session key x, the secure identity ID and the hashed value of secure identity $H(ID)$. It is assumed that $x_{old} = x$ initially. A step by step description of Cheng et al.'s protocol is given below

(i) The reader generates a random number r and sends it to the tag.

(ii) The tag generates a random number t and computes $M_1 \leftarrow H(ID) \oplus t \oplus r$,$M_2 \leftarrow T_r(T_t(x))$ and $M_3 \leftarrow x \oplus t$ and sends them to the reader.

Table 7.1. Notations of Cheng et al.'s protocol

Notation	Description
ID	The secure identity of the tag
$H(ID)$	The hash value of the identity of the tag
x	The current session key
x_{old}	The last successfully verified session key
$H(ID) \oplus x$	The value used as an index to query the database
$T_{\cdot}(.)$	The enhanced Chebyshev polynomial
$\oplus$	The bit-wise XOR operation
$\in$	Random choice operator
$\leftarrow$	The substitution operation

(iii) After receiving the messages from the tag, the reader forwards them with the random number r to the back-end server.

(iv) After receiving the messages from the reader, the back-end server computes $H(ID) \oplus x = M_1 \oplus M_3 \oplus r$. It checks if there is a record matching with the index $H(ID) \oplus x$. If it finds a record, it gets $H(ID)$, x and x_{old}. Then, it computes $t \leftarrow M_1 \oplus H(ID) \oplus r$ and checks the validity of M_2 by computing $T_r(T_t(x))$ and $T_r(T_t(x_{old}))$. If M_2 is valid, the back-end server generates a random number s and computes $M_4 \leftarrow H(ID) \oplus r \oplus s$, otherwise the session is stopped. If $M_2 = T_r(T_t(x))$, the server computes $M_5 = T_s(T_t(x))$ and replaces x and x_{old} with $x \oplus (t||s)$ and x respectively. If $M_2 = T_r(T_t(x_{old}))$, the server computes $M_5 = T_s(T_t(x_{old}))$ and replaces x with $x_{old} \oplus (t||s)$. The server sends M_4 and M_5 to the reader.

(v) After receiving the messages from the back-end server, the reader forwards them to the tag.

(vi) After receiving the messages from the reader, the tag computes $s \leftarrow M_4 \oplus H(ID) \oplus r$ and checks the validity of M_5 by computing $T_s(T_t(x))$. If M_5 is valid, it replaces x with $x \oplus (t||s)$.

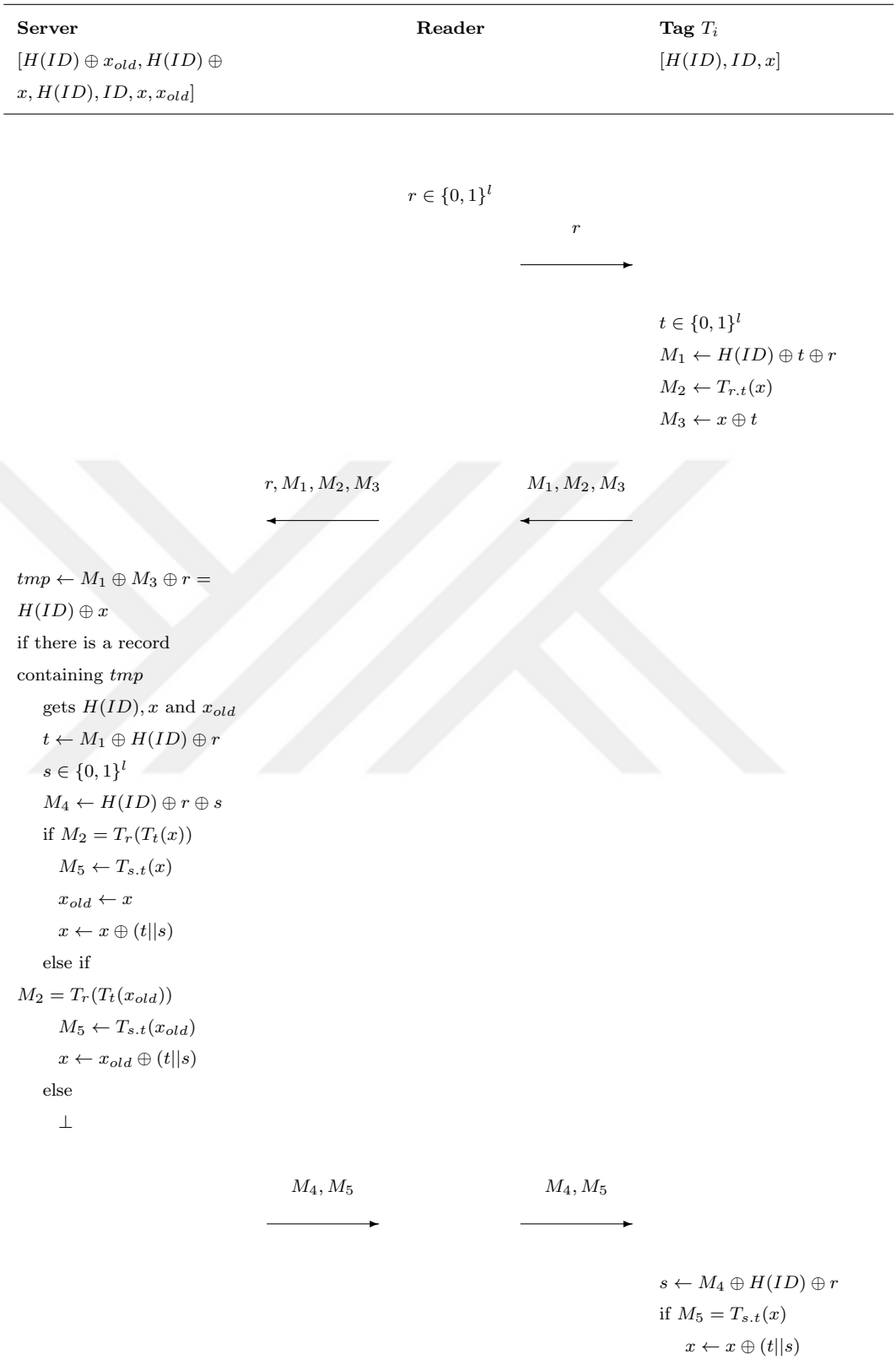

Figure 7.1. Cheng et al.'s protocol

7.2.2. Security Properties

Cheng et al.'s protocol is asserted to have a list of security properties. These properties provided in [190] are summarized below.

- **Mutual authentication:** Mutual authentication is proved by using Burrows-Abadi-Needham (BAN) logic proof [193].
- **Secrecy:** Any secret data cannot be retrieved by any attacker from the communications between the tag and the back-end server. The secret value x is well protected by the enhanced Chebyshev polynomial.
- **Resistance to impersonation attack:** Without knowing the random value t selected by the legal tag and the secret value x stored in the memory of the tag, an attacker cannot pass the authentication in the server side. Only the valid server can compute the correct values M_4 and M_5 with its own selected random number so the attacker cannot pass the tag's authentication.
- **Resistance to replay attack** It is impossible to intercept messages with the intention of replaying them, since any message or information sent from the three components (tag, reader, and server) can always be changed by using random numbers t, r, and s. The random numbers t and s are transmitted securely by using the enhanced Chebyshev polynomials.
- **Resistance to denial-of-service attack** Although the synchronous updating is thus interrupted, the tag's original secret value still can match x_{old} to pass the authentication, such that $M_2 = T_r(T_t(x_{old}))$.
- **Location privacy** Random values t and s that are randomly selected by the tag and the server, respectively, are used to generate the essential data M_2 and M_5 and are used to update the secret constantly. r, t, and s values make the communication messages unpredictable for attackers.
- **Forward secrecy** Even if the attacker has the ability to compromise current session negotiations and retrieve the secret value, he or she still cannot use the compromised data to derive details of previous communications. This is because each session has a different secret x, and the shared key is always updated after individual tag reading.

7.2.3. De-synchronization Attack

We present an attack in which a passive adversary impersonates a tag to the back-end server without knowing the tag's secrets. At the end of the attack, the back-end server performs key-updating but the tag does not. Therefore, the synchronization of the session key between the tag and the back-end server is broken. The details of this attack are given below:

We know that the back-end server has two registers for x values corresponding to the attacked tag namely: x^s_{old} and x^s_{new}. The tags has a register for the current value of x namely: x^t. At the beginning of the attack, the content of the registers are shown in Table 7.2.

Table 7.2. De-synchronization attack on Cheng et al.'s protocol: the content of the registers at the beginning of the attack.

Register	Value
x^s_{new}	x
x^s_{old}	x
x^t	x

Phase 1:

(i) An adversary queries a tag T with a number $r^1 = 1$.

(ii) After receiving the number r^1, the tag T computes $M^1_1 \leftarrow H(ID) \oplus t^1 \oplus r^1$,$M^1_2 \leftarrow T_{r^1}(T_{t^1}(x))$ and $M^1_3 \leftarrow x \oplus t^1$ and sends them to the adversary.

(iii) The adversary computes $H(ID) \oplus t^1 \leftarrow M^1_1 \oplus r^1$. She knows M^1_2 equals $T_{t^1}(x)$ because r_1 equals to 1 (Definition 7.1).

At the end of the Phase 1, neither the tag nor the back-end server performs key-updating. The content of the registers are shown in Table 7.2.

Phase 2:

(i) The reader initiates a valid session by querying tags with a random number r^2.

(ii) After receiving the random number r^2, the tag T computes $M_1^2 \leftarrow H(ID) \oplus t^2 \oplus r^2$,$M_2^2 \leftarrow T_{r^2}(T_{t^2}(x))$ and $M_3^2 \leftarrow x \oplus t^2$ and sends them to the reader.

(iii) The reader forwards r^2, M_1^2, M_2^2 and M_3^2 to the back-end server.

(iv) The server identifies the tag T. It computes $M_4^2 \leftarrow H(ID) \oplus r^2 \oplus s^2$ and $M_5^2 \leftarrow T_{s^2.t^2}(x)$ and sends them to the reader.

(v) The reader forwards M_4^2 and M_5^2 to the tag.

(vi) At the end of this valid session, the tag and the back-end server perform key-updating.

At the end of the Phase 2, the content of the registers are as shown in Table 7.3.

Table 7.3. De-synchronization attack on Cheng et al.'s protocol: the content of the registers at the end of Phase 2.

Register	Value
x_{new}^s	$x \oplus (t^2 \| \| s^2)$
x_{old}^s	x
x^t	$x \oplus (t^2 \| \| s^2)$

Phase 3:

(i) The reader initiates a valid session by querying tags with a random number r^3.

(ii) After receiving the random number r^3, the adversary has to create valid messages in order to pass the check by the back-end server. She obtained $H(ID) \oplus t^1$, $T_{t^1}(x)$ and $x \oplus t^1$ in the Phase 1. She will use these values to create valid M_1^3, M_2^3 and M_3^3. She computes $M_1^3 \leftarrow H(ID) \oplus t^1 \oplus r^3$, $M_2^3 \leftarrow T_{r^3}(T_{t^1}(x))$ and $M_3^3 \leftarrow x \oplus t^1$ and sends them to the reader.

(iii) After receiving messages r^3, M_1^3, M_2^3 and M_3^3 from the adversary, the reader forwards them to the back-end server.

(iv) The back-end server computes $H(ID) \oplus x = M_1^3 \oplus M_3^3 \oplus r^3$. The back-end server gets $H(ID)$ and x_{old}^s from the record matching with the index $H(ID) \oplus x$. We know that the content of the register x_{old}^s equals x. The back-end server computes $t^1 \leftarrow M_1^3 \oplus H(ID) \oplus r^3$. It checks the validity of M_2^3 by computing $T_{r^3}(T_{t^1}(x))$. The adversary passes this check because she creates M_2^3 with the valid r^3 and t^1 values. After that the back-end server generates a random number s^3 and replaces x_{new}^s and x_{old}^s with $x \oplus (t^1||s^3)$ and x respectively.

At the end of the Phase 3, the content of the registers are as shown in Table 7.4. In the above attack, the adversary is authenticated by the back-end database as a legitimate tag with a success probability of 1. The given attack makes the shared secrets out-of-synchronization.

Table 7.4. De-synchronization attack on Cheng et al.'s protocol: the content of the registers at the end of Phase 3.

Register	Value
x_{new}^s	$x \oplus (t^1\|\|s^3)$
x_{old}^s	x
x^t	$x \oplus (t^2\|\|s^2)$

7.2.4. Secret Disclosure Attack

In this section, we present a passive attack in which an adversary retrieves secret information $H(ID)$ and x in the tag. In this attack, an adversary benefits from weakness in key-updating mechanism. She can disclose all secret parameters by eavesdropping one session of the protocol as follows:

(i) An adversary eavesdrops a transcript of one protocol session between the tag T and the reader. She stores r, M_1, M_2 and M_3.

(ii) The adversary queries the tag T with the random number r'.

(iii) After receiving r', the tag T computes M_1', M_2' and M_3' and sends them to the adversary.

(iv) The adversary computes $(M_1' \oplus M_3' \oplus r') \oplus (M_1 \oplus M_3 \oplus r) = (H(ID) \oplus x') \oplus (H(ID) \oplus x) = x' \oplus x = x \oplus (t||s) \oplus x = (t||s)$. The adversary gets the values of t and s. She computes $M_1 \oplus r \oplus t = H(ID)$ and $M_3 \oplus t = x$. The adversary gets the values of $H(ID)$ and x. Finally, she computes $x \oplus (t||s) = x'$.

An adversary knowing the secret values $H(ID)$ and x' can easily perform traceability, tag impersonation, reader impersonation and de-synchronization attacks with a success probability 1.

7.2.5. Revised Protocol

Cheng et al. utilized chaotic maps in their protocol. However, they do not use any advantage of chaotic maps such as semi-group property. In the previous section, we show that an adversary can use semi-group property of chaotic maps to make the shared secrets desynchronized. Cheng et al. also use inexpensive $\oplus$ operation for key-updating. In the previous section, we show that an adversary can use weaknesses of $\oplus$ operation in order to disclose the secrets of a tag.

Our revised mutual authentication protocol is the same as Cheng et al.'s protocol except the way the message M_2 and M_5 are created. In Cheng et al.'s protocol, M_2 and M_5 are created by $T_{r.t}(x)$ and $T_{s.t}(x)$ respectively. In our revised mutual authentication protocol, we utilize a keyed hash function $f_k(.)$ for computing M_2 and M_5. M_2 and M_5 are created by $f_{r||t}(x)$ and $f_{s||t}(x)$ respectively. Furthermore, we change the way the key-updating. In Cheng et al.'s protocol, x is updated with $x \oplus (t||s)$. We revise it to be $x \leftarrow f_1(x \oplus (t||s))$. We make these revisions in order to prevent attacks detailed in Section 7.2.3 and Section 7.2.4. Our revised protocol is summarized in Figure 7.2.

In Cheng et al.'s protocol, the back-end server authenticates the tag by checking the validity of the message M_2. The weakness of this protocol is that an adversary who does not know the values of x and t can create the valid $M_2 \leftarrow T_{r.t}(x)$ by using the

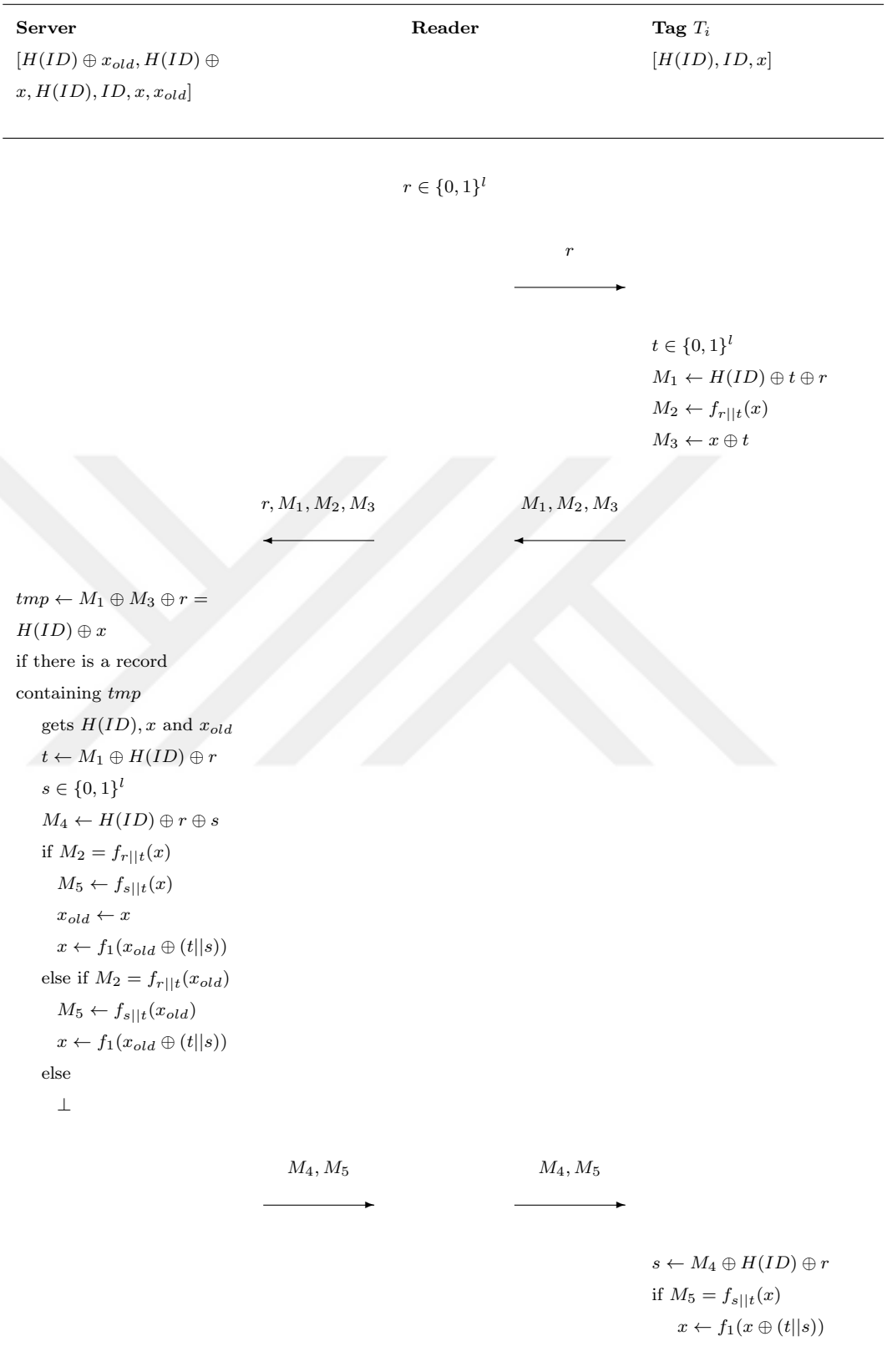

Figure 7.2. Revised version of Cheng et al.'s protocol

semi-group property of Chebyshev polynomials. In our revised protocol, we utilize a keyed hash function $f_k(.)$ to create M_2. If the adversary can create valid $M_2 \leftarrow f_{r||t}(x)$ without knowing the values of x and t, this will contradict with pseudo-randomness of $f_k(.)$.

In Cheng et al.'s protocol, the key-updating is done by replacing x with $x \oplus (t||s)$. In the above, we show that an adversary can learn the value of $H(ID) \oplus x$ for each protocol session. $H(ID)$ value is constant value. That means the adversary can get the difference of x values used in two consecutive protocol sessions. This difference equals to concatenation of t and s values used in the first session. As a result, the adversary can learn the values of x and $H(ID)$.

In our revised protocol, the secret value x is replaced with $f_1(x \oplus (t||s))$. If the adversary learns the values of t and s from $f_1(x \oplus (t||s))$, this will contradict with pseudo-randomness of $f_k(.)$.

7.3. New Attacks and Improvements to Benssalah et al.'s Protocol

7.3.1. Preliminaries

Definition 7.8 (Security [26])**.** *A scheme provides security if it carries out tag and reader authentication securely.*

(i) If a polynomial-time adversary is identified as an uncorrupted legitimate tag T_i by the reader on the session π with non-negligible probability and there is not any matching conversation between T_i and π, tag authentication is considered as insecure.

(ii) If a polynomial-time adversary is identified as a legitimate reader the an uncorrupted legitimate tag T_i on the session π with non-negligible probability and there is not any matching conversation between the reader and π, reader authentication is considered as insecure.

Definition 7.9 (Universal Untraceability)**.** *Universal untraceability is privacy notion defined by Avoine [21]. Universal untraceability requires that an adversary cannot find a correlation between a tag's two responses which are separated by a successful authentication with a valid reader. In the game below, universal untraceability is modeled among an adversary $\mathcal{A}$ and the challenger $\mathcal{C}$.*

Phase 1: (Learning) *An adversary $\mathcal{A}$ interacts with any two legitimate tags T_0 and T_1. $\mathcal{A}$ is able to start, monitor, and break authentication sessions between T_0 the reader R and $\mathcal{A}$ is able to start, monitor, and break authentication sessions between T_1 the reader R.*

Phase 2: (Challenge) *A challenger $\mathcal{C}$ performs protocol instances on T_0 and T_1 with the reader successfully. $\mathcal{C}$ chooses one of the two tags as T_b. $\mathcal{A}$ interacts with the tag T_b. $\mathcal{A}$ is able to start, monitor, and break authentication sessions between T_b and R.*

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

Definition 7.10 (Forward Privacy (Backward Untraceability))**.** *Forward Privacy (backward untraceability) is a strong privacy notions proposed in [23]. Forward Privacy requires that an adversary getting access to internal state of a tag at time t cannot identify the past interactions of the tag occurred before time t. In [194], forward privacy is modeled among an adversary $\mathcal{A}$ and the challenger $\mathcal{C}$.*

Phase 1: (Learning) *An adversary $\mathcal{A}$ interacts with any two legitimate tags T_0 and T_1. $\mathcal{A}$ is able to start, monitor, and break authentication sessions between T_0 the reader R and $\mathcal{A}$ is able to start, monitor, and break authentication sessions between T_1 the reader R.*

Phase 2: (Challenge) *A challenger $\mathcal{C}$ performs protocol instances on T_0 and T_1 with the reader successfully. $\mathcal{C}$ chooses one of the two tags as T_b. $\mathcal{A}$ interacts with the tag T_b. $\mathcal{A}$ is able to start, monitor, and break authentication sessions between T_b and R. $\mathcal{A}$ is also given access to authentication outcomes. Then, $\mathcal{A}$ is given access to internal state of the tag T_b.*

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

Definition 7.11 (Existential Untraceability)**.** *Existential untraceability is privacy notion defined by Avoine [21]. Existential untraceability requires that an adversary cannot find a correlation between a tag's two responses which are not necessarily separated by a successful authentication with a valid reader. In the game below, existential untraceability is modeled among an adversary $\mathcal{A}$ and the challenger $\mathcal{C}$.*

Phase 1: (Learning) *An adversary $\mathcal{A}$ interacts with any two legitimate tags T_0 and T_1. $\mathcal{A}$ is able to start, monitor, and break authentication sessions between T_0 the reader R and $\mathcal{A}$ is able to start, monitor, and break authentication sessions between T_1 the reader R.*

Phase 2: (Challenge) *$\mathcal{C}$ selects one of the tags as T_b. $\mathcal{A}$ interacts with the tag T_b. $\mathcal{A}$ is able start, monitor, and break authentication sessions between T_b and R.*

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

The advantage of any adversary identifying the tag in the above games is defined as follows:

$$Adv(A) = 2(Pr[b' = b] - \frac{1}{2}) \tag{7.6}$$

7.3.2. Description of Benssalah et al.'s Protocol

In 2014, Benssalah et al. [191] showed the vulnerabilities of Cheng et al.'s [190] chaotic map based RFID authentication protocol. They also proposed an improved RFID authentication protocol based on chaotic maps. They utilized enhanced Chebyshev polynomials in the proposed protocol (Definition 7.5). The proposed protocol needs one random number generation and four chaotic cryptographic operations on the tag side. The authors claim that their protocol provides the following security re-

quirements: resistance to replay attacks, resistance to impersonation attacks, resistance to denial-of-service attacks, mutual authentication and mobility.

We give the notations used in Benssalah et al.'s protocol in Table 7.5.

Table 7.5. Notations for Benssalah et al.'s protocol

Notation	Description
ID	The secure identity of the tag
$H(ID)$	The hash value of the identity of the tag
x	The current session key
x_{old}	The last successfully verified session key
$T_{\cdot}(.)$	The enhanced Chebyshev polynomial
$\oplus$	The bit-wise XOR operation
$\in$	Random choice operator
$\|\|$	Concatenation operator
$\leftarrow$	Substitution operation

We give the overview of Benssalah et al. protocol in Figure 7.3. This protocol has two phases: initialization phase and authentication phase.

7.3.2.1. Initialization Phase. In the initialization phase, a secret key x is generated for each tag in the back-end server. The back-end server stores [x_{old}, x_{new}, c_{old}, c_{new}, $H(ID)$, ID] entry for each tag in its database where c_{old} and c_{new} are index values and ID is a tag identifier. A tag stores [ID, $H(ID)$, x, c_i] in its memory. A reader stores its identifier RID. In the beginning $x_{new} = x_{old} = x$ and $c_{new} = c_{old} = 0$.

7.3.2.2. Authentication Phase.

(i) The reader generates a random number r and sends it to the tag.

(ii) After receiving r, the tag generates a random number t and computes M_1, M_2, M_3. The tag sends (M_1, M_2, M_3, c_i) to the reader.

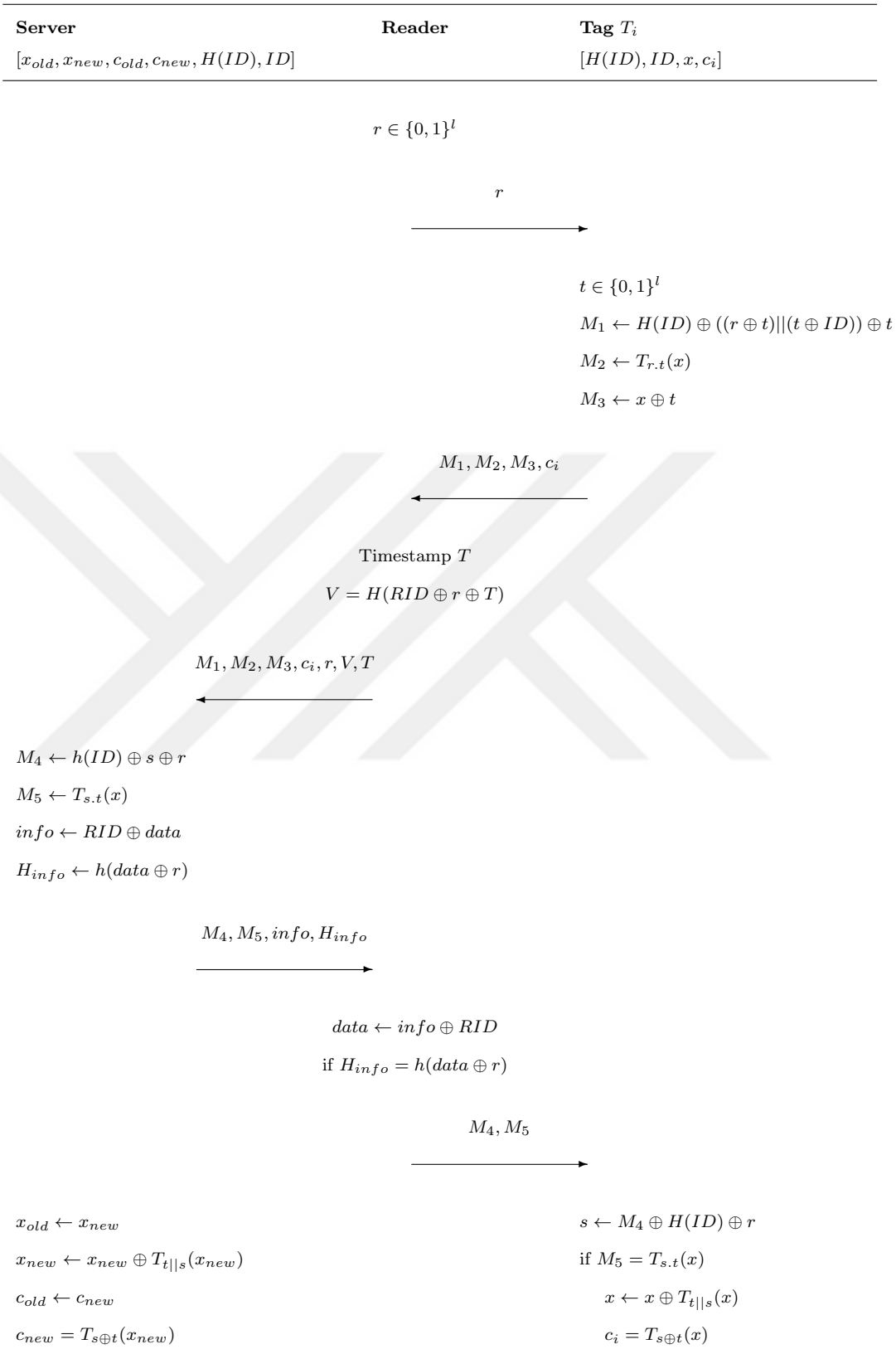

Figure 7.3. Benssalah et al.'s protocol

(iii) The reader creates a timestamp T and computes the following value.

$$V = h(RID \oplus r \oplus t) \tag{7.7}$$

The reader sends $(M_1, M_2, M_3, c_i, r, V, T)$ to the back-end server.

(iv) The back-end server checks the validity of V. If V is valid, it performs the following operations.

(a) If $c_i = 0$,

- The back-end server performs an exhaustive search on its database in order to find corresponding records. For each entry in its database, the back-end server computes the following values.

$$\begin{aligned} T_{old} &= T_{(M_3 \oplus x_{old}).r}(x_{old}) \\ T_{new} &= T_{(M_3 \oplus x_{old}).r}(x_{new}) \end{aligned} \tag{7.8}$$

  if M_2 matches T_{old} or T_{new}, the back-end server finds the corresponding records. It sets x to x_{old} or x_{new}.
- The back-end server checks the validity of M_1. If M_1 is not valid, it rejects the tag.

(b) If $c_i \neq 0$ (common case),

- c_i is index of corresponding database entry. The back-end server finds a database entry where c_i matches with c_{old} or c_{new} and sets x to x_{old} or x_{new}.
- The back-end server checks the validity of M_1 and M_2. If one of these values is not valid, it rejects the tag.

(c) The back-end server computes $(M_4, M_5, info, H_{info})$ and sends them to the reader.

$$\begin{aligned} M_4 &= h(ID) \oplus s \oplus r \\ M_5 &= T_{s.t}(x) \\ info &= RID \oplus data \\ H_{info} &= h(data \oplus r) \end{aligned} \tag{7.9}$$

(d) The back-end server performs the following operations for key updating.

$$\begin{aligned} x_{old} &= x_{new} \\ x_{new} &= x_{new} \oplus T_{t||s}(x_{new}) \\ c_{old} &= c_{new} \\ c_{new} &= T_{s\oplus t}(x_{new}) \end{aligned} \tag{7.10}$$

(v) The reader computes $data = info \oplus RID$ and checks the validity of H_{info}. If H_{info} is valid, the reader sends (M_4, M_5) to the tag.

(vi) The tag retrieves s from M_4.

$$s = M_4 \oplus h(ID) \oplus r \tag{7.11}$$

(vii) The tag checks the validity of M_5. If M_5 is valid, the tag performs the following operations for key updating.

$$\begin{aligned} x &= x \oplus T_{t||s}(x) \\ c_i &= T_{s\oplus t}(x) \end{aligned} \tag{7.12}$$

7.3.3. Tracking Attack

The bit length of ID, $H(ID)$, x and c_i is specified as m in [191]. The bit length of random numbers is also m. Bensallah et al. use concatenation operation $\|$ in their protocol in the generation of message M_1.

$$M_1 = h(ID) \oplus ((r \oplus t) \parallel (t \oplus ID)) \oplus t \tag{7.13}$$

It is easily seen that the length of message M_1 is $2m$ (Equation 7.13). The first m-bit part is $h(ID) \oplus (r \oplus t) \oplus r$ and the second m-bit part is $t \oplus ID$. The calculation of M_1 reveals a serious weakness. In the following attack, we show that how a passive adversary can use this weakness in order to trace tags.

(i) An adversary A eavesdrops the last successful session between a tag T_i and the reader or sends challenge r to a tag T_i to start a new protocol session.

(ii) A records messages r and M_1. If A started the session, A aborts the session and ignores the next step.

Because of concatenation operation used in calculation of message M_1, this bit length of M_1 is actually $2m$. Below, we show how the first and second m-bit parts are calculated in Equation 7.14.

$$\begin{aligned} M_1 &= h(ID) \oplus ((r \oplus t) \parallel (t \oplus ID)) \oplus t \\ &= (h(ID) \oplus r \oplus t \oplus t) \parallel (t \oplus ID) \\ &= (h(ID) \oplus r) \parallel (t \oplus ID) \end{aligned} \tag{7.14}$$

(iii) A gets the first m-bit part $(h(ID) \oplus r)$ from message M_1. A derives the significant value $H(ID) = (h(ID) \oplus r) \oplus r$. $H(ID)$ is constant for all protocol sessions. Therefore, A can use it to track person or object carrying the tag.

In the above, we show that how a passive adversary can trace tags. The success probability of the proposed attack is 1 and the attack complexity is one protocol run.

7.3.4. Impersonation Attack

In tag impersonation attack, forged tags are identified by a legitimate reader. In the following attack, an active adversary fools the reader and is identified as a legitimate tag by the reader.

(i) An adversary A eavesdrops the last successful session between a tag T_i and the reader or sends challenge r to a tag T_i to start a new protocol session.

(ii) A records messages r, M_1, M_2, M_3 and c_i. If A started the session, A cancels the session and neglects the following steps.

(iii) The reader R sends challenge r' to start a new session.

(iv) A divides the message M_1 into two parts and learns the constant value $h(ID)$ as shown in Equation 7.14. A impersonate the tag T_i by calculating messages (M_1', M_2', M_3', c_i') as follows:

$$\begin{aligned} M_1' &= (h(ID) \oplus r') \parallel (t \oplus ID) \\ M_2' &= T_{\frac{r'}{r}}(M_2) \\ &= T_{\frac{r'}{r}}(T_{r.t}(x)) \\ &= T_{\frac{r'}{r}.r.t}(x) \\ &= T_{r'.t}(x) \\ M_3' &= M_3 \\ &= x \oplus t \\ c_i' &= c_i \end{aligned} \tag{7.15}$$

(v) Upon receiving the messages M_1', M_2', M_3', c_i' and r', the back-end goes to database record by using index c_i'. It gets the secret x and calculates $t = M_3' \oplus x$. It checks the validity of M_1' and M_2 as follows:

$$\begin{aligned} M_1' &= h(ID) \oplus ((r' \oplus t) \parallel (t \oplus ID)) \oplus t \\ &= (h(ID) \oplus r' \oplus t \oplus t) \parallel (t \oplus ID) \\ &= (h(ID) \oplus r') \parallel (t \oplus ID) \\ M_2' &= T_{r'.t}(x) \end{aligned} \tag{7.16}$$

(vi) As shown in Equation 7.16, A is authenticated by the back-end server. The back-end server calculates M_4 and M_5 and sends them to the reader R.

In the attack above, the active adversary is identified as a legitimate tag by the back-end server. The success probability of the attack is 1. The complexity of the attack is two protocol runs and memory and time requirements is negligible.

7.3.5. Desynchronization attack

In desynchronization attacks, adversaries try to force the back-end server or the tag to update their common secrets. As a result, one party updates common secrets but other party does not. To defend against de-synchronization attacks, the back end server stores both the current and previous secrets of the tag in its database. The only way for the adversary to desynchronize the back-end server and the tag is to impersonate the back-end server to the tag. In the following attack, we show that how a passive adversary can desynchronize the tag and the reader easily.

(i) An adversary A queries a tag T_i by sending r.

(ii) The tag T_i calculates messages M_1, M_2 and M_3 and send (M_1,M_2,M_3,c_i) to A.

(iii) After receiving (M_1,M_2,M_3,c_i), A learns the constant value $h(ID)$ as shown in Equation 7.14. A impersonate the the reader R by calculating messages (M_4, M_5) as follows:

$$\begin{aligned} s &= \{0,1\}^l \\ M_4 &= h(ID) \oplus s \oplus r \\ M_5 &= T_{\frac{s}{r}}(M_2) \\ &= T_{\frac{s}{r}}(T_{r.t}(x)) \\ &= T_{\frac{s}{r}.r.t}(x) \\ &= T_{s.t}(x) \end{aligned} \tag{7.17}$$

(iv) After receiving (M_4, M_5), the tag T_i extracts s and checks whether M_5 is equal to $T_{s.t}(x)$. If equality is correct, the tag T_i updates $x = x \oplus T_{t||s}(x)$ and $c_i = T_{s \oplus t}(x)$.

Following the above attack, secret values contained in T_i are set to $x \oplus T_{t||s}(x)$ and $T_{s \oplus t}(x)$ while the stored values in the back-end server are x and c_i. Hence, the back-end server never authenticates T_i in the next sessions of protocol. The success probability of attack is 1 and the complexity of attack is only one run of protocol.

7.3.6. Improved Protocol

We propose some improvements over Benssalah et al.'s protocol and overview of the improved protocol is given in Figure 7.4. This protocol also has two phases: initialization phase and authentication phase.

7.3.6.1. Initialization Phase. In the initialization phase, a secret key x is generated for each tag in the back-end server. The back-end server stores $[x_{old}, x_{new}, c_{old}, c_{new}, ID]$ entry for each tag in its database where c_{old} and c_{new} are index values and ID is a tag identifier. A tag stores $[ID, x, c_i]$ in its memory. A reader stores its identifier RID. In the beginning, $x_{new} = x_{old} = x$ and $c_{new} = c_{old} = 0$.

7.3.6.2. Authentication Phase.

(i) The reader generates a random number r and sends it to the tag.

(ii) After receiving r, the tag generates a random number t, computes M_1, M_2 and sends them to the reader.

$$\begin{aligned} M_1 &= T_{r.t}(x \oplus t) \\ M_2 &= x \oplus t \end{aligned} \tag{7.18}$$

(iii) The reader creates a timestamp T and sends (M_1, M_2, c_i, r, V, T) to the back-end server..

(iv) If V is valid, it performs the following operations.

(a) If $c_i = 0$,

- For each entry, the back-end server computes the following values.

$$\begin{aligned} T_{old} &= T_{(M_2 \oplus x_{old}).r}(x_{old} \oplus r) \\ T_{new} &= T_{(M_2 \oplus x_{old}).r}(x_{new} \oplus r) \end{aligned} \tag{7.19}$$

if M_1 matches T_{old} or T_{new}, the back-end server finds the corresponding records. It sets x to x_{old} or x_{new}.

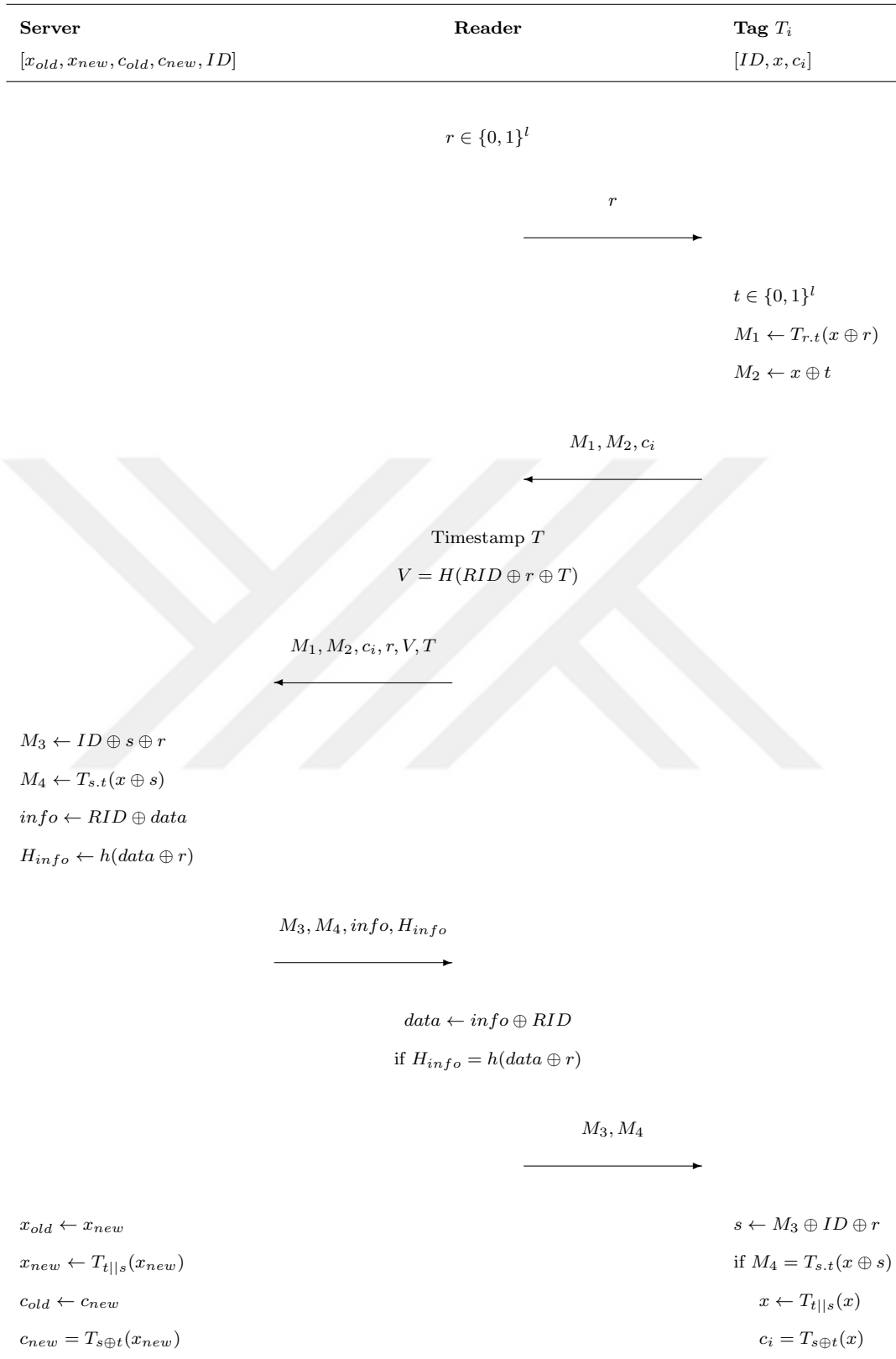

Figure 7.4. Revised version of Benssalah et al.'s protocol

(b) If $c_i \neq 0$ (common case),

- c_i is index of corresponding database entry. The back-end server finds a database entry where c_i matches with c_{old} or c_{new} and sets x to x_{old} or x_{new}.
- The back-end server checks the validity of M_1. The value is not valid, it rejects the tag.

(c) The back-end server computes $(M_3, M_4, info, H_{info})$ and sends them to the reader.

$$\begin{aligned} M_3 &= ID \oplus s \oplus r \\ M_4 &= T_{s.t}(x \oplus s) \\ info &= RID \oplus data \\ H_{info} &= H(data \oplus r) \end{aligned} \tag{7.20}$$

(d) The back-end server performs the following operations for key updating.

$$\begin{aligned} x_{old} &= x_{new} \\ x_{new} &= x_{new} \oplus T_{t||s}(x_{new}) \\ c_{old} &= c_{new} \\ c_{new} &= T_{s \oplus t}(x_{new}) \end{aligned} \tag{7.21}$$

(v) The reader computes $data = info \oplus RID$ and checks the validity of H_{info}. If H_{info} is valid, the reader sends (M_3, M_4) to the tag.

(vi) The tag retrieves s from M_3.

$$s = M_3 \oplus ID \oplus r \tag{7.22}$$

(vii) If M_4 is valid, the tag performs the following operations for key updating.

$$\begin{aligned} x &= T_{t||s}(x) \\ c_i &= T_{s \oplus t}(x) \end{aligned} \tag{7.23}$$

7.3.7. Security Analysis of Improved Protocol

Lemma 7.1. *The secret values of a tag cannot be exposed without corrupting the tag.*

Proof. The tag responds with $T_{r.t}(x \oplus r)$, $x \oplus t$ and c_i to each query. The current random value t is used in construction of messages $T_{r.t}(x \oplus r)$ and $x \oplus t$. An adversary $\mathcal{A}$ can not retrieve the random number t from messages $T_{r.t}(x \oplus r)$ and $x \oplus t$ with non-negligible probability because $T.(.)$ is a Chebyshev chaotic map and t is protected by Chebyshev chaotic map hard problem (7.6 and 7.7). The back-end server responds to the tag with $ID \oplus s \oplus r$ and $T_{s.t}(x \oplus s)$. These messages are constructed with random number s which is unknown to $\mathcal{A}$. An adversary $\mathcal{A}$ can not retrieve the random number s from messages $ID \oplus s \oplus r$ and $T_{s.t}(x \oplus s)$ with non-negligible probability because $T.(.)$ is a Chebyshev chaotic map and t is protected by Chebyshev chaotic map hard problem (7.6 and 7.7). After each successful authentication session, x and c_i are updated by using Chebyshev chaotic map. Therefore, the outdated and updated values of x and c_i cannot be correlated with non-negligible probability.

As a result, $\mathcal{A}$ cannot obtain secret values of the tag without corrupting it.

7.3.7.1. Mutual Authentication.

Theorem 7.2. *The proposed protocol provides tag authentication if $T_{\cdot}(.)$ is a Chebyshev chaotic map (Definiton 7.6 and Definiton 7.7).*

Proof. We assume that there is an adversary $\mathcal{A}$ that can impersonate a tag T_i to a reader R with non-negligible probability. After receiving a message r, $\mathcal{A}$ has to generate messages M_1, M_2 and c_i such that

$$\begin{aligned} M_1 &= T_{r.t}(x \oplus r) \\ M_2 &= x \oplus t \end{aligned} \tag{7.24}$$

$\mathcal{A}$ can use previous responses of the tag T_i. For example, $\mathcal{A}$ eavesdrops the transcript $(M_1^s,\ M_2^s,\ c_i^s)$ from the session s between the tag T_i and the reader. In the session $s+1$, the reader queries $\mathcal{A}$ with r^{s+1}. $\mathcal{A}$ can easily generate $M_2^{s+1} = M_2^s$ and $c_i^{s+1} = c_i^s$. In order to generate M_1^{s+1}, $\mathcal{A}$ needs to know t^s. M_1 is generated by computing $T_{r.t}(x \oplus r)$ where r comes from the reader. $\mathcal{A}$ can compute $T_{t^s}(x^s \oplus r^s)$ as follows:

$$\begin{aligned} T_{t^s}(x^s \oplus r^s) &= T_{\frac{1}{r^s}}(M_1^s) \\ &= T_{\frac{1}{r^s}}(T_{r^s.t^s}(x^s \oplus r^s)) \\ &= T_{\frac{1}{r^s}.r^s.t^s}(x^s \oplus r^s) \\ &= T_{t^s}(x^s \oplus r^s) \end{aligned} \tag{7.25}$$

It is difficult and ineffective retrieving t^s from $T_{t^s}(x^s \oplus r^s)$ because $T.(.)$ is a Chebyshev chaotic map and t^s is protected by Chebyshev chaotic map hard problem (7.6 and 7.7). As a result, $\mathcal{A}$ can use these previous responses with the negligible probability $2^{1-l}N$ where l is the security parameter (the bit length of random nonces and messages).

Theorem 7.3. *The proposed protocol provides reader authentication if $T_{.}(.)$ is a Chebyshev chaotic map (Definiton 7.6 and Definiton 7.7).*

Proof. We assume that there is an adversary $\mathcal{A}$ that can impersonate a reader R to a tag T_i with non-negligible probability. After receiving messages M_1, M_2 and c_i $\mathcal{A}$ has to generate messages M_3 and M_4 such that

$$\begin{aligned} M_3 &= ID \oplus s \oplus r \\ M_4 &= T_{s.t}(x \oplus s) \end{aligned} \tag{7.26}$$

$\mathcal{A}$ can use previous responses of the reader R. For example, $\mathcal{A}$ records the messages M_3^s and M_4^s from the session s between the tag T_i and the reader and prevents the tag T_i from receiving messages M_3^s and M_4^s. Thus, the tag T_i does not update x and c_i. $\mathcal{A}$ starts the session $s+1$ by querying the tag T_i with r^s. The tag T_i gen-

erates a random number t^{s+1} and sends M_1^{s+1}, M_1^{s+1}, c_i^{s+1} $\mathcal{A}$. $\mathcal{A}$ can easily generate $M_3^{s+1} = M_3^s = ID \oplus s^s \oplus r^s$. In order to generate M_4^{s+1}, $\mathcal{A}$ needs to know t^{s+1} and $T_{s^s}(x^s \oplus r^s)$. It is difficult and ineffective retrieving $T_{s^s}(x^s \oplus r^s)$ from $T_{t^s.s^s}(x^s \oplus r^s)$ because $T.(.)$ is a Chebyshev chaotic map and t^s and s^s are protected by Chebyshev chaotic map hard problem (7.6 and 7.7). As a result, $\mathcal{A}$ can use these previous responses with the negligible probability $2^{1-l}N$ where l is the security parameter (the bit length of random nonces and messages).

7.3.7.2. Privacy.

Theorem 7.4. *The proposed protocol is universal untraceable if $T_{\cdot}(.)$ is a Chebyshev chaotic map (Definiton 7.6 and Definiton 7.7).*

Proof. We assume that there is an adversary $\mathcal{A}$ that wins the universal untraceability experiment with non-negligible probability.

In the learning phase, $\mathcal{A}$ starts, monitors, and breaks authentication sessions between a tag T_0 and a reader R and starts, monitors, and breaks authentication sessions between a tag T_1 and the reader R.

In the challenge phase, the reader R carries out successful authentications with T_0 and T_1. Therefore, T_0 and T_1 update their secret values. $\mathcal{A}$ starts, monitors, and breaks authentication sessions between the tag T_b (b equals to 0 or 1) and the reader R.

In the guess phase, $\mathcal{A}$ outputs a guess b' for the value of b. $\mathcal{A}$ has to find a correlation between tags' responses obtained in the learning phase and responses of T_b obtained in challenge phase. $\mathcal{A}$ cannot find the correlation with non-negligible probability because these responses are separated by at least one successful authentication in which secret values of T_b are updated with Chebyshev chaotic map (7.6 and 7.7).

Theorem 7.5. *The proposed protocol achieves forward privacy if $T_{\cdot}(.)$ is a Chebyshev chaotic map (Definiton 7.6 and Definiton 7.7).*

Proof. We assume that there is an adversary $\mathcal{A}$ that wins the forward privacy experiment with non-negligible probability.

In the learning phase, $\mathcal{A}$ starts, monitors, and breaks authentication sessions between a tag T_0 and a reader R and starts, monitors, and breaks authentication sessions between a tag T_1 and the reader R.

In the challenge phase, $\mathcal{A}$ starts, monitors, and breaks authentication sessions between the tag T_b (b equals to 0 or 1) and a reader R. At the end of the challenge phase, $\mathcal{A}$ is given access to internal state of the tag T_b and learns ID, the current value of x^{s+1} and c_i^{s+1}.

$$\begin{aligned} & ID \\ & x^{s+1} = T_{t^s||s^s}(x^s) \\ & c_i^{s+1} = T_{s^s \oplus t^s}(x^s) \end{aligned} \tag{7.27}$$

In the guess phase, $\mathcal{A}$ outputs a guess b' for the value of b. $\mathcal{A}$ has to find correlation between the current internal state of the tag T_b and authentication exchanges eavesdropped in the learning phase.

Suppose that $\mathcal{A}$ eavesdrops one authentication exchange in the challenge phase. This means the internal state of the tag T_b is changed one time after the learning phase. $\mathcal{A}$ has to find correlation between the current internal state of the tag T_b and the authentication exchange eavesdropped in the challenge phase.

The last authentication exchange of the tag T_b in the learning phase as follows:

$$\begin{aligned} & r^s \\ M_1^s &= T_{r^s.t^s}(x^s \oplus r^s) \\ M_2^s &= x^s \oplus t^s \\ c_i^s &= c_i^s \\ M_3^s &= ID \oplus s^s \oplus r^s \\ M_4^s &= T_{s^s.t^s}(x^s \oplus s^s) \end{aligned} \tag{7.28}$$

In order to find correlation, $\mathcal{A}$ can use $M_3^s = ID \oplus s^s \oplus r^s$ because ID is fixed value for all authentication exchanges. However, $\mathcal{A}$ has to know the random value s^s in order to retrieve ID. $\mathcal{A}$ can retrieve the possible value of $s^s = M_3^s \oplus ID \oplus r^s$. In key updating phase, another random value t^s is used. In order to retrieve the x^s by using equations in Equation 7.28, $\mathcal{A}$ needs to know the random value t^s. It is difficult and ineffective retrieving t^s because $T.(.)$ is a Chebyshev chaotic map and t^s and s^s are protected by Chebyshev chaotic map hard problem (7.6 and 7.7). As a result, $\mathcal{A}$ can win forward privacy game with the negligible probability.

Theorem 7.6. *The proposed protocol is not existential untraceable.*

Proof. An adversary $\mathcal{A}$ queries a tag T_0 two times. If the reader R does not perform successful authentication with T_0 between two queries of $\mathcal{A}$,the value of c_i becomes fixed for these two query. This means $\mathcal{A}$ is able to track the tag by querying it; in other words, the protocol is not existential untraceable.

7.3.7.3. De-synchronization Attacks. In our protocol, If an adversary $\mathcal{A}$ blocks the first message flow from the reader to the tag, the tag will not answer to the reader's query. This will not cause any security and privacy violation.

If $\mathcal{A}$ blocks the message flow from the tag to the reader, the reader will not get the tag's response. The same situation occurs when $\mathcal{A}$ queries the tag with a random

number which is not determined by $\mathcal{A}$. This will not cause any security and privacy violation.

If $\mathcal{A}$ blocks the second message flow from the reader to the tag, although the reader authenticated the tag, the tag will not authenticate the reader. This means the reader updates the secret values while the tag does not. This is called as de-synchronization attack. To prevent this attack, we store updated and outdated secret values in back-end server.

7.3.8. Performance Evaluation of Improved Protocol

RFID tags are limited devices in terms of computational cost and storage requirements. Therefore, when designing a protocol for RFID systems, we have to consider the low-cost implementation of cryptographic functions. In our protocol, tags performs PRNG, Chebyshev chaotic map, XOR and concatenation operations.

Chebyshev polynomials are used in key agreement protocols [195], password-based authentication protocols [196] and RFID authentication protocols [190, 191]. Chebyshev polynomials are implemented in smart cards [196] and low-cost RFID tags [190] in an energy-efficient manner.

In our improved protocol, the most costly operation carried out by the tag is the semi-group property (Definition 7.3) of enhanced Chebyshev chaotic maps (Definiton 7.5) like protocols in [190, 191], because the rest of operations carried out by the tag are lightweight such as XOR and concatenation. In the computation of $T_n(x)$, the number of steps required grows linearly with n. In [197], the computation is reduced to a logarithmic number of steps by considering the following observation:

$$\begin{aligned} T_{2n}(x) &= T_2(T_n(x)) \\ T_{2n+1}(x) &= 2.T_{n+1}(x).T_n(x) - x \end{aligned} \tag{7.29}$$

Table 7.6. Security and privacy comparison of protocols

Security Features	Yeh et al.'s Protocol [185]	Yoon et al.'s Protocol [186]	Cheng et al.'s Protocol [190]	Benssalah et al.'s Protocol [191]	Improved Protocol
Mutual Authentication	✓	✗	✗	✗	✓
Universal Untraceability	✗	✗	✗	✗	✓
Backward Untraceability	✗	✗	✗	✗	✓
Existential Untraceability	✗	✗	✗	✗	✗
De-synchronization Resistance	✓	✗	✗	✗	✓

The computation of Chebyshev polynomials can be re-organized by using the recursive relation.

$$\begin{aligned} T_0 &= 1 \\ T_1 &= x \\ T_n(x) &= \begin{cases} 2 \cdot T^2_{\frac{n}{2}}(x) - 1 \text{ n is even} \\ 2 \cdot T_{\frac{n-1}{2}}(x) \cdot T_{\frac{n+1}{2}}(x) - x \text{ n is odd} \end{cases} \end{aligned} \tag{7.30}$$

In [192], several methods have been proposed to reduce the computation time of $T_n(x)$. Furthermore, Chebyshev polynomials can be efficiently implemented with less logic gates requirements and low latency by using the trigonometric version $T_n(x) = cos(narccos(x))$ [191, 198, 199], thus making them implementable on low-cost devices such as RFID tags.

In Table 7.6, we compare the security features of our protocol with some previously proposed protocols. Our protocol provides all security features except existential untraceability. If we do not use c_i value in our protocol, we make it existential untraceable. However, our protocol requires a linear search for each authentication request in

Table 7.7. Performance comparison of protocols

Features	Yeh et al.'s Protocol [185]	Yoon et al.'s Protocol [186]	Cheng et al.'s Protocol [190]	Benssalah et al.'s Protocol [191]	Improved Protocol
Communication Rounds	5	5	5	5	5
Tag Computation	3S + 4H + 1R	1R + 6P	1R + 4T	1R + 4T	1R + 4T
Reader Computation	1R	1R + 1H	1R	1R + 2H	1R + 2H
Back-end Server Computation	3SRS	1R + 4P	1R + 2T	1R + 4T	1R + 4T
Key Search Complexity	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
Storage overhead	4m	4m	3m	4m	3m
Communication cost	4m	4m	3m	4m	3m
Crypto primitive	X,P	X,P	T,X,C	T,X,C	T,X,C

^{C} Concatenation

^{H} Hash function

^{m} message length

^{P} Pseudo random number generation

^{R} Random number generation

^{S} Modular squaring

^{SRS} Square root solving

^{T} Chebyshev polynomials

^{X} XOR

this case. In Table 7.7, we compare performance features. Our protocol does not need an extra computational task in order to provide security and privacy.

8. NEW PHYSICALLY UNCLONABLE FUNCTION-BASED RFID AUTHENTICATION PROTOCOLS

In the literature, there are many solutions that consider the security and privacy problems related to RFID technology. Most of these solutions provide privacy against weak adversaries that are not capable of corrupting tags [1], [26]. Ng et al. [34] showed that symmetric-key based RFID authentication protocols only provides narrow-forward privacy or wide-weak privacy. Another disadvantage of these solutions is that they are not scalable. They perform a linear search for every identification request. There are some proposals such as [51], [53], [54] and [55] that use tree data structure for the keys. The search complexity of these protocols is O(log n) time where n is the number of tags. However, Avoine et al. [58], [28] demonstrated that these protocols have serious security and privacy deficiencies.

In [1], Vaudenay presented an RFID privacy model. He defined eight privacy classes. Privacy classification is determined by defining some restrictions for adversaries. Vaudenay assumed that strong adversary that does not have any restriction for the oracle access should be used to analyze real-life RFID systems. Vaudenay also showed that strong privacy cannot be achieved without public-key cryptography. In the literature, there is no solution to demolish this proof. The second most powerful adversary in Vaudenay's model is destructive adversary. Destructive adversary can access corrupt oracle only once because corrupted tags become unusable. Destructive adversaries can be used to analyze real-life systems, if PUF enabled tags are used. Several authentication protocols that utilizes PUF in order to achieve destructive privacy have been proposed in [200], [201], [83]. However, these protocols require the search process on the server side. As the number of tags, the search complexity of these protocols increases in a linear or logarithmic.

In this chapter, we propose destructive private RFID authentication protocols. Our protocols do not need search operation on the server side to identify tags. They can be used for large scale RFID systems. They do not reveal any extra computation and communication requirements in the tag and reader side. Our solutions are based on the use of Physically Unclonable Functions (PUFs). They are destructive-private in the Vaudenay-Model. Our protocols use one master key shared by all tags. In [202], it is stated that a protocol using only one master key has constant-time identification, but no privacy/security as soon as one tag is compromised. Our protocols provide resistance against tag compromising attack by using PUFs as a secure storage to keep secrets of the tag. Furthermore, they have resistance to side-channel attacks in which an adversary with full side-channel capacity tries to access the master key. To the best of our knowledge, they are first to provide such a privacy level without lookup property.

8.1. Physically Unclonable Functions (PUFs)

Physically Unclonable Functions (PUFs) was initially built by Pappu [203] by using the random physical variations that can be found in various objects [204]. PUFs are embodied into a device physically [205]. When queried with a challenge c, the PUF generates a response r depending on both the physical properties of the object containing PUF and the challenge c [206, 207]. This dependency is generally called as challenge-response behaviour of the PUF. It is impossible to build two PUFs with the same challenge-response behaviour, because each PUF acquires a unique randomness property in the manufacturing process. A particular PUF instance produces slightly different responses for the same challenge c. Fuzzy Extractor [208] maps these slightly different responses to a unique response r. Two different PUF instances generate different responses for the same challenge with overwhelming probability. That means PUFs having the same logical circuits design produce different responses. A particular PUF instance should have the following properties: [204, 209]:

(i) Robustness: the difference between two separate responses of a particular PUF instance to the same challenge should be small.

(ii) Unclonability: the difference between expected responses of two different PUF instances to the same challenge should be sufficiently large.

(iii) Unpredictability: it is infeasible to predict the response of a particular PUF instance to an unknown challenge, even if a certain number of previous challenge-response pairs of the PUF instance can be obtained.

(iv) Tamper-evident: any unauthorized attempt to access the PUF instance changes its challenge-response behaviour.

In the literature, there are various PUF implementations. The most important implementations are delay-based PUFs, memory-based PUFs and coating PUFs. Delay-based PUFs are based on race conditions and frequency variations in integrated circuits [210–212]. Memory-based PUFs use the instability of volatile memory cells, such as SRAM, flip-flops and latches [213–215]. Coating PUFs are based on the capacitance between each couple of metal wires [216]. The properties and the most basic usage of delay-based PUFs and memory-based PUFs were analysed in [206, 209]. Katzenbeisser et al. [206] stated that SRAM PUFs seem to achieve all desired properties of a PUF.

PUFs are promising functions that are used for secret key storage [217], authentication [218] and binding software to hardware platforms [219]. Furthermore, PUFs can be integrated into cryptographic algorithms and remote attestation protocols [206]. The most important advantages of PUFs are uniqueness, physical unclonability, tamper evidence and small hardware requirements. A PUF construction does not need expensive hardware such as EEPROMs [220]. It can be implemented in hardware proportional to the number of challenge bits. PUFs can also be utilized to achieve security and privacy requirements of RFID systems. In the literature, there are some studies that presented PUF based RFID security protocols like in [200] and [201].

Several studies have already been made to implement PUFs on RFID tags. Devadas et al. [77] implemented a PUF in an RFID tag. They designed and fabricated RFID ICs with the silicon PUF circuit based on MUXes and an arbiter. The PUF has been implemented in less than $0.02mm^2$ and has been designed in 0.18 μ fabrication

technology. This PUF-enabled RFID IC operates at 13.56MHz. Devadas et al. also tested intra-PUF variation which is a measure of the reproducibility of responses from an individual PUF circuit and the inter-PUF variation is a measure of the uniqueness of an individual PUF circuit. The inter-PUF variation is high and the intra-PUF variation is low for their PUF implementation. These results are ideal for secure and reliable authentication. This PUF implementation meets the needs of our proposed protocol. Furthermore, Devadas et al. founded Verayo Inc. and developed the first commercial PUF embedded RFID tag [221] . Verayo provides PUF-based security products for authentication of products and anti-counterfeiting.

8.2. RFID Security and Privacy Model

In this section, we describe the general RFID security and privacy model in [1] with its extension proposed in [222].

In this model, the tag $\mathcal{T}$ is a restricted device in terms of power, memory and computation. It has a unique ID used for identification by the reader $\mathcal{R}$. The reader $\mathcal{R}$ has several transceivers and a back-end database. There is a secure communication between the transceiver and the back-end database. Tag identifiers and other information about tags are stored in the back-end database.

8.2.1. System Model

An RFID scheme is defined by the following procedures:

- $\mathsf{SetupReader}(1^s) \rightarrow (K_S, K_P)$ generates a public parameter K_P, a private parameter K_S and a security parameter s for the reader. It also generates a database in which identifiers of tags generated by SetupTag(ID) will be stored.
- $\mathsf{SetupTag}^{K_P}(\mathsf{ID}) \rightarrow (K, S)$ generates a tag having a unique identifier ID, a key K and an updateable memory states S. ID and K are stored in the back-end database if the tag is legitimate.

- $\mathsf{IdentTag} \rightarrow out$ is an interactive protocol between a $\mathcal{T} \in Tags$ and a $\mathcal{R} \in Readers$. At the end of the protocol, if the tag is not identified by the reader then $out = \perp$; otherwise $out = \mathsf{ID}$ where ID is the identifier of the tag.

8.2.2. Adversarial Model

There are three criterias that determine the features of an adversary $\mathcal{A}$: actions she is allowed to perform, the goal of the attack and how the attack is carried out.

At the beginning of each experiment, a challenger $\mathcal{C}$ executes the $\mathsf{SetupReader}(1^s)$ procedure. Thus, 1^s, K_S and K_P parameters are generated, and 1^s and K_P are given to $\mathcal{A}$. Furthermore, it is assumed that there is no tag in the system. $\mathcal{A}$ can create tags using the $\mathsf{CreateTag}^b(\mathsf{ID})$ oracle. In this model, tags are classified according to whether they are in the reading range of the adversary or not. If a tag is in the reading range of the adversary, it is considered a *drawn* tag. If a tag is not accessible by the adversary, it is considered a *free* tag.

The following oracles are defined to represent the abilities of the adversary.

- $\mathsf{CreateTag}^b(\mathsf{ID})$ creates a *free* tag with a unique identifier ID. This oracle sets up the tag with $\mathsf{SetupTag}^{K_P}(\mathsf{ID})$. If the tag is legitimate ($b = 1$), it is added to the database.
- $\mathsf{DrawTag}(\mathsf{distr}, \mathsf{n}) \rightarrow (\mathsf{vtag}_0, \mathsf{b}_0, ..., \mathsf{vtag}_{n-1}, \mathsf{b}_{n-1})$ randomly chooses n tags from the set of free tags with distribution probability *distr*. The status of chosen tags is changed from *free* to *drawn*. The oracle assigns virtual identifiers to the chosen tags and outputs these identifiers (vtag_0,...,vtag_{n-1}). The adversary can access to drawn tags only once because they have temporary virtual identifiers. The oracle outputs $\perp$ for already drawn or nonexistent tags. Furthermore, the oracle returns array of bits (b_0,...,b_{n-1}) telling whether drawn tags are legitimate or not. $\mathsf{DrawTag}$ oracle also keeps real identifiers and theirs associated virtual identifiers ($\mathsf{ID}_i, \mathsf{vtag}_i$) in a table Tab. All ID values and table Tab remain unknown to the adversary $\mathcal{A}$.

- Free(vtag) changes the status of a tag with virtual identifier vtag from *drawn* to *free*. Thus, the adversary cannot access to the tag no more.
- Launch() $\rightarrow \pi$ enables the reader to start a new IdentTag protocol instance. This oracle outputs the identifier π of this protocol instance.
- SendReader(m, π) $\rightarrow m'$ sends a message m to the reader in the protocol instance π. The reader responds with a message m'.
- SendTag(m, vtag) $\rightarrow m'$ sends a message m to the tag with virtual identifier vtag. The tag responds with a message m'.
- Execute(vtag) $\rightarrow (\pi, transcript)$ executes a complete protocol between a tag with virtual identifier vtag and the reader. The oracle starts by using Launch() query and continues with SendReader and SendTag queries. It outputs the list of successive messages of the protocol instance π.
- Result(π) $\rightarrow x$ returns either 1 if the reader identifies a legitimate tag, and 0 otherwise at the end of the protocol instance π.
- Timer(π) $\rightarrow \delta$ returns the time δ taken by the reader for its overall computations during the protocol instance π [222].
- Corrupt(vtag) $\rightarrow S$ gets the current state S of a tag with virtual identifier vtag. The adversary cannot use vtag because it is considered as destroyed.

8.2.3. Privacy Classes [1]

Vaudenay defines different classes of adversaries by putting some restrictions to the adversary in the use of oracles.

- *weak*: $\mathcal{A}$ cannot access to Corrupt(vtag) oracle.
- *forward*: $\mathcal{A}$ can only access more Corrupt(vtag) oracles after the first call of Corrupt(vtag) oracle.
- *destructive*: $\mathcal{A}$ cannot call any other oracles for *vtag* after she call Corrupt(vtag) oracle because the tag is destroyed and cannot be used again.
- *strong*: $\mathcal{A}$ has no restriction on accessing to all the oracles.

narrow adversaries cannot access to $\mathsf{Result}(\pi)$. In [222], Avoine et al. introduced the notion of time and formalized it by extending Vaudenay's model. $\mathcal{A}$ accessing the Timer oracle, tries to deduce anything about a tag identity by using the time that the reader has spent in order to identify the tag. *timeful* adversaries can access to $\mathsf{Timer}(\pi)$ oracle. Privacy notions and their implications are summarized in Figure 8.1.

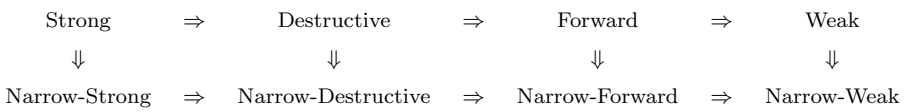

Figure 8.1. Implications of privacy notions

8.2.4. Security Properties

Definition 8.1 (Security [1])**.** *A scheme provides security if it provides secure tag authentication and reader authentication.*

8.2.5. Privacy

In this model, privacy is determined by using a simulator $\mathcal{B}$, called blinder. $\mathcal{B}$ can simulate a real RFID system without knowing any secrets. In privacy experiment described in Figure 8.2, an adversary $\mathcal{A}$ tries to distinguish the blinder $\mathcal{B}$ from the real RFID system.

Definition 8.2 (Blinder, trivial adversary [1])**.** *A Blinder $\mathcal{B}$ is a polynomial-time algorithm which simulates* $\mathsf{Launch}()$*,* $\mathsf{SendReader}(\mathsf{m}, \pi)$*,* $\mathsf{SendTag}(\mathsf{m}, \mathsf{vtag})$*,* $\mathsf{Result}(\pi)$ *and* $\mathsf{Timer}(\pi)$ *to an adversary $\mathcal{A}$ without knowing any secret. A blinded adversary $\mathcal{A}^{\mathcal{B}}$ is an adversary who does not use* $\mathsf{Launch}()$*,* $\mathsf{SendReader}(\mathsf{m}, \pi)$*,* $\mathsf{SendTag}(\mathsf{m}, \mathsf{vtag})$*,* $\mathsf{Result}(\pi)$ *and* $\mathsf{Timer}(\pi)$ *oracles. An adversary $\mathcal{A}$ is trivial if there exist a $\mathcal{B}$ such that the advantage of $\mathcal{A}$:*

$$|Pr[\mathcal{A} \ wins] - Pr[\mathcal{A}^{\mathcal{B}} \ wins]|$$

is negligible.

Definition 8.3 (Privacy [1]). *An RFID scheme is said to be C-private if all the adversaries which belong to class C and that interact with the whole system using oracles are trivial (Definition 8.2).*

Privacy experiment $\mathrm{Exp}_{\mathcal{A}_P}^{Vaud-priv}$:

(i) A challenger $\mathcal{C}$ executes $\mathsf{SetupReader}(1^s)$ procedure generates 1^s, K_S and K_P parameters and sends 1^s and K_P to $\mathcal{A}_P$.

(ii) $\mathcal{A}_P$ interacts with the RFID system according to limitations on class P.

(iii) $\mathcal{A}_P$ analyzes system without using oracle queries.

(iv) $\mathcal{A}_P$ submits his hypothesis and receives the secret table Tab of the $\mathsf{DrawTag}$ oracle.

(v) $\mathcal{A}_P$ returns a bit b'. $b' = 1$ if her hypothesis is correct and 0 otherwise.

Figure 8.2. Privacy experiment

8.3. Definitions

Definition 8.4 (Hash Function). *Let $l \in \mathbf{N}$ be a security parameter, $\gamma, \kappa \in \mathbf{N}$ be polynomially bounded in l. A hash function H is defined as $\{0,1\}^\gamma \to \{0,1\}^\kappa$ with the following basic requirements:*

(i) For a given output y_i, it is computationally infeasible to find a input x_i satisfying $H(x_i) = y_i$.

(ii) It is computationally infeasible to find a pair (x_i, x_j) satisfying $x_i \neq x_j$ and $H(x_i) = H(x_j)$.

(iii) Any probabilistic polynomial time adversary who queried H for a polynomial number of times can distinguish the output of H with at most negligible probability.

Definition 8.5 (Physically Unclonable Function (PUF) [201]). *Let $l \in \mathbf{N}$ be a security parameter, $\gamma, \kappa \in \mathbf{N}$ be polynomially bounded in l. An ideal PUF P is defined as $\{0,1\}^\gamma \to \{0,1\}^\kappa$ that has the following parameters:*

(i) For a pair $(c_i, c_j) \in \{0,1\}^\gamma$, $P(c_i) = r_i$ and $P(c_j) = r_j$. If $c_i = c_j$, then the probability $Pr[r_i = r_j] = 1$.

(ii) Any physical attempt to tamper the device on which P is implemented results in destruction of P. Thus P cannot be evaluated any more.

(iii) Any probabilistic polynomial time adversary who queried P for a polynomial number of times can compute the output of P with at most negligible probability.

8.4. A New Scalable RFID Authentication Protocol I

In this section, we propose scalable destructive-private RFID authentication protocol. Our protocol is improved version of the protocol in [53]. Our protocol achieves destructive-privacy with the help of a physically unclonable functions. Each tag has unique PUF P. Each tag has a random seed value S and stores the values that are obtained by XORing keys with the value that is derived by evaluating the PUF P with input S. Our protocol is summarized in Figure 8.3.

Table 8.1. Notations of scalable RFID authentication protocol I

Notation	Description
T_i	The i-th Tag.
$k_{j,k}$	The k-th key on the j-th level of key tree.
S	The random seed value for P.
l	The bit-length of secrets and random values.
f	A pseudo-random function $\{0,1\} \rightarrow \{0,1\}^l$.
P	PUF $\{0,1\} \rightarrow \{0,1\}^l$.
$\oplus$	XOR operator.
$\in$	The random choice operator.

8.4.1. Initialization

(i) The server chooses random $S_i \in \{0,1\}^l$ for the tag T_i.

(ii) The tag T_i stores S_i and $k'_{j,k} = k_{j,k} \oplus P(S)$ for each key $k_{j,k}$ that are associated with the tag T_i.

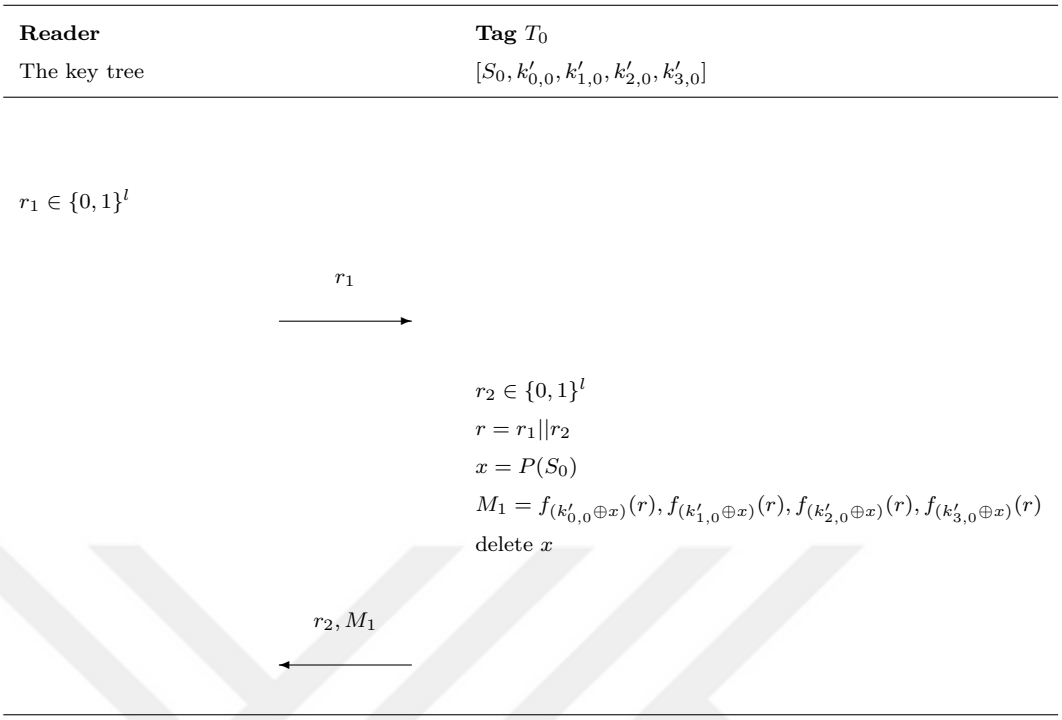

Figure 8.3. Scalable RFID authentication protocol I

8.4.2. Authentication

(i) Reader: Generates random bit string $r_1 \in \{0,1\}^l$ and sends r_1 to the tag T_i.

(ii) Tag: T_i generates random bit string $r_2 \in \{0,1\}^l$, computes $r = r_1||r_2$, $x = P(S)$ and $M_1 = f_{(k'_{0,0}\oplus x)}(r), f_{(k'_{1,0}\oplus x)}(r), f_{(k'_{2,0}\oplus x)}(r), f_{(k'_{3,0}\oplus x)}(r)$.

(iii) Tag: T_i deletes x.

(iv) Tag: T_i sends (r_2, M_1) to the reader.

(v) Reader: Checks each value in M_1 by traversing the key tree from the root to the leaves.

8.4.3. Security and Privacy Analysis

The security of our protocol relies on PUFs. An adversary compromising a tag cannot obtain the secrets of the tag so she cannot obtain the secrets of other tags. As a result, we eliminate the main vulnerability of tree based hash protocols.

Theorem 8.1. *The proposed protocol provides tag authentication if f is a pseudo-random function and P is a PUF.*

Proof. We assume that there is an adversary $\mathcal{A}$ that can generate (r_2, M_1) for a given r_1 with non-negligible probability. In Vaudenay-Model, $\mathcal{A}$ is not allowed to use Corrupt and SendTag queries to exclude trivial attacks. $\mathcal{A}$ try to impersonate the target tag T_i to the legitimate reader without knowing the secrets of T_i. As a result, $\mathcal{A}$ wins the security experiment, if the reader returns ID of the target tag T_i.

The adversary $\mathcal{A}$ has to simulate f to generate (r_2, M_1) pair for a given r_1 correctly. This will contradict the pseudo-randomness of f.

Definition 8.6 (Destructive-Privacy)**.**

Phase 1: (Learning)

a. $\mathcal{A}$ gets access a number of tags by calling DrawTag *oracle query.*

b. $\mathcal{A}$ is able to send any oracle queries including Corrupt *oracle to tags.*

c. $\mathcal{A}$ frees chosen tags by calling Free *oracle query.*

Phase 2: (Challenge)

a. $\mathcal{A}$ chooses two uncorrupted tags vtag_i *and* vtag_j *as its challenge candidates.*

b. $\mathcal{A}$ gets access to one of these two tags by calling $\mathsf{DrawTag}(1/2, 1)$ *oracle query and gets the fresh identifier of tag* vtag_b *where* $b \in \{i, j\}$.

c. $\mathcal{A}$ calls all oracle queries on vtag_b*, except* Corrupt *oracle.*

d. $\mathcal{A}$ frees a chosen tag by calling $\mathsf{Free}(\mathsf{vtag_b})$ *oracle query.*

Phase 3: (Guess)

a. Eventually, $\mathcal{A}$ terminates the game simulation and outputs a bit b' which is its guess of the value of b. $\mathcal{A}$ wins if $b' = b$.

Theorem 8.2. *The protocol depicted in Figure 8.3 is destructive-private in the Vaudenay-Model.*

Proof. We assume that there is an adversary $\mathcal{A}$ that wins **Destructive-Privacy-Game** with non-negligible probability. We build a blinded adversary $\mathcal{A}^B$ that simulates **Destructive-Privacy-Game**. $\mathcal{A}^B$ starts game by accessing a number of tags. $\mathcal{A}^B$ can send any oracle queries to tags except Corrupt oracle query. This is the first

contradiction. In the learning phase, $\mathcal{A}^B$ cannot learn keys that are still used by other tags by compromising a tag.

$\mathcal{A}^B$ can continue the game. In the challenge phase, $\mathcal{A}^B$ chooses two uncorrupted tags. $\mathcal{A}^B$ gets access to one of them and calls any oracle queries, except Corrupt oracle. $\mathcal{A}^B$ has to distinguish responses of vtag_b. $\mathcal{A}^B$ can do this either if she knows the secrets of vtag_b or f is not pseudo-random. We know that $\mathcal{A}^B$ does not know the secrets of any tag. Therefore, f cannot be pseudo-random. This will contradict the pseudo-randomness of f.

8.5. A New Scalable RFID Authentication Protocol II

8.5.1. Notations

Table 8.2 gives the notations used in describing the proposed protocol. The $C(A, B)$ function generates a permutation vector p from the second input B. For example, B can be used as a key for the key scheduling algorithm of RC4 to initialize the permutation vector p. After that, it permutes A according to p.

8.5.2. Protocol Description

8.5.2.1. Initialization Phase.

(i) Two random master keys S_1 and S_2 are generated and assigned to the back-end server $\mathcal{BS}$.

$$\begin{aligned} S_1 &\in \{0,1\}^l \\ S_2 &\in \{0,1\}^l \\ \{S_1, S_2\} &\Rightarrow \mathcal{BS} \end{aligned} \tag{8.1}$$

Table 8.2. Notations of scalable RFID authentication protocol II

Notation	Description
S_1	The shared secret 1
S_2	The shared secret 2
ID_i	The identifier of a tag T_i
$DATA_i$	Data about a tag T_i
$(a, b, c, d, e, f)_i$	Secret values of a tag T_i
H	A hash function $\{0,1\}^l \times \{0,1\}^l \times \{0,1\}^l \rightarrow \{0,1\}^l$
P_i	The PUF $\{0,1\}^l \rightarrow \{0,1\}^l$ of a tag T_i
$C(A, B)$	A permutation function that computes the permutation of $A = \{0,1\}^l$ according to permutation vector which is generated from $B = \{0,1\}^l$
$\oplus$	XOR operator
$\Rightarrow$	Assignment operator
$\longrightarrow$	Sending over a channel operator
$\in$	Random choice operator

(ii) Four random unique keys a, b, d, e are generated for each tag. These keys are written on the tag $\mathcal{T}$.

$$\begin{aligned} a &\in \{0,1\}^l \\ b &\in \{0,1\}^l \\ d &\in \{0,1\}^l \\ e &\in \{0,1\}^l \\ \{a, b, d, e\} &\Rightarrow \mathcal{T} \end{aligned} \tag{8.2}$$

(iii) Two keys c and f are computed for each tag using their own embedded PUF $P(.)$. These keys are written on the tag.

$$\begin{aligned} c &= S_1 \oplus P(a) \oplus P(b) \\ f &= S_2 \oplus P(d) \oplus P(e) \\ \{c, f\} &\Rightarrow \mathcal{T} \end{aligned} \tag{8.3}$$

(iv) The back-end server $\mathcal{BS}$ stores $[ID, DATA]$ for each tag.

8.5.2.2. Authentication Phase.

(i) The reader $\mathcal{R}$ creates a nonce r_1 and sends it to a tag $\mathcal{T}_i$.

$$\begin{aligned} r_1 &\in \{0,1\}^l \\ r_1 &\longrightarrow \mathcal{T}_i \end{aligned} \tag{8.4}$$

(ii) The tag $\mathcal{T}_i$ creates a nonce r_2.

$$r_2 \in \{0,1\}^l \tag{8.5}$$

$\mathcal{T}_i$ calculates the permutation key p_1.

$$p_1 = H(r_1, r_2, 1) \tag{8.6}$$

$\mathcal{T}_i$ calculates the first part of a session key $H(P_i(a_i), r_1, r_2)$ and hides it with the permutation of $P_i(a_i)$ according to the permutation key p_1. $P_i(a_i)$ is deleted from the volatile memory.

$$\begin{aligned} tmp_1 &= H(P_i(a_i), r_1, r_2) \oplus C(P_i(a_i), p_1) \\ &\text{delete } P_i(a_i) \end{aligned} \tag{8.7}$$

Reader $S_1, S_2, [ID_i, DATA_i]$		**Tag** T_i $ID_i, a_i, b_i, c_i, d_i, e_i, f_i$
$r_1 \in \{0,1\}^l$		
	$r_1 \longrightarrow$	
		$r_2 \in \{0,1\}^l$
		$p_1 = H(r_1, r_2, 1)$
		$tmp_1 = H(P_i(a_i), r_1, r_2) \oplus C(P_i(a_i), p_1)$
		delete $P_i(a_i)$
		$tmp_2 = H(P_i(b_i), r_1, r_2) \oplus C(P_i(b_i), p_1)$
		delete $P_i(b_i)$
		$M_1 = tmp_1 \oplus tmp_2 \oplus C(c_i, p_1)$
	$\longleftarrow r_2, M_1$	
$r_3 \in \{0,1\}^l$		
$p_1 = H(r_1, r_2, 1)$		
$p_2 = H(r_2, r_3, 2)$		
$p_3 = H(r_3, r_2, 3)$		
$tmp_3 = M_1 \oplus C(S_1, p_1)$		
$tmp_4 = C(tmp_3, p_2) \oplus C(S_2, p_3)$		
$M_2 = H(tmp_4, r_2, r_3)$		
	$p_2, p_3, r_3, M_2 \longrightarrow$	
		$tmp'_4 = C(H(P_i(a_i), r_1, r_2), p_2)$
		delete $P_i(a_i)$
		$tmp'_4 = tmp'_4 \oplus C(P_i(d_i), p_3)$
		delete $P_i(d_i)$
		$tmp'_4 = tmp'_4 \oplus C(H(P_i(b_i), r_1, r_2), p_2)$
		delete $P_i(b_i)$
		$tmp'_4 = tmp'_4 \oplus C(P_i(e_i), p_3)$
		delete $P_i(e_i)$
		$tmp'_4 = tmp'_4 \oplus C(f_i, p_3)$
		$if(M_2 = H(tmp'_4, r_2, r_3))$
		$r_4 \in \{0,1\}^l$
		$M_3 = H(tmp'_4, r_3, r_4) \oplus ID_i$
		$M_4 = H(ID_i, r_4, r_3)$
		else
		$\perp$
	$\longleftarrow r_4, M_3, M_4$	
$ID'_i = H(tmp_4, r_3, r_4) \oplus M_3$		
$if(M_4 \neq H(ID'_i, r_4, r_3))$		
$\perp$		

Figure 8.4. Scalable RFID authentication protocol II

$\mathcal{T}_i$ calculates the second part of a session key $H(P_i(b_i), r_1, r_2)$ and hides it with the permutation of $P_i(b_i)$ according to the permutation key p_1. $P_i(b_i)$ is deleted from the volatile memory.

$$\begin{aligned} tmp_2 &= H(P_i(b_i), r_1, r_2) \oplus C(P_i(b_i), p_1) \\ &\text{delete } P_i(b_i) \end{aligned} \tag{8.8}$$

$\mathcal{T}_i$ calculates M_1 by XORing tmp_1, tmp_2 and the permutation of c_i according to the permutation key p_1. tmp_1 and tmp_2 are deleted from the volatile memory. As a result, the tag $\mathcal{T}_i$ obtains M_1 which is equal to XOR of the session key $H(P_i(a_i), r_1, r_2) \oplus H(P_i(b_i), r_1, r_2)$ and the permutation of the master key S_1 according to the permutation key p_1. An adversary corrupting the tag $\mathcal{T}_i$ while $\mathcal{T}_i$ is calculating M_1 cannot obtain the master key S_1.

$$\begin{aligned} M_1 &= tmp_1 \oplus tmp_2 \oplus C(c_i, p_1) \\ &= H(P_i(a_i), r_1, r_2) \oplus H(P_i(b_i), r_1, r_2) \oplus C(S_1, p_1) \end{aligned} \tag{8.9}$$

$\mathcal{T}_i$ sends r_2 and M_1 to the reader $\mathcal{R}$.

$$\{r_2, M_1\} \longrightarrow \mathcal{R} \tag{8.10}$$

(iii) The reader $\mathcal{R}$ creates a nonce r_3, calculates three permutation keys and calculates the session key.

$$\begin{aligned} r_3 &\in \{0,1\}^l \\ p_1 &= H(r_1, r_2, 1) \\ p_2 &= H(r_2, r_3, 2) \\ p_3 &= H(r_3, r_2, 3) \\ tmp_3 &= M_1 \oplus C(S_1, p_1) \\ &= H(P_i(a_i), r_1, r_2) \oplus H(P_i(b_i), r_1, r_2) \end{aligned} \tag{8.11}$$

$\mathcal{R}$ calculates the permutation of the session key according to the permutation key p_2 and hides it with the permutation of the master key S_2 according to the permutation key p_3.

$$tmp_4 = C(tmp_3, p_2) \oplus C(S_2, p_3) \tag{8.12}$$

$\mathcal{R}$ calculates the hash of tmp_4 and sends p_2, p_3, r_3 and M_2 to the tag $\mathcal{T}_i$.

$$\begin{aligned} M_2 &= H(tmp_4, r_2, r_3) \\ \{p_2, p_3, r_3, M_2\} &\longrightarrow \mathcal{T}_i \end{aligned} \tag{8.13}$$

(iv) The tag $\mathcal{T}_i$ calculates the permutation of the first part of the session key according to the permutation key p_2. $P_i(a_i)$ is deleted from the volatile memory.

$$\begin{aligned} tmp'_4 &= C(H(P_i(a_i), r_1, r_2), p_2) \\ &\text{delete } P_i(a_i) \end{aligned} \tag{8.14}$$

$\mathcal{T}_i$ calculates the permutation of the first part of the master key S_2 according to the permutation key p_3 and XORes it with tmp'_4. $P_i(d_i)$ is deleted from the volatile memory.

$$\begin{aligned} tmp'_4 &= tmp'_4 \oplus C(P_i(d_i), p_3) \\ &\text{delete } P_i(d_i) \end{aligned} \tag{8.15}$$

$\mathcal{T}_i$ calculates the permutation of the second part of the session key according to the permutation key p_2 and XORes it with tmp'_4. $P_i(b_i)$ is deleted from the volatile memory.

$$\begin{aligned} tmp'_4 &= tmp'_4 \oplus C(H(P_i(b_i), r_1, r_2), p_2) \\ &\text{delete } P_i(b_i) \end{aligned} \tag{8.16}$$

$\mathcal{T}_i$ calculates the permutation of the second part of the master key S_2 according to the permutation key p_3 and XORes it with tmp'_4. $P_i(e_i)$ is deleted from the volatile memory.

$$\begin{aligned} tmp'_4 &= tmp'_4 \oplus C(P_i(e_i), p_3) \\ &\text{delete } P_i(e_i) \end{aligned} \tag{8.17}$$

$\mathcal{T}_i$ calculates the permutation of the third part of the master key S_2 according to the permutation key p_3 and XORes it with tmp'_4. As a result, the tag obtains tmp'_4 which is equal to XOR of the permutation of the session key $H(P_i(a_i), r_1, r_2) \oplus H(P_i(b_i), r_1, r_2)$ according to the permutation key p_2 and the permutation of the master key S_2 according to the permutation key p_3. An adversary corrupting the tag $\mathcal{T}_i$ while $\mathcal{T}_i$ is calculating tmp'_4 cannot obtain the master key S_1 or S_2.

$$\begin{aligned} tmp'_4 &= tmp'_4 \oplus C(f_i, p_3) \\ &= C(H(P_i(a_i), r_1, r_2) \oplus H(P_i(b_i), r_1, r_2), p_2) \oplus C(S_2, p_3) \end{aligned} \tag{8.18}$$

The tag checks the validity of M_2 by computing $H(tmp'_4, r_2, r_3)$ in order to authenticate the reader.

$$M_2 \stackrel{?}{=} H(tmp'_4, r_2, r_3) \tag{8.19}$$

If the reader is authenticated, the tag creates a nonce r_4 and computes $M_3 = H(tmp'_4, r_3, r_4) \oplus ID_i$ and $M_4 = H(ID_i, r_4, r_3)$. The tag sends r_4, M_3 and M_4 to the reader.

$$\begin{aligned} r_4 &\in \{0,1\}^l \\ M_3 &= H(tmp'_4, r_3, r_4) \oplus ID_i \\ M_4 &= H(ID_i, r_4, r_3) \\ \{r_4, M_3, M_4\} &\longrightarrow \mathcal{R} \end{aligned} \tag{8.20}$$

(v) The reader decrypts M_3 and learns ID_i' identifier of the tag $\mathcal{T}_i$. It compares $H(ID_i', r_3, r_4)$ with M_4. If they are equal, the tag is identified.

$$\begin{aligned} ID_i' &= H(tmp_4, r_3, r_4) \oplus M_3 \\ M_4 &\stackrel{?}{=} H(ID_i', r_4, r_3) \end{aligned} \tag{8.21}$$

8.5.3. Security Analysis

Lemma 8.3. *Let $\mathcal{A}$ be a destructive adversary. The advantage of $\mathcal{A}$ in obtaining the master keys S_1 and S_2 by corrupting a tag is negligible.*

Proof. There are six deletion operations in the proposed protocol. We consider each of them one by one to determine $\mathcal{A}$'s advantage. Assume $\mathcal{A}$ corrupts a tag T_i before the first deletion. $\mathcal{A}$ obtains $P_i(a_i)$ and tmp_1. To generate the master key S_1, $\mathcal{A}$ has to obtain the secret value $P_i(b_i)$. Thus, $\mathcal{A}$ has to simulate $P(.)$. Assume $\mathcal{A}$ corrupts a tag T_i before the second deletion. $\mathcal{A}$ obtains $P_i(b_i)$, tmp_1 and tmp_2. To generate the master key S_1, $\mathcal{A}$ has to obtain the secret value $P_i(a_i)$. Thus, $\mathcal{A}$ has to clone or simulate $P_i(.)$ or has to solve tmp_1. The same scenario is valid for the remaining four deletion operations. $\mathcal{A}$ can obtain one secret value created with $P_i(.)$ at a time. Thus, $\mathcal{A}$ has to simulate $P(.)$ or $\mathcal{A}$ has to generate an input for a given hash output. As a result, $\mathcal{A}$ can learn S_1 or S_2 by calling Corrupt oracle with negligible probability.

Theorem 8.4. *The proposed protocol provides tag authentication if H is a hash function (Definition 8.4) and P is a PUF (Definition 8.5).*

Proof. We assume that there is an adversary $\mathcal{A}$ that can generate (r_2, M_1) for a given r_1 and (r_4, M_3, M_4) for a given (p_2, p_3, r_3, M_2) with non-negligible probability. $\mathcal{A}$ knowing or not knowing the secrets of a tag T_i tries to impersonate it to the legitimate reader. As a result, A wins the security experiment if the reader returns the identifier of the target tag T_i.

Let $\mathcal{A}$ have access to all oracles. $\mathcal{A}$ can corrupt a tag T_i and learn ID_i, a_i, b_i, c_i, d_i, e_i, f_i. She has to simulate $P_i(.)$ to correctly answer the reader's queries. This contradicts the unclonability of $P_i(.)$. Kardaş et. al [134] showed that a secret value created with a PUF can be learned by $\mathcal{A}$ if the corruption occurs in the time period in which the secret is in volatile memory. In our protocol, $\mathcal{A}$ can obtain just one secret ($P_i(a)$ or $P_i(b)$ and $P_i(d)$ or $P_i(e)$) created using $P_i(.)$. Thus, $\mathcal{A}$ cannot create valid responses without knowing other secrets created with $P_i(.)$.

Let $\mathcal{A}$ have access to all oracles except for the Corrupt oracle. $\mathcal{A}$ obtains n_1 protocol transcripts between a tag T_i and the reader by using the Execute oracle. Furthermore, $\mathcal{A}$ queries the tag T_i n_2 times using the SendReader oracle. $\mathcal{A}$ obtains a total of $N = n_1 + n_2$ responses generated by the tag T_i where N is polynomially bounded. $\mathcal{A}$ has to generate a pair (r_2, M_1) to impersonate a tag T_i. M_1 is generated by computing $H(P_i(a_i), r_1, r_2)$ and $H(P_i(b_i), r_1, r_2)$ where r_1 comes from the reader and $P_i(a_i)$ and $P_i(b_i)$ are not known by the adversary. We know that $H(.)$ is a random-like function (Definition 8.4). As a result, $\mathcal{A}$ can use these previous responses with the negligible probability $2^{1-l}N$ where l is the security parameter (the bit length of random nonces and messages).

Theorem 8.5. *The proposed protocol provides reader authentication if H is a hash function (Definition 8.4) and P is a PUF (Definition 8.5).*

Proof. We assume that there is an adversary $\mathcal{A}$ that can generate (p_2, p_3, r_3, M_2) for a given (r_2, M_1) with non-negligible probability. $\mathcal{A}$ knowing or not knowing secrets of a tag T_i tries to impersonate the legitimate reader R to the tag T_i. As a result, $\mathcal{A}$ wins the security experiment if the tag T_i sends (r_4, M_3, M_4) to $\mathcal{A}$.

Let $\mathcal{A}$ have access to all oracles. $\mathcal{A}$ can corrupt a tag T_i and learn ID_i, a_i, b_i, c_i, d_i, e_i, f_i. She has to simulate $P_i(.)$ to correctly generate M_2. This contradicts the unclonability of $P_i(.)$.

Let $\mathcal{A}$ have access to all oracles except for the Corrupt oracle. $\mathcal{A}$ obtains n_1 protocol transcripts between a tag T_i and the reader R by using the Execute oracle. Furthermore, $\mathcal{A}$ queries the reader R n_2 times using the SendTag oracle. $\mathcal{A}$ obtains a total of $N = n_1 + n_2$ responses generated by the reader R where N is polynomially bounded. $\mathcal{A}$ has to generate a pair (p_2, p_3, r_3, M_2) to impersonate the reader R. She has to know S_1 and S_2 to generate M_2. $\mathcal{A}$ can use these previous responses with the negligible probability $2^{1-l}N$ where l is the security parameter (the bit length of random nonces and messages). This contradicts our assumption.

8.5.4. Privacy Analysis

Theorem 8.6. *The protocol depicted in Figure 8.4 achieves timeful-destructive privacy (Definition 8.3) if the protocol achieves tag authentication, P is PUF (Definition 8.5) and H is Hash Function (Definition 8.4).*

Proof. We assume that there is an adversary $\mathcal{A}$ that can distinguish oracles simulated by a blinder $\mathcal{B}$ from the real oracles with non-negligible probability.

We first show how $\mathcal{B}$ simulates oracles.

- Launch() The simulation of Launch is trivial.
- SendTag(r_1, vtag) Returns $r_2 \in \{0,1\}^l$ and $M_1 \in \{0,1\}^l$.
- SendTag($\mathsf{p}_2, \mathsf{p}_3, \mathsf{r}_3, \mathsf{M}_2$, vtag) Returns $r_4 \in \{0,1\}^l$, $M_3 \in \{0,1\}^l$ and $M_4 \in \{0,1\}^l$.
- SendReader(π) Returns $r_1 \in \{0,1\}^l$
- SendReader($(\mathsf{r}_2, \mathsf{M}_1), \pi$) Returns $p_2 \in \{0,1\}^l$, $p_3 \in \{0,1\}^l$, $r_3 \in \{0,1\}^l$ and $M_2 \in \{0,1\}^l$.
- SendReader($(\mathsf{r}_4, \mathsf{M}_3, \mathsf{M}_4), \pi$) $\mathcal{B}$ does not need to simulate this query because it does not produce any output.
- Result(π) Returns 1 if π has been generated with Launch oracle and the corresponding protocol transcript has been generated with the real SendTag and SendReader oracles and 0 otherwise.

- $\mathsf{Timer}(\pi)$ Returns the constant time required for carrying out operations on the reader side.

Assume that there is a blinder $\mathcal{B}_0$ whose simulation is equivalent to real oracles. We construct a new blinder $\mathcal{B}_1$ from $\mathcal{B}_0$. The differences between $\mathcal{B}_1$ and $\mathcal{B}_0$ are that the states of all tags are simulated with randomly chosen values and the simulation of the $\mathsf{SendTag}$ oracle. For example, $\mathcal{B}_1$ assigns random values $P_i(a_i) \in \{0,1\}^l$, $P_i(b_i) \in \{0,1\}^l$, $c_i \in \{0,1\}^l$, $P_i(d_i) \in \{0,1\}^l$, $P_i(e_i) \in \{0,1\}^l$ and $f_i \in \{0,1\}^l$ for a tag T_i. $\mathcal{B}_1$ simulates the $\mathsf{SendTag}$ oracle by evaluating $H(.)$ and $C(.)$ with randomly assigned values. When $\mathcal{A}$ wants to use the $\mathsf{SendTag}$ oracle, the challenger $\mathcal{C}$ either evaluates the $\mathsf{SendTag}$ with real values as in $\mathcal{B}_0$ or with random values as in $\mathcal{B}_1$. During the attack time, $\mathcal{A}$ can use the $\mathsf{Corrupt}(\mathsf{vtag})$ to obtain the state of the tag $vtag$. However, $\mathcal{A}$ cannot obtain any secret from the tag $vtag$ (Lemma 8.3) and the tag $vtag$ cannot be used further (Definition 8.5). After a polynomial number of oracle queries, $\mathcal{A}$ can distinguish $\mathcal{B}_1$ from $\mathcal{B}_0$, which means $\mathcal{A}$ can distinguish the output of a PUF from a randomly chosen value with non-negligible probability. This statement contradicts the security property of the PUF (Definition 8.5). As a result $|Pr[\mathcal{A}^{B_0} \text{ wins}] - Pr[\mathcal{A}^{B_1} \text{ wins}]|$ is negligible.

We construct a new blinder $\mathcal{B}_2$ from $\mathcal{B}_1$. The only difference between $\mathcal{B}_2$ and $\mathcal{B}_1$ is that the $\mathsf{SendTag}$ oracle is simulated as described above. When $\mathcal{A}$ attempts to use the $\mathsf{SendTag}$ oracle, the challenger $\mathcal{C}$ either evaluates the $\mathsf{SendTag}$ as in $\mathcal{B}_1$ or sends random values as in $\mathcal{B}_2$. After a polynomial number of oracle queries, $\mathcal{A}$ can distinguish $\mathcal{B}_2$ from $\mathcal{B}_1$, which means $\mathcal{A}$ can distinguish the output of a hash function from a randomly chosen value with non-negligible probability. To accomplish this goal, $\mathcal{A}$ must solve the output of the hash function. This requirements contradicts the security property of the hash function (Definition 8.4). As a result $|Pr[\mathcal{A}^{B_1} \text{ wins}] - Pr[\mathcal{A}^{B_2} \text{ wins}]|$ is negligible.

We construct a new blinder $\mathcal{B}_3$ from $\mathcal{B}_2$. The only difference between $\mathcal{B}_3$ and $\mathcal{B}_2$ is that the Result oracle is simulated as described above. When $\mathcal{A}$ attempts to use the Result oracle, the challenger $\mathcal{C}$ either evaluates the real Result as in $\mathcal{B}_2$ or the

simulated Result as in $\mathcal{B}_3$. After a polynomial times of oracle queries, $\mathcal{A}$ can distinguish $\mathcal{B}_3$ from $\mathcal{B}_2$. That means $\mathcal{A}$ runs a protocol instance π and the simulated Result oracle returns a different output than the real Result oracle. This can only happen when $\mathcal{A}$ generates a protocol transcript that causes the real Result oracle to return 1. Theorem 8.4 says that this result can only happen with negligible probability. As a result $|Pr[\mathcal{A}^{B_2} \text{ wins}] - Pr[\mathcal{A}^{B_3} \text{ wins}]|$ is negligible.

We construct a new blinder $\mathcal{B}_4$ from $\mathcal{B}_3$. The only difference between $\mathcal{B}_4$ and $\mathcal{B}_3$ is that the Timer oracle is simulated as described above. When $\mathcal{A}$ wants to use the Timer oracle, the challenger $\mathcal{C}$ either evaluates the real Timer as in $\mathcal{B}_3$ or the simulated Timer as in $\mathcal{B}_4$. After a polynomial number of oracle queries, $\mathcal{A}$ can distinguish $\mathcal{B}_4$ from $\mathcal{B}_3$, which means that $\mathcal{A}$ runs a protocol instance π and the simulated Timer oracle returns a different output than the real Timer oracle. We know that the simulation of the Timer oracle is perfect and that the reader performs a fixed number of operations for tag identification. Both the real and simulated Timer oracles always return the same result. As a result $|Pr[\mathcal{A}^{B_3} \text{ wins}] - Pr[\mathcal{A}^{B_4} \text{ wins}]|$ is negligible.

The full proof shows that $|Pr[\mathcal{A}^{B_0} \text{ wins}] - Pr[\mathcal{A}^{B_4} \text{ wins}]|$ is negligible. The blinder $\mathcal{B}_4$ is equal to the full blinder $\mathcal{B}$ and the blinder $\mathcal{B}_0$ is equal to the real oracles, which means that $\mathcal{A}$ cannot distinguish oracles simulated by the blinder $\mathcal{B}$ from the real oracles with non-negligible probability.

8.6. A New Scalable RFID Authentication Protocol III

The proposed protocol has two phases: initialization and authentication. Table 8.3 gives the notations used in describing the proposed protocol.

8.6.1. Initialization Phase

Two random keys S_1 and S_2 are generated for the back-end server. Four random unique keys a, b, d, e generated for each tag. Then, keys $c = S_1 \oplus P(a) \oplus P(b)$ and

Table 8.3. Notations of scalable RFID authentication protocol III

Notation	Description
S_1	The master secret 1
S_2	The master secret 2
ID_i	The identifier of a tag T_i
$DATA_i$	Information about a tag T_i
$(a, b, c, d, e, f)_i$	Secret values of a tag T_i
H	A hash function $\{0,1\}^l \times \{0,1\}^l \times \{0,1\}^l \rightarrow \{0,1\}^l$
P_i	The PUF $\{0,1\}^l \rightarrow \{0,1\}^l$ of a tag T_i
$\oplus$	XOR operator
$\in$	Random choice operator

$f = S_2 \oplus P(d) \oplus P(e)$ are computed for each tag. Each tag uses its own embedded PUF $P(.)$ for the calculation of c and f. The back-end server stores $[ID, DATA]$ for each tag.

8.6.2. Authentication Phase

(i) The reader creates a nonce $r_1 \in \{0,1\}^l$ and sends $M_1 \leftarrow S_1 \oplus r_1$ to tags.

(ii) Upon receiving M_1, a tag T_i creates a nonce $r_2 \in \{0,1\}^l$ and calculates $M_2 = H(r_2, ID_i, M_1)$, $k \leftarrow H(r_2, 1, 2)$, $M_3 \leftarrow k \oplus ID_i$, $M_4 \leftarrow M_1 \oplus r_2$. After these calculations, r_2 is deleted from the volatile memory. The tag continues to calculate M_4 as follows: It XORes M_4 with $P_i(a_i) \oplus P_i(d_i)$. $P_i(a_i)$ and $P_i(d_i)$ are deleted from the volatile memory. It then XORes M_4 with $P_i(b_i) \oplus P_i(e_i)$. $P_i(b_i)$ and $P_i(e_i)$ are deleted from the volatile memory. Finally, it computes M_4 by XORing with $c_i \oplus f_i$ and sends M_2, M_3 and M_4 to the reader.

(iii) The reader calculates $r_2 \leftarrow M_4 \oplus S_2 \oplus r_1$ and $ID_i \leftarrow M_3 \oplus H(r_2, 1, 2)$. It checks the validity of M_2. If M_2 is not valid, the reader stops the session.

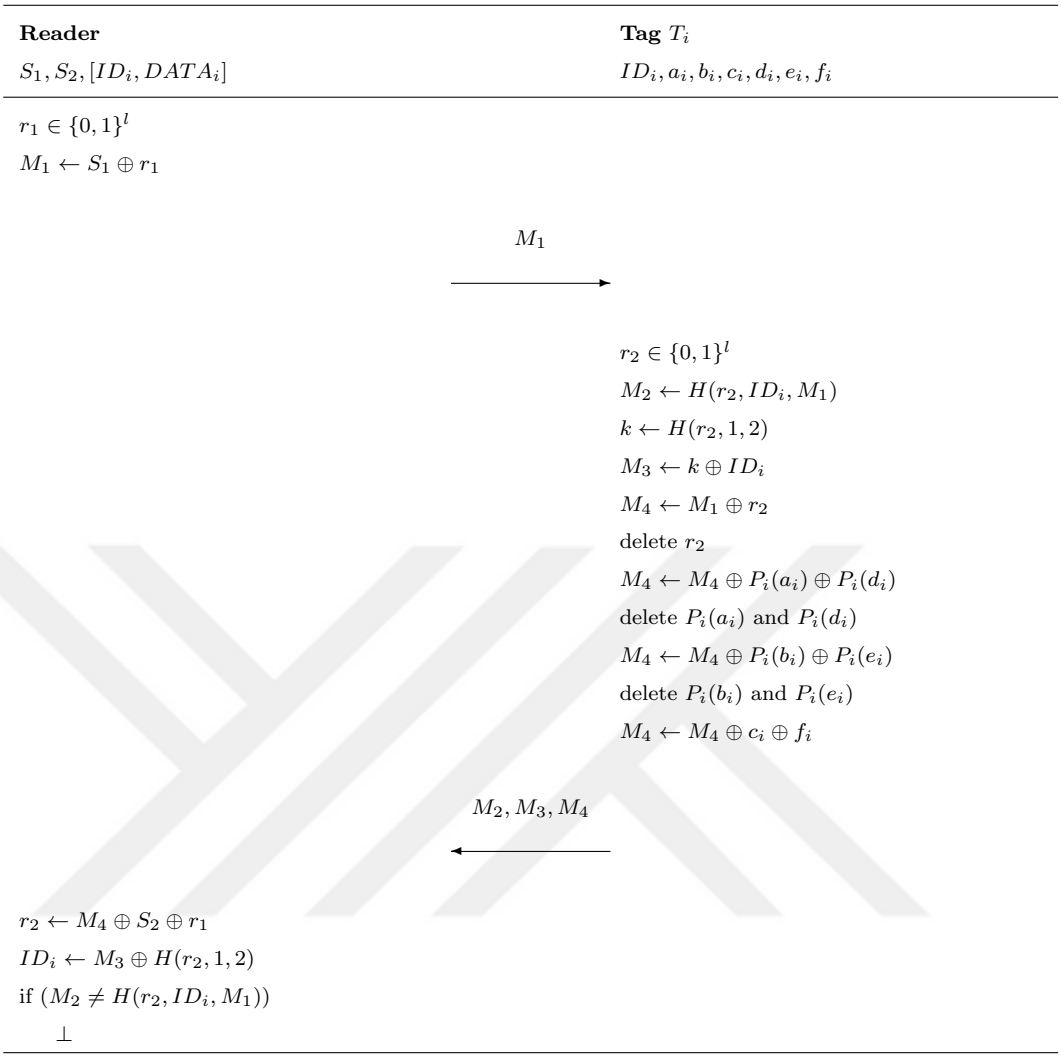

Figure 8.5. Scalable RFID authentication protocol III

8.6.3. Security and Privacy Analysis

Lemma 8.7. *Let $\mathcal{A}$ be a destructive adversary. The advantage of $\mathcal{A}$ of obtaining the master keys S_1 and S_2 by corrupting a tag is negligible.*

Proof. There are three deletion operations in the proposed protocol. We consider each of them one by one to determine $\mathcal{A}$'s advantage. Let assume $\mathcal{A}$ corrupts a tag T_i before the first deletion. $\mathcal{A}$ obtains r_2 and $H(r_2, 1, 2)$. In order to obtain the master key S_1, $\mathcal{A}$ has to know the random number r_1 or has to simulate $P_i(.)$. In order to obtain the master key S_2, $\mathcal{A}$ has to simulate $P_i(.)$. Let assume $\mathcal{A}$ corrupts the tag T_i before the second deletion. $\mathcal{A}$ obtains $H(r_2, 1, 2)$, $P_i(a_i)$ and $P_i(d_i)$. In order to

generate the master key S_1, $\mathcal{A}$ has to know random numbers r_1 or has to simulate $P_i(.)$ to calculate $P_i(b_i)$. In order to generate the master key S_2, $\mathcal{A}$ has to simulate $P_i(.)$ to calculate $P_i(e_i)$. Let assume $\mathcal{A}$ corrupts the tag T_i before the third deletion. $\mathcal{A}$ obtains $H(r_2, 1, 2)$, $P_i(b_i)$ and $P_i(e_i)$. In order to generate the master key S_1, $\mathcal{A}$ has to know random numbers r_1 or has to simulate $P_i(.)$ to calculate $P_i(a_i)$. In order to generate the master key S_2, $\mathcal{A}$ has to simulate $P_i(.)$ to calculate $P_i(d_i)$. $\mathcal{A}$ can obtain one secret value created with $P_i(.)$ at a time. Thus, $\mathcal{A}$ has to simulate $P_i(.)$ or $\mathcal{A}$ has to generate an input for a given hash output. As a result, $\mathcal{A}$ can learn S_1 and S_2 by corrupting the tag T_i with negligible probability.

Theorem 8.8. *The proposed protocol provides tag authentication if H is a hash function (Definition 8.4) and P is a PUF (Definition 8.5).*

Proof. We assume that there is an adversary $\mathcal{A}$ that can impersonate a tag T_i to a reader R with non-negligible probability. After receiving a message M_1, $\mathcal{A}$ has to generate messages M_2, M_3 and M_4 such that

$$\begin{aligned} M_2 &= H(r_2, ID_i, M_1) \\ M_3 &= H(r_2, 1, 2) \oplus ID_i \\ M_4 &= M_1 \oplus r_2 \oplus P_i(a_i) \oplus P_i(d_i) \oplus P_i(b_i) \oplus P_i(e_i) \oplus c_i \oplus f_i \end{aligned} \tag{8.22}$$

We know that $\mathcal{A}$ can not learn master secrets S_1 and S_2 by corrupting tags (Lemma 8.7). $\mathcal{A}$ can corrupt the tag T_i and learn ID_i, a_i, b_i, c_i, d_i, e_i, f_i. She has to simulate $P_i(.)$ in order to correctly generate the messages M_4. This will contradict with unclonability of $P_i(.)$. $\mathcal{A}$ may use previous responses of the tag T_i. For example, $\mathcal{A}$ eavesdrops the transcript (M_1^s, M_2^s, M_3^s, M_4^s) from the session s between the tag T_i and the reader. In the session $s+1$, the reader queries $\mathcal{A}$ with M_1^{s+1}. In order to generate valid responses, $\mathcal{A}$ uses $r_2^{s+1} = r_2^s$. $\mathcal{A}$ can easily generate $M_3^{s+1} = M_3^s$ and $M_4^{s+1} = M_4^s \oplus M_1^{s+1} \oplus M_1^s$. In order to generate M_2^{s+1}, $\mathcal{A}$ needs to know r_2^s. M_2 is generated by computing $H(r_2, ID_i, M_1)$ where M_1 comes from the reader and r_2 is not known by the adversary. We know that $H(.)$ is a random-like function (Definition 8.4). As a result, $\mathcal{A}$ can use these previous responses with the negligible probability $2^{1-l}N$ where

l is the security parameter (the bit length of random nonces and messages).

Theorem 8.9. *The proposed protocol achieves narrow-destructive privacy if the protocol achieves tag authentication, P is PUF (Definition 8.5) and H is hash function (Definition 8.4).*

Proof. We assume that there is an adversary $\mathcal{A}$ that can distinguish simulated RFID system from the real RFID system with non-negligible probability. In simulated RFID system $\mathcal{S}$, all protocol flows are simulated with random messages and all protocol functions return random values.

Let assume that there is a system $\mathcal{S}_0$ whose simulation equals to real system. We construct a new system $\mathcal{S}_1$ from $\mathcal{S}_0$. In the system $\mathcal{S}_1$, the states of all tags are simulated with randomly chosen values. For example, $\mathcal{S}_1$ assigns random values $P_i(a_i) \in \{0,1\}^l$, $P_i(b_i) \in \{0,1\}^l$, $c_i \in \{0,1\}^l$, $P_i(d_i) \in \{0,1\}^l$, $P_i(e_i) \in \{0,1\}^l$ and $f_i \in \{0,1\}^l$ for a tag T_i. $\mathcal{S}_1$ generates messages M_2, M_3 and M_4 by using these random values. In this game, $\mathcal{A}$ tries to distinguish $\mathcal{S}_1$ from $\mathcal{S}_0$. During the attack time, $\mathcal{A}$ can corrupt tags. However, $\mathcal{A}$ cannot obtain any secret (Lemma 8.7) and corrupted tags cannot be used any more (Definition 8.5). After a polynomial times of queries, $\mathcal{A}$ can distinguish $\mathcal{B}_1$ from $\mathcal{B}_0$. That means $\mathcal{A}$ can distinguish the output of a PUF from a randomly chosen value with non-negligible probability. This statement contradicts with the security property of the PUF (Definition 8.5). As a result, the success probability of $\mathcal{A}$ is negligible.

We construct a new system $\mathcal{S}_2$ from $\mathcal{S}_1$. In the system $\mathcal{S}_2$, messages M_2, M_3 and M_4 are generated by assigning random values to them, as in the system $\mathcal{S}$. In this game, $\mathcal{A}$ tries to distinguish $\mathcal{S}_2$ from $\mathcal{S}_1$. After a polynomial times of queries, $\mathcal{A}$ can distinguish $\mathcal{S}_2$ from $\mathcal{S}_1$. That means $\mathcal{A}$ can distinguish the output of a hash function from a randomly chosen value with non-negligible probability. To do this, $\mathcal{A}$ must solve the output of the hash function. This contradicts with the security property of the hash function (Definition 8.4). As a result, the success probability of $\mathcal{A}$ is negligible.

We construct a new system $\mathcal{S}_3$ from $\mathcal{S}_2$. The system $\mathcal{S}_0$ returns 1, if the reader identifies a legitimate tag, and 0 otherwise at the end of the protocol session. At the end of the protocol session, the system $\mathcal{S}_3$ returns 1 if the protocol session and protocol messages are generated, as in the system $\mathcal{S}_0$ and 0 otherwise. In this game, $\mathcal{A}$ tries to distinguish $\mathcal{S}_3$ from $\mathcal{S}_2$. After a polynomial times of queries, $\mathcal{A}$ can distinguish $\mathcal{S}_3$ from $\mathcal{S}_2$. That means $\mathcal{A}$ runs a protocol instance π and $\mathcal{S}_3$ returns a different output than $\mathcal{S}_2$. We know that the simulation in $\mathcal{S}_3$ is perfect and this can only happen when $\mathcal{A}$ generates a protocol transcript that makes the system $\mathcal{S}_2$ returns 1. Theorem 8.8 says that this can happen with negligible probability. As a result the success probability of $\mathcal{A}$ is negligible.

Full proof shows that the system $\mathcal{S}_3$ equals to simulated system $\mathcal{S}$ and the system $\mathcal{S}_0$ equals to real system. This means $\mathcal{A}$ cannot distinguish the simulated system $\mathcal{S}$ from the real system with non-negligible probability.

8.7. A New Scalable RFID Authentication Protocol IV

The proposed protocol has two phases: initialization and authentication. Table 8.4 gives the notations used in describing the proposed protocol.

Table 8.4. Notations of scalable RFID authentication protocol IV

Notation	Description
S	The shared secret
ID_i	The identifier of a tag T_i
$DATA_i$	Information about a tag T_i
$(a, b, c)_i$	Secret values of a tag T_i
H	A hash function $\{0,1\}^l \times \{0,1\}^l \times \{0,1\}^l \rightarrow \{0,1\}^l$
P_i	The PUF $\{0,1\}^l \rightarrow \{0,1\}^l$ of a tag T_i
$\oplus$	XOR operator
$\in$	Random choice operator

8.7.1. Initialization Phase

A random key S is generated for the back-end server. Two random unique keys a and b generated for each tag. Then, the key $c = S \oplus P(a) \oplus P(b)$ is computed for each tag. Each tag uses its own embedded PUF $P(.)$ for the calculation of c. The back-end server stores $[ID, a, b, DATA]$ for each tag.

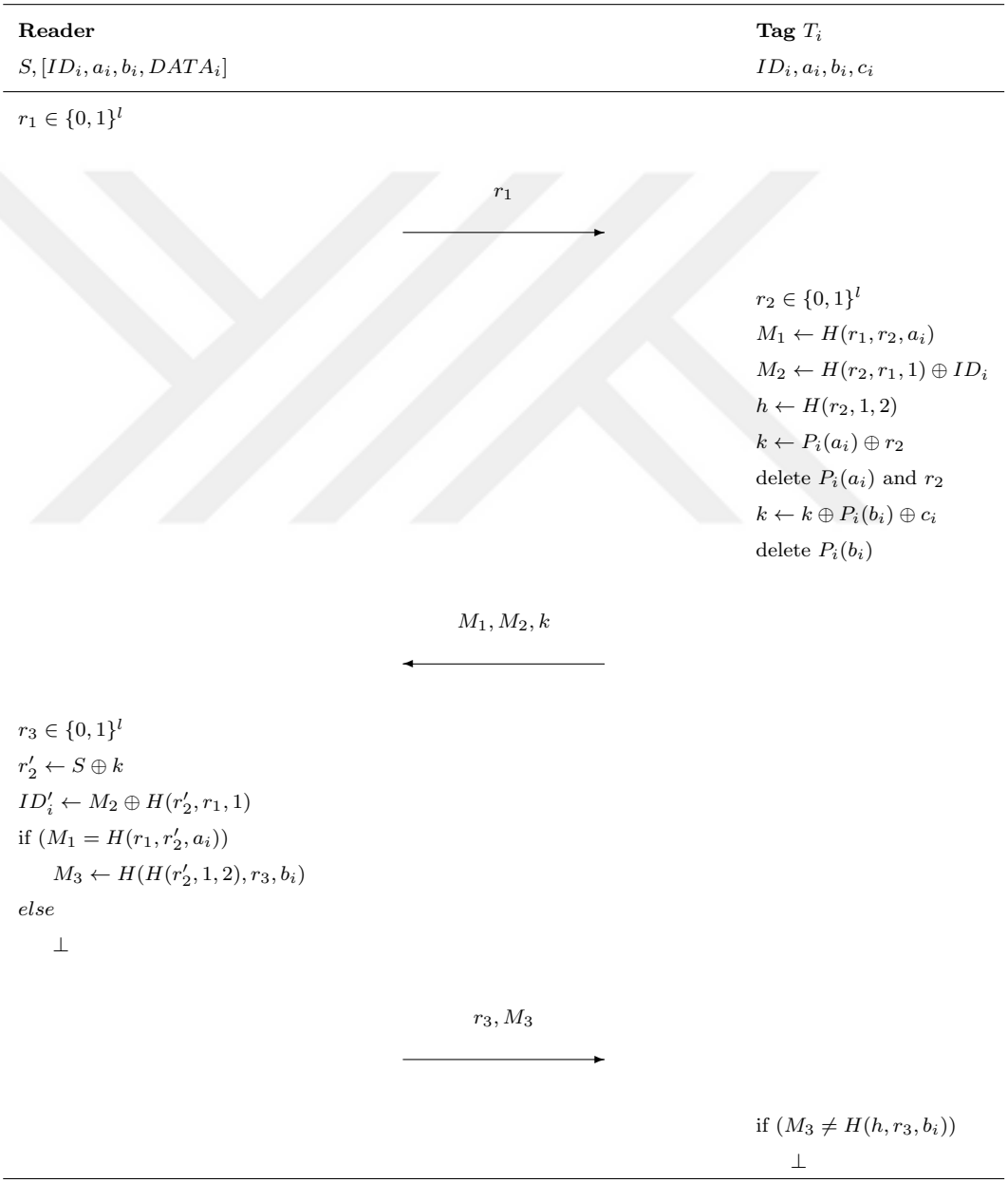

Figure 8.6. Scalable RFID authentication protocol IV

8.7.2. Authentication Phase

(i) The reader creates a nonce $r_1 \in \{0,1\}^l$ and broadcasts it.

(ii) A tag T_i creates a nonce $r_2 \in \{0,1\}^l$. It calculates $M_1 \leftarrow H(r_1, r_2, a_i)$, $M_2 \leftarrow H(r_2, r_1, 1) \oplus ID_i$ and $h \leftarrow H(r_2, 1, 2)$. Then, it calculates the message k by XORing $P_i(a_i)$ and r_2. $P_i(a_i)$ and r_2 are deleted from the volatile memory. The message k is replaced with $k \oplus P_i(b_i) \oplus c_i$ and $P_i(b_i)$ is deleted from the volatile memory. The tag sends M_1, M_2 and k to the reader.

(iii) The reader creates a nonce $r_3 \in \{0,1\}^l$. It calculates $r_2' \leftarrow S \oplus k$ and $ID_i' \leftarrow M_2 \oplus H(r_2', r_1, 1)$. The reader checks the validity of M_1 by computing $H(r_1, r_2', a_i)$ in order to authenticate the tag T_i. If the tag T_i is authenticated, the reader computes $M_3 = H(H(r_2', 1, 2), r_3, b_i)$. It sends r_3 and M_3 to the tag T_i.

(iv) The tag T_i checks the validity of M_3 by computing $H(h, r_3, b_i)$. If it is valid, the tag T_i authenticates the reader.

8.7.3. Security Analysis

Our protocol provides resistance to impersonation attacks. In the following, we will formally prove impersonation resistance of our protocol. Our protocol is a stateless protocol. Tags do not need to remain synchronized with the back-end database. Therefore, desynchronization attacks cannot be applied to our protocol.

Lemma 8.10. *Let $\mathcal{A}$ be a destructive adversary. The advantage of $\mathcal{A}$ of obtaining the shared key S without calling* Corrupt *oracle is negligible.*

Proof. We assume that there is an adversary $\mathcal{A}$ that can learn the shared secret S by without using Corrupt oracle. Each tag responds to reader's query with (M_1, M_2, k) where $k = S \oplus r_2$ and r_2 is random number created by a tag. In order to reveal the shared secret S, $\mathcal{A}$ has to known the random number r_2. However, r_2 is not sent as a cleartext. $\mathcal{A}$ has to expose r_2 from messages M_1, M_2 and M_3. This contradicts with the security property of the Hash functions (Definition 8.4).

Moreover, the identifier of the tag ID_i is sent to the reader in encrypted format $H(r_2, r_1, 1) \oplus ID_i$. $\mathcal{A}$ can not expose the ID_i without knowing the random number r_2.

Lemma 8.11. *Let $\mathcal{A}$ be a destructive adversary. The advantage of $\mathcal{A}$ of obtaining the shared key S by corrupting a tag is negligible.*

Proof. We assume that there is an adversary $\mathcal{A}$ that can learn the shared secret S by corrupting a tag. In this proof, we evaluate the timing of $\mathcal{A}$ who will corrupt a tag T_i. In the first case, $\mathcal{A}$ corrupts the tag when it is not interacting with the reader. In this case, $\mathcal{A}$ can access the values of a_i, b_i, c_i and ID_i. She can not learn the shared secret S because $S = P_i(a_i) \oplus P_i(b_i) \oplus c_i$ and the volatile memory of the tag is empty. Therefore, $\mathcal{A}$ has to simulate $P_i(.)$ in order to calculate the shared secret S. In the second case, $\mathcal{A}$ corrupts the tag when it is interacting with the reader. This case has two sub cases which are determined according to two deletion operation in the tag side. In the first sub case, $\mathcal{A}$ corrupts the tag T_i before the first deletion. $\mathcal{A}$ obtains $P_i(a_i)$, c_i, r_2 and $H(r_2, 1, 2)$. In order to generate the shared key S, $\mathcal{A}$ has to obtain the secret value $P_i(b_i)$. Thus, $\mathcal{A}$ has to simulate $P_i(.)$. In the second sub case, $\mathcal{A}$ corrupts the tag T_i before the second deletion. $\mathcal{A}$ obtains $P_i(b_i)$, $(P_i(a_i) \oplus r_2)$, $H(r_2, 1, 2)$ and c_i. In order to generate the shared key S, $\mathcal{A}$ has to obtain the secret value $P_i(a_i)$. Thus, $\mathcal{A}$ has to simulate $P_i(.)$ or has to obtain $P_i(a_i)$ from $(P_i(a_i) \oplus r_2)$. As a result, $\mathcal{A}$ has to simulate $P(.)$ in order to calculate the shared secret S. This contradicts with the security property of the PUF (Definition 8.5).

Theorem 8.12. *The proposed protocol provides tag authentication if H is a hash function (Definition 8.4) and P is a PUF (Definition 8.5).*

Proof. We assume that there is an adversary $\mathcal{A}$ that can generate (M_1, M_2, k) for a given r_1 with non-negligible probability. At the end of the security experiment, $\mathcal{A}$ wins if the reader sends r_3, M_3 to $\mathcal{A}$

Let $\mathcal{A}$ allowed access all oracles. $\mathcal{A}$ can corrupt a tag T_i and learn ID_i, a_i, b_i, c_i. She has to simulate $P_i(.)$ in order to correctly answer the reader's queries. This will

contradict with unclonability of $P_i(.)$. Kardaş et. al [134] showed that a secret value created with a PUF can be learnt by $\mathcal{A}$, if the corruption occurs in the time period in which the secret is in volatile memory. In our protocol, $\mathcal{A}$ can obtain just one secret $P_i(a)$ or $P_i(b)$ created with $P_i(.)$. $P_i(a)$ and $P_i(b)$ cannot be in the volatile memory at the same time. $\mathcal{A}$ knowing one secret generated with $P_i(.)$ cannot generate valid responses in order to pass the check by the reader.

Let $\mathcal{A}$ allowed access all oracles except Corrupt oracles. $\mathcal{A}$ obtains n_1 protocol transcripts between a tag T_i and the reader by using Execute oracle. se Furthermore, $\mathcal{A}$ queries the tag T_i n_2 times by using SendReader oracle. $\mathcal{A}$ obtains a total of $N = n_1+n_2$ responses generated by the tag T_i where N is polynomially bounded. $\mathcal{A}$ has to generate a pair (M_1, M_2, k) to impersonate a tag T_i. M_1 equals to $H(r_1, r_2, a_i)$ and M_2 is created by computing $H(r_2, r_1, 1)$ where r_1 comes from the reader and r_2 and a_i are not known by the adversary. k is generated by computing $P_i(a_i) \oplus r_2 \oplus P_i(b_i) \oplus c_i$ where r_2 is a random number and is not known by the adversary. We know that $H(.)$ is a random-like function (Definition 8.4). As a result, $\mathcal{A}$ can use these previous responses with the negligible probability $2^{1-l}N$ where l is the security parameter (the bit length of random nonces and messages).

8.7.4. Privacy Analysis

Theorem 8.13. *The protocol depicted in Figure 8.6 achieves timeful-destructive privacy (Definition 8.3) if the protocol achieves tag authentication, P is PUF (Definition 8.5) and H is Hash Function (Definition 8.4).*

Proof. We assume that there is an adversary $\mathcal{A}$ that can distinguish oracles simulated by blinder $\mathcal{B}$ from the real oracles with non-negligible probability. We first show how $\mathcal{B}$ simulates oracles.

- Launch() It is trivial to simulate Launch.
- SendTag(r_1, vtag) Returns $M_1 \in \{0,1\}^l$, $M_2 \in \{0,1\}^l$ and $k \in \{0,1\}^l$.

- SendTag($\mathsf{r}_3, \mathsf{M}_3, \mathsf{vtag}$) It does not produce any output so $\mathcal{B}$ does not need to simulate it.
- SendReader(π) Returns $r_1 \in \{0,1\}^l$
- SendReader($(\mathsf{M}_1, \mathsf{M}_2, \mathsf{k}), \pi$) Returns $r_3 \in \{0,1\}^l$ and $M_3 \in \{0,1\}^l$.
- Result(π) Returns 1 if π has been generated with Launch oracle and the corresponding protocol transcript has been generated with the real SendTag and SendReader oracles and 0 otherwise.
- Timer(π) Returns the constant time required for carrying out operations in the reader side.

Let assume that there is blinder $\mathcal{B}_0$ whose simulation equals to real oracles. We construct a new blinder $\mathcal{B}_1$ from $\mathcal{B}_0$. The differences between $\mathcal{B}_1$ and $\mathcal{B}_0$ are that the states of all tags are simulated with randomly chosen values and the simulation of SendTag oracle. For example, $\mathcal{B}_1$ assigns random values $P_i(a_i) \in \{0,1\}^l$, $P_i(b_i) \in \{0,1\}^l$ and $c_i \in \{0,1\}^l$ for a tag T_i. $\mathcal{B}_1$ simulates SendTag oracle by evaluating $H(.)$ with randomly assigned values. When $\mathcal{A}$ wants to use SendTag oracle, the challenger $\mathcal{C}$ either evaluates SendTag with real values as in $\mathcal{B}_0$ or with random values as in $\mathcal{B}_1$. During the attack time, $\mathcal{A}$ can call Execute(vtag) and analyze the successive messages between $vtag$ and the reader in order to obtain any secret. However, $\mathcal{A}$ cannot obtain any secret from the list of successive messages (Lemma 8.10). $\mathcal{A}$ can also use Corrupt(vtag) in order to obtain the state of the tag $vtag$. However, $\mathcal{A}$ cannot obtain any secret from the tag $vtag$ (Lemma 8.11) and the tag $vtag$ cannot be used any more (Definition 8.5). After a polynomial times of oracle queries, $\mathcal{A}$ can distinguish $\mathcal{B}_1$ from $\mathcal{B}_0$. That means $\mathcal{A}$ can distinguish the output of a PUF from a randomly chosen value with non-negligible probability. This statement contradicts with the security property of the PUF (Definition 8.5). As a result $|Pr[\mathcal{A}^{B_0} \text{ wins}] - Pr[\mathcal{A}^{B_1} \text{ wins}]|$ is negligible.

We construct a new blinder $\mathcal{B}_2$ from $\mathcal{B}_1$. The only difference between $\mathcal{B}_2$ and $\mathcal{B}_1$ is that SendTag oracle is simulated as described above. When $\mathcal{A}$ wants to use SendTag oracle, the challenger $\mathcal{C}$ either evaluates SendTag as in $\mathcal{B}_1$ or sends random values as in $\mathcal{B}_2$. After a polynomial times of oracle queries, $\mathcal{A}$ can distinguish $\mathcal{B}_2$ from $\mathcal{B}_1$. That means $\mathcal{A}$ can distinguish the output of a hash function from a randomly chosen

value with non-negligible probability. To do this, $\mathcal{A}$ must solve the output of the hash function. This contradicts with the security property of the hash function (Definition 8.4). As a result $|Pr[\mathcal{A}^{B_1} \text{ wins}] - Pr[\mathcal{A}^{B_2} \text{ wins}]|$ is negligible.

We construct a new blinder $\mathcal{B}_3$ from $\mathcal{B}_2$. The only difference between $\mathcal{B}_3$ and $\mathcal{B}_2$ is that Result oracle is simulated as described above. When $\mathcal{A}$ wants to use Result oracle, the challenger $\mathcal{C}$ either evaluates the real Result as in $\mathcal{B}_2$ or the simulated Result as in $\mathcal{B}_3$. After a polynomial times of oracle queries, $\mathcal{A}$ can distinguish $\mathcal{B}_3$ from $\mathcal{B}_2$. That means $\mathcal{A}$ runs a protocol instance π and the simulated Result returns a different output than the real Result. We know that the simulation of Result oracle is perfect and this can only happen when $\mathcal{A}$ generates a protocol transcript that makes the real Result oracle returns 1. Theorem 8.12 says that this can happen with negligible probability. As a result $|Pr[\mathcal{A}^{B_2} \text{ wins}] - Pr[\mathcal{A}^{B_3} \text{ wins}]|$ is negligible.

We construct a new blinder $\mathcal{B}_4$ from $\mathcal{B}_3$. The only difference between $\mathcal{B}_4$ and $\mathcal{B}_3$ is that Timer oracle is simulated as described above. When $\mathcal{A}$ wants to use Timer oracle, the challenger $\mathcal{C}$ either evaluates the real Timer as in $\mathcal{B}_3$ or the simulated Timer as in $\mathcal{B}_4$. After a polynomial times of oracle queries, $\mathcal{A}$ can distinguish $\mathcal{B}_4$ from $\mathcal{B}_3$. That means $\mathcal{A}$ runs a protocol instance π and the simulated Timer returns a different output than the real Timer. We know that the simulation of Timer oracle is perfect and the reader performs a fixed number operations for a tag identification. Both the real and simulated Timer always return the same result. an As a result $|Pr[\mathcal{A}^{B_3} \text{ wins}] - Pr[\mathcal{A}^{B_4} \text{ wins}]|$ is negligible.

Full proof shows that $|Pr[\mathcal{A}^{B_0} \text{ wins}] - Pr[\mathcal{A}^{B_4} \text{ wins}]|$ is negligible.

The blinder $\mathcal{B}_4$ equals to full blinder $\mathcal{B}$ and the blinder $\mathcal{B}_0$ equals to real oracles. This means $\mathcal{A}$ cannot distinguish oracles simulated by blinder $\mathcal{B}$ from the real oracles with non-negligible probability.

8.8. Comparison of Proposed Protocols

In this section, we compare our protocols with some previously proposed protocols that are explained briefly in Chapter 5. Table 8.5 gives the identification complexity, computational cost in a system with 2^{32} tags and privacy level of the protocols. The identification complexity of our first protocol is $\mathcal{O}(\log N)$ and the identification complexity of our other protocols is $\mathcal{O}(1)$. Wu and Stinson's protocol [63] and Alomair et al.'s protocol [64] also have the the same identification time. However, these protocols are susceptible to tracing attacks; therefore, they do not provide privacy in the Vaudenay-Model. Our first protocol provides *narrow-weak* privacy because it is susceptible to the cold boot attack. Our remaining three protocols provides *narrow-destructive* privacy in the Vaudenay-Model. Bringer et al.'s protocol [200] and Kardaş et al.'s protocol [83] also provide *narrow-destructive* privacy in the Vaudenay-Model. However, the identification complexities of Bringer et al.'s protocol and Kardaş et al.'s protocol are $\mathcal{O}(\log N)$ and $\mathcal{O}(N)$, respectively. Table 8.5 shows that our protocols give the highest privacy level with a lowest computational cost on the reader side.

In Table 8.6, we study the computational and storage costs of the selected protocols on both the tag and the back-end server sides. Our first protocol performs less operations on both tag and server when compared with Bringer et al.'s protocol [200]. However, our first protocol provides weak privacy, while Bringer et al.'s protocol provides destructive privacy. In our second protocol, a tag creates two nonces and performs eight hash operations, six PUF operations and eight permutation operations. When compared with Avoine et al.'s protocol [58], Alomair et al.'s protocol [64], Sadeghi et al.'s protocol [201] and Kardaş et al.'s protocol [83], our protocol performs more operations on the tag side. However, in our protocol a tag takes RF signal from the reader two times, which means the tag takes power two times. When the tag is first powered, it creates a nonce and performs three hash operations, two PUF operations and three permutation operations. When the tag is powered for the second time, it creates a nonce and performs five hash operations, four PUF operations and five permutation operations. The computational cost corresponding to the power taken from the RF

Table 8.5. Scalability and privacy level comparison of protocols

	Search Time	Cost*	Privacy Level
Molnar and Wagner [51]	$\mathcal{O}(\log N)$	32	*narrow-weak*
Bringer et al. [200]	$\mathcal{O}(\log N)$	32	*narrow-destructive*
Avoine et al. [58]	$\mathcal{O}(N^{2/3})$	$2^{32/3}$	*narrow-forward*
Wu and Stinson [63]	$\mathcal{O}(1)$	128	*no-privacy*
Alomair et al. [64]	$\mathcal{O}(1)$	1	*no-privacy*
Sadeghi et al. [201]	$\mathcal{O}(N)$	2^{32}	*narrow-weak*
Kardaş et al. [83]	$\mathcal{O}(N)$	2^{32}	*narrow-destructive*
Protocol I	$\mathcal{O}(\log N)$	32	*narrow-weak*
Protocol II	$\mathcal{O}(1)$	1	*narrow-destructive*
Protocol III	$\mathcal{O}(1)$	1	*narrow-destructive*
Protocol IV	$\mathcal{O}(1)$	1	*narrow-destructive*

* Computational cost in the system with 2^{32} tags

Table 8.6. Computational and storage costs comparison of protocols

	Time Complexity		Space Complexity	
	Tag	Back-end Server	Tag	Back-end Server
Protocol in [51]	1 nonce + $(\log_2 N + 1)$ PRFs	$(\log_2 N + 1)$ L	$(2N - 1)$ L	
Protocol in [200]	$(\log_2 N + 1)$ nonces + $(\log_2 N + 1)$ hashes + $2(\log_2 N + 1)$ PUFs	$(\log_2 N + 1)$ hashes	$(\log_2 N + 1)$ L	$(2N - 1)$ L
Protocol in [58]	2 hashes	1 nonce + $N^{2/3}$ hashes	1 L	$N^{2/3}$ L
Protocol in [63]	$(2b - 1)$ nonces + 1 hash + 1 polynomial	mb polynomial equations	$m(k + 1)$ L	$(m(k + 1)^2)/8$ L
Protocol in [64]	5 hashes + 1 addition	1 nonce + 3 hashes	3 L	CN L
Protocol in [201]	1 nonce + 1 PUF + 1 PRF	1 nonce + N PRFs	L	N L
Protocol in [83]	1 nonce + 4 hashes + 2 PUFs	1 nonce + 2N hashes	2 L	2N L
Protocol 1	1 nonce + $(\log_2 N + 1)$ PRFs + 1 PUF	1 nonce + $(\log_2 N + 1)$ PRFs	$(\log_2 N + 2)$ L	$(2N - 1)$ L
Protocol 2	2 nonces + 8 hashes + 6 PUFs + 8 permutations	2 nonces + 6 hashes + 3 permutations	6 L	2 L
Protocol 3	1 nonces + 2 hashes + 4 PUFs	1 nonces + 2 hashes	6 L	2 L
Protocol 4	1 nonces + 4 hashes + 2 PUFs	2 nonces + 4 hashes	3 L	$(2N + 1)$ L

^{L} bit length of keys

^{b} security parameter stated in [63]

^{m} number of bivariate polynomials in [63]

^{k} degree of polynomial in [63]

^{C} maximum counter value in [64]

signal is almost the same as the other protocols. On the back-end side, the computational cost of our second protocol is higher than that of Alomair et al.'s protocol. This difference is too small for ordinary servers. On the tag side, our second protocol keeps six items. On the back-end server side, all protocols except Alomair et al.'s protocol consume almost the same amount of storage. We only evaluate the disk space used for the keys. Avoine et al. [28] stated that Alomair et al.'s protocol needs 31-40 TB of storage space on a back-end server with 10^9 tags in the system. Our second protocol needs only 512 bits of storage on the back-end server if the size of the master keys is 256 bits. Our third and fourth protocols have reasonable time and space complexity on both tag and server.

9. CONCLUSION

In this thesis, security and privacy aspects of Radio Frequency Identification (RFID) is studied. In particular, we focus on security analysis of RFID protocols and scalability and privacy issues in RFID systems. After giving a brief overview of RFID systems and defined security and privacy problems in RFID respectively in Chapter 2 and Chapter 3, we give the overview of previously proposed RFID security models and solutions respectively in Chapter 4 and Chapter 5.

In Chapter 6, we have presented security or privacy flaws in some recent RFID protocols that have received no attacks yet. Also, we have proposed some revisions, if possible, to eliminate weakness in these target schemes.

In Chapter 7, we analyze the security of Chaotic-map based RFID protocols. Despite the proposed improvements, these protocols have fundamental weaknesses stem from message generation. They are vulnerable to tracking, tag impersonation and de-synchronization attacks. The success probabilities of the proposed attacks are significant and their complexities are polynomial. Furthermore, we propose improved RFID authentication protocols based on the Chebyshev chaotic map hard problem. Our protocols eliminates the weaknesses of previous protocols.

In Chapter 8, we propose four RFID authentication protocols that work with passive RFID tags. Our first protocol requires work logarithmic in the number of RFID tags in a system and vulnerable to cold bold attack. Our remaining protocols do not need a search operation on the server side to identify tags. The identification complexity of the proposed solutions is $\mathcal{O}(1)$. They completely solve the scalability problem by utilizing master keys shared by all tags. They are *destructive* private under the Vaudenay-Model, which means that our protocols provide privacy against adversaries who are capable of destroying tags permanently. They provide resistance against physical attacks (corrupting) by using Physically Unclonable Functions (PUFs) as a secure storage mechanism to preserve the privacy of the tag. In our protocols,

master keys are not stored on the tag. They are reproduced using PUFs when needed. Any physical attempt to obtain master keys results in destruction of the PUF, which means that adversaries that corrupt a tag cannot reproduce the master keys. To the best of our knowledge, our protocols are first to provide such a privacy level without requiring lookup.

9.1. Open Problems

Researchers meet some challenges while designing a security solutions for RFID systems. Although many of these challenges are overcame, there are still many open problems. In this section, we state these open problems. When designing a security protocol, there are three goals to be achieved: security, privacy and scalability. Although it is very easy to accomplish these goals one by one, it is very difficult to reach all of them at the same time.

9.1.1. Level of Privacy

All of previously proposed protocols try to provide tag authentication and prevent the tag from tracking and becoming identified by the third parties. However, these protocols can not provide the required high level of privacy. None of these provide strong privacy stated in [1]. There are several PUF-based solutions such as [200] providing narrow-destructive privacy and the protocol proposed in [58] provides narrow-forward privacy without needing PUF. Vaudenay [1] states that achieving strong privacy without public-key cryptography is impossible. He also asks that is there any way to provide destructive privacy or forward privacy without public-key techniques. This is still an open question.

9.1.2. Scalability

In a typical RFID protocol, tag's identifiers are not sent in plaintext in order to achieve privacy. That means tags randomize their responses in a way which can not be distinguished from by third parties. If tags are public-key crypto-capable devices,

sending randomized identifiers to the reader by encrypting them with the reader's public key provides both scalability and privacy. However, tags are restricted devices in terms of computational power and storage and they does not support public-key cryptography. As a conclusion, privacy and scalability goals of RFID systems must be solved with symmetric-key based protocols. It is very challenging issue because the reader is confronted with a deadlock such that it must decrypt the tag's response to obtain its identifier and it must know the tag's identifier to obtain its key. This problem is solved with non-scalable method in which all entries in the reader's database are searched. In the literature, many of the proposed solutions are carrying out the process of identification by doing a linear search on the number of tags. There are some RFID security protocols that require O(1) or O(log n) time where n is the number of tags for tag identification. However, these protocols can not provide the highest level of privacy and have some security deficiencies.

9.1.3. Efficiency

Security solutions for RFID systems should be efficient in terms of computation and communication. Because RFID tags have a very small amount of memory and their computational capabilities are very limited. As observed in this study, the majority of security solutions are based on classical cryptographic operations such as hash function, PRNGs and PUFs. However it is not easy to implement these cryptographic operations on low-cost RFID tags. Some researchers have proposed solutions based on lightweight operations such as CRC, modular addition, exclusive OR, etc. All proposed solutions have some of the security and privacy weaknesses. There is no solution in the literature that solves all security and privacy issues. Using lightweight operations to produce an efficient security solution for RFID systems is still an important research area. Furthermore, HB-based lightweight protocols can be considered as promising. However, the security and privacy provided by HB-based protocols do not seem sufficient for RFID systems. Almost all of these protocols can be broken with man-in-the-middle attacks. An HB-based protocol meeting all the security requirements of RFID systems is still not developed.

9.1.4. Resistance to Relay Attacks

Developing distance bounding protocols for RFID systems is still a popular research topic. It is still open question whether it is possible to develop an RFID distance bounding protocol with the ideal security level $(1/2)^n$ for mafia, terrorist and distance frauds defined in Section 5.7.

REFERENCES

1. Vaudenay, S., "On Privacy Models for RFID", K. Kurosawa (Editor), *Advances in Cryptology – Asiacrypt 2007*, Vol. 4833 of *Lecture Notes in Computer Science*, pp. 68–87, Springer, Kuching, Malaysia, December 2007.

2. Henrici, D., *RFID Security and Privacy*, Springer, 2008.

3. Shepard, S., *RFID Radio Frequency Identification*, McGraw-Hill, 2005.

4. Akgün, M. and M. Çaglayan, "Server Impersonation Attacks and Revisions to SLAP, RFID Lightweight Mutual Authentication Protocol", *Systems and Networks Communications (ICSNC), 2010 Fifth International Conference on*, pp. 148–153, Aug 2010.

5. Akgün, M., A. Özhan Gürel and M. U. Çaglayan, "Attacks to a lightweight RFID mutual authentication protocol.", *5th International Conference for Internet Technology and Secured Transactions, ICITST 2010, London, United Kingdom, November 8-10, 2010*, pp. 1–5, 2010.

6. Akgün, M. and M. U. Çaglayan, "Weaknesses of two RFID protocols regarding de-synchronization attacks", *International Wireless Communications and Mobile Computing Conference, IWCMC 2015, Dubrovnik, Croatia, August 24-28, 2015*, pp. 828–833, 2015.

7. Akgün, M. and M. U. Çaglayan, "Vulnerabilities of RFID Security Protocol Based on Chaotic Maps", *22nd IEEE International Conference on Network Protocols, ICNP 2014, Raleigh, NC, USA, October 21-24, 2014*, pp. 648–653, 2014.

8. Akgün, M., A. O. Bayrak and M. U. Çaglayan, "Attacks and improvements to chaotic map-based RFID authentication protocol", *Security and Communication Networks*, Vol. 8, No. 18, pp. 4028–4040, 2015.

9. Akgün, M. and M. U. Çaglayan, "PUF Based Scalable Private RFID Authentication.", *Sixth International Conference on Availability, Reliability and Security, ARES 2011, Vienna, Austria, August 22-26, 2011*, pp. 473–478, 2011.

10. Akgün, M. and M. U. Çağlayan, "Providing destructive privacy and scalability in RFID systems using PUFs", *Ad Hoc Networks*, Vol. 32, pp. 32–42, 2015.

11. Akgün, M. and M. U. Çaglayan, "Towards Scalable Identification in RFID Systems", *Wireless Personal Communications*, Vol. 86, No. 2, pp. 403–421, 2016.

12. Peris-Lopez, P., *Lightweight Cryptography in Radio Frequency Identification (RFID) Systems*, Ph.D. Thesis, Computer Science Department, Carlos III University of Madrid, November 2008.

13. Cole, P. H. and D. C. Ranasinghe, *Networked RFID Systems and Lightweight Cryptography*, Springer-Verlag, 2008.

14. Thornton, F., B. Hanies, A. M. Das, H. Bhargava, A. Campbell and J. Kleinschmidt, *RFID Security*, Syngress, 2006.

15. Ohkubo, M., K. Suzuki and S. Kinoshita, "Efficient Hash-Chain Based RFID Privacy Protection Scheme", *International Conference on Ubiquitous Computing – Ubicomp, Workshop Privacy: Current Status and Future Directions*, Nottingham, England, September 2004.

16. Lim, C. H. and T. Kwon, "Strong and Robust RFID Authentication Enabling Perfect Ownership Transfer", P. Ning, S. Qing and N. Li (Editors), *International Conference on Information and Communications Security – ICICS'06*, Vol. 4307 of *Lecture Notes in Computer Science*, pp. 1–20, Springer, Raleigh, North Carolina, USA, December 2006.

17. Song, B. and C. J. Mitchell, "RFID Authentication Protocol for Low-cost Tags", V. D. Gligor, J.-P. Hubaux and R. Poovendran (Editors), *Proceedings of the 1st*

ACM Conference on Wireless Network Security – WiSec'08, pp. 140–147, ACM, ACM Press, Alexandria, Virginia, USA, March–April 2008.

18. Hjorth, T., *Supporting Privacy in RFID Systems*, Master thesis, Technical University of Denmark, Lyngby, Denmark, December 2004.

19. Weis, S., *Security and Privacy in Radio-Frequency Identification Devices*, Master thesis, Massachusetts Institute of Technology (MIT), MIT, Massachusetts, USA, May 2003.

20. Feldhofer, M., S. Dominikus and J. Wolkerstorfer, "Strong Authentication for RFID Systems using the AES Algorithm", M. Joye and J.-J. Quisquater (Editors), *Workshop on Cryptographic Hardware and Embedded Systems – CHES 2004*, Vol. 3156 of *Lecture Notes in Computer Science*, pp. 357–370, IACR, Springer, Boston, Massachusetts, USA, August 2004.

21. Avoine, G., *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.

22. Juels, A. and S. Weis, "Defining Strong Privacy for RFID", *International Conference on Pervasive Computing and Communications – PerCom 2007*, pp. 342–347, IEEE, IEEE Computer Society, New York City, New York, USA, March 2007.

23. Ohkubo, M., K. Suzuki and S. Kinoshita, "Cryptographic Approach to "Privacy-Friendly" Tags", *RFID Privacy Workshop*, MIT, Massachusetts, USA, November 2003.

24. Ouafi, K. and R. C.-W. Phan, "Privacy of Recent RFID Authentication Protocols", L. Chen, Y. Mu and W. Susilo (Editors), *4th International Conference on Information Security Practice and Experience – ISPEC 2008*, Vol. 4991 of *Lecture Notes in Computer Science*, pp. 263–277, Springer, Sydney, Australia, April

2008.

25. Ouafi, K. and R. C.-W. Phan, "Traceable Privacy of Recent Provably-Secure RFID Protocols", S. M. Bellovin, R. Gennaro, A. D. Keromytis and M. Yung (Editors), *Proceedings of the 6th International Conference on Applied Cryptography and Network Security – ACNS 2008*, Vol. 5037 of *Lecture Notes in Computer Science*, pp. 479–489, Springer, New York City, New York, USA, June 2008.

26. Paise, R.-I. and S. Vaudenay, "Mutual Authentication in RFID: Security and Privacy", M. Abe and V. D. Gligor (Editors), *Proceedings of the 3rd ACM Symposium on Information, Computer and Communications Security – ASIACCS'08*, pp. 292–299, ACM, ACM Press, Tokyo, Japan, March 2008.

27. Armknecht, F., A.-R. Sadeghi, I. Visconti and C. Wachsmann, "On RFID Privacy with Mutual Authentication and Tag Corruption", J. Zhou and M. Yung (Editors), *Proceedings of the 8th International Conference on Applied Cryptography and Network Security – ACNS 2010*, Vol. 6123 of *Lecture Notes in Computer Science*, pp. 493–510, Springer, Beijing, China, June 2010.

28. Avoine, G., I. Coisel and T. Martin, "Time Measurement Threatens Privacy-Friendly RFID Authentication Protocols", S. O. Yalcin (Editor), *Workshop on RFID Security – RFIDSec'10*, Vol. 6370 of *Lecture Notes in Computer Science*, pp. 138–157, Springer, Istanbul, Turkey, June 2010.

29. Akgün, M. and M. Çağlayan, "Extending an RFID Security and Privacy Model by Considering Forward Untraceability", *Security and Trust Management*, pp. 239–254, Technical University of Denmark, Copenhagen, June 2011.

30. Ha, J., S. Moon, J. Zhou and J. Ha, "A New Formal Proof Model for RFID Location Privacy", S. Jajodia and J. López (Editors), *13th European Symposium on Research in Computer Security – ESORICS 2008*, Vol. 5283 of *Lecture Notes in Computer Science*, pp. 267–281, Springer, Malaga, Spain, October 2008.

31. van Deursen, T. and S. Radomirović, "On a New Formal Proof Model for RFID Location Privacy", *Inf. Process. Lett.*, Vol. 160, pp. 57,61, 2009.

32. van Deursen, T., S. Mauw and S. Radomirović, "Untraceability of RFID Protocols", J. A. Onieva, D. Sauveron, S. Chaumette, D. Gollmann and C. Markantonakis (Editors), *Workshop on Information Security Theory and Practice – WISTP'08*, Vol. 5019 of *Lecture Notes in Computer Science*, pp. 1–15, Springer, Sevilla, Spain, May 2008.

33. Ng, C., W. Susilo, Y. Mu and R. Safavi-Naini, "RFID Privacy Models Revisited", S. Jajodia and J. Lopez (Editors), *13th European Symposium on Research in Computer Security – ESORICS 2008*, Vol. 5283 of *Lecture Notes in Computer Science*, pp. 251–266, Springer Berlin / Heidelberg, Malaga, Spain, October 2008.

34. Yu Ng, C., W. Susilo, Y. Mu and R. Safavi-Naini, "New Privacy Results on Synchronized RFID Authentication Protocols against Tag Tracing", M. Backes and P. Ning (Editors), *14th European Symposium on Research in Computer Security – ESORICS 2009*, Vol. 5789 of *Lecture Notes in Computer Science*, pp. 321–336, Springer, Saint-Malo, France, September 2009.

35. DArco, P., A. Scafuro and I. Visconti, "Revisiting DoS Attacks and Privacy in RFID-Enabled Networks", S. Dolev (Editor), *Algorithmic Aspects of Wireless Sensor Networks*, Vol. 5804 of *Lecture Notes in Computer Science*, pp. 76–87, Springer Berlin / Heidelberg, 2009.

36. Ma, C., Y. Li, R. H. Deng and T. Li, "RFID Privacy: Relation Between Two Notions, Minimal Condition, and Efficient Construction", E. Al-Shaer, S. Jha and A. D. Keromytis (Editors), *Conference on Computer and Communications Security – ACM CCS'09*, pp. 54–65, ACM, ACM Press, Chicago, Illinois, USA, November 2009.

37. Milner, R., J. Parrow and D. Walker, "A calculus of mobile processes, I", *Information and Computation*, Vol. 100, No. 1, pp. 1 – 40, 1992.

38. Abadi, M. and C. Fournet, "Mobile values, new names, and secure communication", *Proceedings of the 28th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, POPL '01, pp. 104–115, ACM, New York, NY, USA, 2001.

39. Bruso, M., K. Chatzikokolakis and J. den Hartog, "Formal Verification of Privacy for RFID Systems", *Computer Security Foundations Symposium – CSF 2010*, IEEE, Edinburgh, United Kingdom, July 2010.

40. Lai, J., R. H. Deng and Y. Li, "Revisiting Unpredictability-Based RFID Privacy Models", J. Zhou and M. Yung (Editors), *Proceedings of the 8th International Conference on Applied Cryptography and Network Security – ACNS 2010*, Vol. 6123 of *Lecture Notes in Computer Science*, pp. 475–492, Springer, Beijing, China, June 2010.

41. Deng, R. H., Y. Li, M. Yung and Y. Zhao, "A New Framework for RFID Privacy", D. Gritzalis, B. Preneel and M. Theoharidou (Editors), *15th European Symposium on Research in Computer Security – ESORICS 2010*, Vol. 6345 of *Lecture Notes in Computer Science*, pp. 1–18, Springer, Athens, Greece, September 2010.

42. Hermans, J., A. Pashalidis, F. Vercauteren and B. Preneel, "A New RFID Privacy Model", *16th European Symposium on Research in Computer Security – ESORICS 2011*, Lecture Notes in Computer Science, Springer, Leuven, Belgium, September 2011.

43. Sarma, S., S. Weis and D. Engels, "RFID Systems and Security and Privacy Implications", B. Kaliski, c. Kaya ço and C. Paar (Editors), *Cryptographic Hardware and Embedded Systems – CHES 2002*, Vol. 2523 of *Lecture Notes in Computer Science*, pp. 454–469, Springer, Redwood Shores, California, USA, August 2002.

44. Juels, A., R. Rivest and M. Szydlo, "The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy", S. Jajodia, V. Atluri and T. Jaeger (Editors), *Conference on Computer and Communications Security – ACM CCS'03*, pp. 103–

111, ACM, ACM Press, Washington, DC, USA, October 2003.

45. Juels, A. and J. Brainard, "Soft Blocking: Flexible Blocker Tags on the Cheap", S. De Capitani di Vimercati and P. Syverson (Editors), *Workshop on Privacy in the Electronic Society – WPES'04*, pp. 1–7, ACM, ACM Press, Washington, DC, USA, October 2004.

46. Weis, S., S. Sarma, R. Rivest and D. Engels, "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems", D. Hutter, G. Müller, W. Stephan and M. Ullmann (Editors), *International Conference on Security in Pervasive Computing – SPC 2003*, Vol. 2802 of *Lecture Notes in Computer Science*, pp. 454–469, Springer, Boppard, Germany, March 2003.

47. Juels, A., "RFID Security and Privacy: A Research Survey", *IEEE Journal on Selected Areas in Communications*, Vol. 24, No. 2, pp. 381–394, February 2006.

48. Henrici, D. and P. Müller, "Hash-Based Enhancement of Location Privacy for Radio-Frequency Identification Devices Using Varying Identifiers", R. Sandhu and R. Thomas (Editors), *International Workshop on Pervasive Computing and Communication Security – PerSec 2004*, pp. 149–153, IEEE, IEEE Computer Society, Orlando, Florida, USA, March 2004.

49. Avoine, G., *Cryptography in Radio Frequency Identification and Fair Exchange Protocols*, Ph.D. Thesis, EPFL, Lausanne, Switzerland, December 2005.

50. Dimitriou, T., "A Lightweight RFID Protocol to protect against Traceability and Cloning attacks", *Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm 2005*, pp. 59–66, IEEE, IEEE Computer Society, Athens, Greece, September 2005.

51. Molnar, D. and D. Wagner, "Privacy and Security in Library RFID: Issues, Practices, and Architectures", V. Atluri, B. Pfitzmann and P. D. McDaniel (Editors), *Conference on Computer and Communications Security – ACM CCS'04*, pp. 210–

219, ACM, ACM Press, Washington, DC, USA, October 2004.

52. Molnar, D., *Security and Privacy in Two RFID Deployments, With New Methods For Private Authentication and RFID Pseudonyms*, Master thesis, University of California Berkeley, Berkeley, California, USA, 2006.

53. Dimitriou, T., "A Secure and Ef[U+FB01]cient RFID Protocol that could make Big Brother (partially) Obsolete", *International Conference on Pervasive Computing and Communications – PerCom 2006*, pp. 269–275, IEEE, IEEE Computer Society, Pisa, Italy, March 2006.

54. Lu, L., J. Han, L. Hu, Y. Liu and L. Ni, "Dynamic Key-Updating: Privacy-Preserving Authentication for RFID Systems", *International Conference on Pervasive Computing and Communications – PerCom 2007*, pp. 13–22, IEEE, IEEE Computer Society, New York City, New York, USA, March 2007.

55. Wang, W., Y. Li, L. Hu and L. Lu, "Storage-awareness: RFID private authentication based on sparse tree", *IEEE International Conference on Pervasive Services, Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing – SecPerU 2007*, pp. 61–66, July 2007.

56. Akgun, M., M. Caglayan and E. Anarim, "A new RFID authentication protocol with resistance to server impersonation", *Parallel Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pp. 1 –8, May 2009.

57. Avoine, G. and P. Oechslin, "A Scalable and Provably Secure Hash Based RFID Protocol", *International Workshop on Pervasive Computing and Communication Security – PerSec 2005*, pp. 110–114, IEEE, IEEE Computer Society, Kauai Island, Hawaii, USA, March 2005.

58. Avoine, G., E. Dysli and P. Oechslin, "Reducing Time Complexity in RFID Systems", B. Preneel and S. Tavares (Editors), *Selected Areas in Cryptography – SAC 2005*, Vol. 3897 of *Lecture Notes in Computer Science*, pp. 291–306, Springer,

Kingston, Canada, August 2005.

59. Hellman, M., "A cryptanalytic time-memory trade-off", *Information Theory, IEEE Transactions on*, Vol. 26, No. 4, pp. 401–406, Jul 1980.

60. Shaoying, C., Y. Li, T. Li and R. H. Deng, "Attacks and Improvements to an RFID Mutual Authentication Protocol and its Extensions", D. A. Basin, S. Capkun and W. Lee (Editors), *Proceedings of the 2nd ACM Conference on Wireless Network Security – WiSec'09*, pp. 51–58, ACM, ACM Press, Zurich, Switzerland, March 2009.

61. Akgun, M., M. Caglayan and E. Anarim, "Secure RFID Authentication with Efficient Key-Lookup", *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, 30 2009.

62. Kardaş, S., A. Levi and E. Murat, "Providing Resistance against Server Information Leakage in RFID Systems", *New Technologies, Mobility and Security – NTMS'11*, pp. 1–7, IEEE, IEEE Computer Society, Paris, France, February 2011.

63. Wu, J. and D. R. Stinson, "A Highly Scalable RFID Authentication Protocol", *Proceedings of the 14th Australasian Conference on Information Security and Privacy*, ACISP '09, pp. 360–376, Springer-Verlag, Berlin, Heidelberg, 2009.

64. Alomair, B., A. Clark, J. Cuellar and R. Poovendran, "Scalable RFID Systems: A Privacy-Preserving Protocol with Constant-Time Identification", *IEEE Transactions on Parallel and Distributed Systems*, 2011.

65. Molnar, D., A. Soppera and D. Wagner, "A Scalable, Delegatable Pseudonym Protocol Enabling Ownership Transfer of RFID Tags", B. Preneel and S. Tavares (Editors), *Selected Areas in Cryptography – SAC 2005*, Vol. 3897 of *Lecture Notes in Computer Science*, pp. 276–290, Springer, Kingston, Canada, August 2005.

66. Song, B., "RFID Tag Ownership Transfer", *Workshop on RFID Security – RFID-*

Sec'08, Budapest, Hungary, July 2008.

67. Saito, J., K. Imamoto and K. Sakurai, "Reassignment Scheme of an RFID Tags Key for Owner Transfer", T. Enokido, L. Yan, B. Xiao, D. Kim, Y. Dai and L. Yang (Editors), *Embedded and Ubiquitous Computing*, Vol. 3823 of *Lecture Notes in Computer Science*, pp. 1303–1312, Springer Berlin / Heidelberg, 2005.

68. Osaka, K., T. Takagi, K. Yamazaki and O. Takahashi, "An Efficient and Secure RFID Security Method with Ownership Transfer", *Computational Intelligence and Security, 2006 International Conference on*, Vol. 2, pp. 1090–1095, Nov 2006.

69. Koralalage, K. H., M. R. Selim, J. Miura, Y. Goto and J. Cheng, "POP Method: An Approach to Enhance the Security and Privacy of RFID Systems Used in Product Lifecycle with an Anonymous Ownership Transferring Mechanism", Y. Cho, R. L. Wainwright, H. Haddad, S. Y. Shin and Y. W. Koo (Editors), *Proceedings of the 2007 ACM Symposium on Applied Computing – SAC'07*, pp. 270–275, ACM, ACM Press, Seoul, Korea, March 2007.

70. Fouladgar, S. and H. Afifi, "An Efficient Delegation and Transfer of Ownership Protocol for RFID Tags", *First International EURASIP Workshop on RFID Technology*, Vienna, Austria, September 2007.

71. Fouladgar, S. and H. Afifi, "A Simple Privacy Protecting Scheme Enabling Delegation and Ownership Transfer for RFID Tags", *JCM*, Vol. 2, No. 6, pp. 6–13, 2007.

72. Peris-Lopez, P., J. C. Hernandez-Castro, J. M. Estevez-Tapiador, T. Li and Y. Li, "Vulnerability analysis of RFID protocols for tag ownership transfer", *Computer Networks, Elsevier*, Vol. 54, No. 9, pp. 1502[U+0096]–1508, June 2010.

73. Dimitriou, T., "RFID-DOT: RFID Delegation and Ownership Transfer made simple", *Conference on Security and Privacy for Communication Networks – SecureComm 2008*, pp. 1–8, IEEE, IEEE Computer Society, Istanbul, Turkey,

September 2008.

74. Ranasinghe, D., D. Engels and P. Cole, "Security and Privacy: Modest Proposals for Low-Cost RFID Systems", *Auto-ID Labs Research Workshop*, Zurich, Switzerland, September 2004.

75. Tuyls, P. and L. Batina, "RFID-Tags for Anti-Counterfeiting", D. Pointcheval (Editor), *The Cryptographers' Track at the RSA Conference – CT-RSA 2006*, Vol. 3860 of *Lecture Notes in Computer Science*, pp. 115–131, Springer, San Jose, California, USA, February 2006.

76. Bolotnyy, L. and G. Robins, "Physically Unclonable Function-Based Security and Privacy in RFID Systems", *International Conference on Pervasive Computing and Communications – PerCom 2007*, pp. 211–220, IEEE, IEEE Computer Society, New York City, New York, USA, March 2007.

77. Devadas, S., 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", *IEEE International Conference on RFID – IEEE RFID 2008*, pp. 58–64, April 2008.

78. Kulseng, L., Z. Yu, Y. Wei and Y. Guan, "Lightweight mutual authentication and ownership transfer for RFID systems", *INFOCOM'10: Proceedings of the 29th conference on Information communications*, pp. 251–255, IEEE Press, Piscataway, NJ, USA, 2010.

79. Kapoor, G. and S. Piramuthu, "Vulnerabilities in some recently proposed RFID ownership transfer protocols", *IEEE Communications Letters*, Vol. 14, No. 3, pp. 260–262, March 2010.

80. Masoumeh Safkhani, N. B., Majid Naderi and S. K. Sanadhya, "Cryptanalysis of Some Protocols for RFID Systems", Cryptology ePrint Archive, Report 2011/061, 2011.

81. Kardas S. and, M. a. K. M., Akgun and H. Demirci, "Cryptanalysis of Lightweight Mutual Authentication and Ownership Transfer for RFID Systems", *Lightweight Security Privacy: Devices, Protocols and Applications (LightSec), 2011 Workshop on*, pp. 20 –25, march 2011.

82. Choi, W., S. Kim, Y. Kim, Y. Park and K. Ahn, "PUF-based Encryption Processor for the RFID Systems", *Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on*, 29 2010.

83. Kardaşa, S., S. Çelika, M. Yıldıza and A. Levi, "PUF-enhanced offline RFID security and privacy", *Journal of Network and Computer Applications*, September 2012.

84. Kumar, S. and C. Paar, "Are Standards Compliant Elliptic Curve Cryptosystems Feasible on RFID?", *Workshop on RFID Security – RFIDSec'06*, Ecrypt, Graz, Austria, July 2006.

85. Hein, D., J. Wolkerstorfer and N. Felber, "ECC is Ready for RFID – A Proof in Silicon", *Workshop on RFID Security – RFIDSec'08*, Budapest, Hungary, July 2008.

86. Braun, M., E. Hess and B. Meyer, "Using Elliptic Curves on RFID Tags", *International Journal of Computer Science and Network Security*, Vol. 8, 2008.

87. Lee, Y. K., L. Batina and I. Verbauwhede, "EC-RAC (ECDLP Based Randomized Access Control): Provably Secure RFID Authentication Protocol", *IEEE International Conference on RFID – IEEE RFID 2008*, pp. 97–104, April 2008.

88. Ahamed, S., F. Rahman and E. Hoque, "ERAP: ECC Based RFID Authentication Protocol", *Future Trends of Distributed Computing Systems, 2008. FTDCS '08. 12th IEEE International Workshop on*, pp. 219 –225, 2008.

89. Lee, Y. K., L. Batina and I. Verbauwhede, "Untraceable RFID Authentication

Protocols: Revision of EC-RAC", *IEEE International Conference on RFID – IEEE RFID 2009*, IEEE, IEEE Computer Society, Orlando, Florida, USA, April 2009.

90. Chien, H.-Y. and C.-S. Laih, "ECC-based lightweight authentication protocol with untraceability for low-cost RFID", *Journal of Parallel and Distributed Computing*, Vol. 69, No. 10, pp. 848 – 853, 2009.

91. Martinez, S., M. Valls, C. Roig, J. Miret and F. Gin, "A Secure Elliptic Curve-Based RFID Protocol", *Journal of Computer Science and Technology*, Vol. 24, pp. 309–318, 2009.

92. Godor, G., N. Giczi and S. Imre, "Elliptic curve cryptography based mutual authentication protocol for low computational capacity RFID systems - performance analysis by simulations", *Wireless Communications, Networking and Information Security (WCNIS), 2010 IEEE International Conference on*, pp. 650 –657, 2010.

93. Góodor, G., P. Szendi and S. Imre, "Elliptic curve cryptography based authentication protocol for small computational capacity RFID systems", *Proceedings of the 6th ACM workshop on QoS and security for wireless and mobile networks*, Q2SWinet '10, pp. 98–105, ACM, New York, NY, USA, 2010.

94. Lee, Y. K., L. Batina, D. Singelée and I. Verbauwhede, "Wide-weak privacy-preserving RFID authetication protocols", *The 2nd International Conference on Mobile Lightweight Wireless Systems – Mobilight 2010*, Springer-Verlag, 2010.

95. Batina, L., J. Guajardo, T. Kerins, N. Mentens, P. Tuyls and I. Verbauwhede, "An Elliptic Curve Processor Suitable For RFID-Tags", Cryptology ePrint Archive, Report 2006/227, 2006.

96. Schnorr, C., "Efficient Identification and Signatures for Smart Cards", J.-J. Quisquater and J. Vandewalle (Editors), *Advances in Cryptology — EUROCRYPT '89*, Vol. 434, pp. 688–689, Springer Berlin Heidelberg, 1990.

97. Bock, H., M. Braun, M. Dichtl, E. Hess, J. Heyszl, W. Kargl, H. Koroschetz, B. Meyer and H. Seuschek, "A milestone towards RFID products offering asymmetric authentication based on elliptic curve cryptography", *In Proceedings of RFIDSec 2008, the 4th Workshop on RFID Security*, 2008.

98. Lee, Y. K., K. Sakiyama, L. Batina and I. Verbauwhede, "Elliptic-Curve-Based Security Processor for RFID", *Computers, IEEE Transactions on*, Vol. 57, No. 11, pp. 1514 –1527, 2008.

99. Luo, P., X. Wang, J. Feng and Y. Xu, "Low-power hardware implementation of ECC processor suitable for low-cost RFID tags", *Solid-State and Integrated-Circuit Technology, 2008. ICSICT 2008. 9th International Conference on*, pp. 1681 –1684, 2008.

100. Martinez, S., M. Valls, C. Roig, F. Gine and J. Miret, "An Elliptic Curve and Zero Knowledge Based Forward Secure RFID Protocol", *Workshop on RFID Security – RFIDSec'07*, Malaga, Spain, July 2007.

101. Batina, L., J. Guajardo, T. Kerins, N. Mentens, P. Tuyls and I. Verbauwhede, "Public-Key Cryptography for RFID-Tags", *International Workshop on Pervasive Computing and Communication Security – PerSec 2007*, pp. 217–222, IEEE, IEEE Computer Society, New York City, New York, USA, March 2007.

102. Bringer, J., H. Chabanne and T. Icart, "Cryptanalysis of EC-RAC, a RFID identification protocol", M. K. Franklin, L. C. K. Hui and D. S. Wong (Editors), *7th International Conference on Cryptology And Network Security – CANS'08*, Vol. 5339 of *Lecture Notes in Computer Science*, pp. 149–161, Springer, Hong Kong, China, December 2008.

103. Lee, Y. K., L. Batina, D. Singelée and I. Verbauwhede, "Low-Cost Untraceable Authentication Protocols for RFID", S. Wetzel, C. Nita-Rotaru and F. Stajano (Editors), *Proceedings of the 3rd ACM Conference on Wireless Network Security – WiSec'10*, pp. 55–64, ACM, ACM Press, Hoboken, New Jersey, USA, March

2010.

104. Fan, J., J. Hermans and F. Vercauteren, "On the claimed privacy of EC-RAC III", S. O. Yalcin (Editor), *Workshop on RFID Security – RFIDSec'10*, Vol. 6370 of *Lecture Notes in Computer Science*, pp. 66–74, Springer, Istanbul, Turkey, June 2010.

105. van Deursen, T. and S. Radomirović, "EC-RAC: Enriching a Capacious RFID Attack Collection", S. O. Yalcin (Editor), *Workshop on RFID Security – RFIDSec'10*, Vol. 6370 of *Lecture Notes in Computer Science*, pp. 75–90, Springer, Istanbul, Turkey, June 2010.

106. Yvo Desmedt, C. G. and S. Bengio, "Special Uses and Abuses of the Fiat-Shamir Passport Protocol", *Advances in Cryptology CRYPTO'87*, Vol. 293 of *Lecture Notes in Computer Science*, pp. 21–39, Springer-Verlag, Santa Barbara, California, USA, August 1987.

107. Hancke, G. P., "A Practical Relay Attack on ISO 14443 Proximity Cards", Manuscript, February 2005.

108. Kfir, Z. and A. Wool, "Picking Virtual Pockets Using Relay Attacks on Contactless Smartcard Systems", *Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm 2005*, pp. 47–58, IEEE, IEEE Computer Society, Athens, Greece, September 2005.

109. Drimer, S. and S. Murdoch, "Keep Your Enemies Close: Distance Bounding Against Smartcard Relay Attacks", *Proceedings of USENIX Security*, September 2007.

110. Hancke, G. P., "Design of a Secure Distance-Bounding Channel for RFID", *Journal of Network and Computer Applications*, May 2010.

111. Brands, S. and D. Chaum, "Distance-Bounding Protocols (Extended Abstract)",

EUROCRYPT, pp. 344–359, 1993.

112. Čapkun, S., L. Buttyán and J.-P. Hubaux, "SECTOR: Secure Tracking of Node Encounters in Multi-hop Wireless Networks", *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*, SASN '03, pp. 21–32, ACM, New York, NY, USA, 2003, http://doi.acm.org/10.1145/986858.986862.

113. Hancke, G. P. and M. Kuhn, "An RFID Distance Bounding Protocol", *Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm 2005*, pp. 67–73, IEEE, IEEE Computer Society, Athens, Greece, September 2005.

114. Singelee, D. and B. Preneel, "Location verification using secure distance bounding protocols", *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference*, Vol. 0, p. 840, 2005.

115. Bussard, L., *Trust Establishment Protocols for Communicating Devices*, Ph.D. Thesis, Eurecom-ENST, Paris, France, September 2004.

116. Reid, J., J. Gonzalez Nieto, T. Tang and B. Senadji, "Detecting Relay Attacks with Timing Based Protocols", QUT ePrint, Report 3264, 2006.

117. Piramuthu, S., "Protocols for RFID tag/reader authentication", *Decision Support Systems*, Vol. 43, No. 3, pp. 897–914, 2007.

118. Mitrokotsa, A., C. Dimitrakakis, P. Peris-Lopez and J. C. Hernandez-Castro, "Reid et al.'s Distance Bounding Protocol and Mafia Fraud Attacks over Noisy Channels", *IEEE Communications Letters*, Vol. 14, No. 2, pp. 121–123, February 2010.

119. Singelée, D. and B. Preneel, "Distance Bounding in Noisy Environments", F. Stajano, C. Meadows, S. Čapkun and T. Moore (Editors), *Security and Privacy in Ad-hoc and Sensor Networks – ESAS 2007*, Vol. 4572 of *Lecture Notes in Com-*

puter Science, pp. 101–115, Springer-Verlag, Cambridge, UK, July 2007.

120. Munilla, J. and A. Peinado, "Attacks on a Distance Bounding Protocol", *Computer Communications, Elsevier*, Vol. 33, No. 7, pp. 884–889, May 2010.

121. Tu, Y.-J. and S. Piramuthu, "RFID Distance Bounding Protocols", *First International EURASIP Workshop on RFID Technology*, Vienna, Austria, September 2007.

122. Kim, C. H., G. Avoine, F. Koeune, F.-X. Standaert and O. Pereira, "The Swiss-Knife RFID Distance Bounding Protocol", P. Lee and J. Cheon (Editors), *International Conference on Information Security and Cryptology – ICISC 2008*, Vol. 5461 of *Lecture Notes in Computer Science*, pp. 98–115, Springer, Seoul, Korea, December 2008.

123. Munilla, J. and A. Peinado, "Security Analysis of Tu and Piramuthu's Protocol", *New Technologies, Mobility and Security – NTMS'08*, pp. 1–5, IEEE, IEEE Computer Society, Tangier, Morocco, November 2008.

124. Kim, C. H. and G. Avoine, "RFID Distance Bounding Protocol with Mixed Challenges to Prevent Relay Attacks", J. A. Garay, A. Miyaji and A. Otsuka (Editors), *8th International Conference on Cryptology And Network Security – CANS'09*, Vol. 5888 of *Lecture Notes in Computer Science*, pp. 119–133, Springer, Kanazawa, Ishikawa, Japan, December 2009.

125. Peris-Lopez, P., J. C. Hernandez-Castro, J. M. Estevez-Tapiador and J. C. A. van der Lubbe, "Shedding Some Light on RFID Distance Bounding Protocols and Terrorist Attacks", arXiv.org, Computer Science, Cryptography and Security, 2009.

126. Trujillo Rasua, R., B. Martin and G. Avoine, "The Poulidor Distance-Bounding Protocol", S. O. Yalcin (Editor), *Workshop on RFID Security – RFIDSec'10*, Vol. 6370 of *Lecture Notes in Computer Science*, pp. 239–257, Springer, Istanbul,

Turkey, June 2010.

127. Avoine, G. and A. Tchamkerten, "An efficient distance bounding RFID authentication protocol: balancing false-acceptance rate and memory requirement", P. Samarati, M. Yung, F. Martinelli and C. A. Ardagna (Editors), *Information Security Conference – ISC'09*, Vol. 5735 of *Lecture Notes in Computer Science*, pp. 250–261, Springer, Pisa, Italy, September 2009.

128. Peris-Lopez, P., J. C. Hernandez-Castro, J. M. Estevez-Tapiador, E. Palomar and J. C. A. van der Lubbe, "Cryptographic Puzzles and Distance-bounding Protocols: Practical Tools for RFID Security", *IEEE International Conference on RFID – IEEE RFID 2010*, pp. 45–52, IEEE, IEEE Computer Society, Orlando, Florida, USA, April 2010.

129. Peris-Lopez, P., A. Orfila, E. Palomar and J. Hernandez-Castro, "A secure distance-based RFID identification protocol with an off-line back-end database", *Personal and Ubiquitous Computing*, Vol. 15, 2011.

130. Gürel, A. O., A. Arslan and M. Akgün, "Non-Uniform Stepping Approach to RFID Distance Bounding Problem", J. Garcia-Alfaro, G. Navarro-Arribas, A. Cavalli and J. Leneutre (Editors), *Fifth International Workshop on Data Privacy Management – DPM'10*, Vol. 6514 of *Lecture Notes in Computer Science*, pp. 64–78, Springer, Athens, Greece, September 2010.

131. Sohizadeh Abyaneh, M. R., "Security Analysis of two Distance-Bounding Protocols", *Workshop on RFID Security – RFIDSec'11*, Amherst, Massachusetts, USA, June 2011.

132. Avoine, G., 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, No. 2, pp. 289–317, March 2011.

133. Kara, O., S. Kardaş, M. A. Bingöl and G. Avoine, "Optimal Security Limits of

RFID Distance Bounding Protocols", S. O. Yalcin (Editor), *Workshop on RFID Security – RFIDSec'10*, Vol. 6370 of *Lecture Notes in Computer Science*, pp. 220–238, Springer, Istanbul, Turkey, June 2010.

134. Kardaş, S., M. S. Kiraz, M. A. Bingöl and H. Demirci, "A Novel RFID Distance Bounding Protocol Based on Physically Unclonable Functions", *Workshop on RFID Security – RFIDSec'11*, Amherst, Massachusetts, USA, June 2011.

135. Avoine, G., C. Lauradoux and B. Martin, "How Secret-sharing can Defeat Terrorist Fraud", *Proceedings of the 4th ACM Conference on Wireless Network Security – WiSec'11*, ACM, ACM Press, Hamburg, Germany, June 2011.

136. Karthikeyan, S. and M. Nesterenko, "RFID Security without Extensive Cryptography", *Workshop on Security of Ad Hoc and Sensor Networks – SASN'05*, pp. 63–67, ACM, ACM Press, Alexandria, Virginia, USA, November 2005.

137. Nguyen Duc, D., J. Park, H. Lee and K. Kim, "Enhancing Security of EPCglobal Gen-2 RFID Tag against Traceability and Cloning", *Symposium on Cryptography and Information Security*, Hiroshima, Japan, January 2006.

138. Chien, H.-Y. and C.-H. Chen, "Mutual Authentication Protocol for RFID Conforming to EPC Class 1 Generation 2 standards", *Computer Standars & Interfaces, Elsevier*, Vol. 29, No. 2, pp. 254–259, February 2007.

139. Juels, A. and S. Weis, "Authenticating Pervasive Devices with Human Protocols", V. Shoup (Editor), *Advances in Cryptology – CRYPTO'05*, Vol. 3126 of *Lecture Notes in Computer Science*, pp. 293–308, IACR, Springer, Santa Barbara, California, USA, August 2005.

140. Bringer, J., H. Chabanne and D. Emmanuelle, "HB^{++}: a Lightweight Authentication Protocol Secure against Some Attacks", *IEEE International Conference on Pervasive Services, Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing – SecPerU 2006*, IEEE, IEEE Computer Society, Lyon,

France, June 2006.

141. Piramuthu, S., "HB and Related Lightweight Authentication Protocols for Secure RFID Tag/Reader Authentication", *Collaborative Electronic Commerce Technology and Research – CollECTeR 2006*, Basel, Switzerland, June 2006.

142. "Class-1 Generation-2 UHF air interface protocol standard version 1.0.9", Available at `http://www.epcglobalinc.org/`,.

143. Burmester, M. and B. de Medeiros, "The Security of EPC Gen2 Compliant RFID Protocols", S. M. Bellovin, R. Gennaro, A. D. Keromytis and M. Yung (Editors), *Proceedings of the 6th International Conference on Applied Cryptography and Network Security – ACNS 2008*, Vol. 5037 of *Lecture Notes in Computer Science*, pp. 490–506, Springer, New York City, New York, USA, June 2008.

144. Yeh, K.-H. and N. Lo, "Improvement of Two Lightweight RFID Authentication Protocols", *Information Assurance and Security Letters – IASL 2010*, Vol. 1, pp. 6–11, 2010.

145. Peris-Lopez, P., J. C. Hernandez-Castro, J. M. Estevez-Tapiador and A. Ribagorda, "M2AP: A Minimalist Mutual-Authentication Protocol for Low-cost RFID Tags", J. Ma, H. Jin, L. T. Yang and J. J. P. Tsai (Editors), *International Conference on Ubiquitous Intelligence and Computing – UIC'06*, Vol. 4159 of *Lecture Notes in Computer Science*, pp. 912–923, Springer, Wuhan and Three Gorges, China, September 2006.

146. Bárász, M., B. Boros, P. Ligeti, K. Lója and D. Nagy, "Passive Attack Against the M2AP Mutual Authentication Protocol for RFID Tags", *First International EURASIP Workshop on RFID Technology*, Vienna, Austria, September 2007.

147. Peris-Lopez, P., J. C. Hernandez-Castro, J. M. Estevez-Tapiador and A. Ribagorda, "EMAP: An Efficient Mutual Authentication Protocol for Low-Cost RFID Tags", *OTM Federated Conferences and Workshop: IS Workshop – IS'06*,

Vol. 4277 of *Lecture Notes in Computer Science*, pp. 352–361, Springer, Montpellier, France, November 2006.

148. Li, T. and R. H. Deng, "Vulnerability Analysis of EMAP - An Efficient RFID Mutual Authentication Protocol", *Second International Conference on Availability, Reliability and Security – AReS 2007*, Vienna, Austria, April 2007.

149. Alomair, B., L. Lazos and R. Poovendran, "Passive Attacks on a Class of Authentication Protocols for RFID", K.-H. Nam and G. Rhee (Editors), *International Conference on Information Security and Cryptology – ICISC 2007*, Vol. 4817 of *Lecture Notes in Computer Science*, pp. 102–115, Springer, Seoul, Korea, November 2007.

150. Peris-Lopez, P., J. C. Hernandez-Castro, J. M. Estevez-Tapiador and A. Ribagorda, "LMAP: A Real Lightweight Mutual Authentication Protocol for Low-cost RFID tags", *Workshop on RFID Security – RFIDSec'06*, Ecrypt, Graz, Austria, July 2006.

151. Bárász, M., B. Boros, P. Ligeti, K. Lója and D. Nagy, "Breaking LMAP", *Conference on RFID Security*, Malaga, Spain, July 2007.

152. Li, T. and G. Wang, "Security Analysis of Two Ultra-Lightweight RFID Authentication Protocols", H. Venter, M. Eloff, L. Labuschagne, J. Eloff and R. Von Solms (Editors), *IFIP TC-11 22nd International Information Security Conference – SEC 2007*, Vol. 232 of *IFIP*, pp. 109–120, IFIP, Springer, Sandton, Gauteng, South Africa, May 2007.

153. Li, T., G. Wang and R. H. Deng, "Security Analysis on a Family of Ultralightweight RFID Authentication Protocols", *Journal of Software*, Vol. 3, No. 3, March 2008.

154. Chien, H.-Y. and C.-W. Huang, "Security of ultra-lightweight RFID authentication protocols and its improvements", *SIGOPS Oper. Syst. Rev.*, Vol. 41, pp.

83–86, July 2007.

155. Chien, H.-Y., "SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity", *IEEE Transactions on Dependable and Secure Computing*, Vol. 4, No. 4, pp. 337–340, December 2007.

156. D'Arco, P. and A. De Santis, "Weaknesses in a recent ultra-lightweight RFID authentication protocol", *Proceedings of the Cryptology in Africa 1st international conference on Progress in cryptology*, AFRICACRYPT'08, pp. 27–39, Springer-Verlag, Berlin, Heidelberg, 2008.

157. Hernandez-Castro, J. C., J. M. Estevez-Tapiador, P. Peris-Lopez and J.-J. Quisquater, "Cryptanalysis of the SASI Ultralightweight RFID Authentication Protocol with Modular Rotations", *International Workshop on Coding and Cryptography – WCC'09*, Ullensvang, Norway, May 2009.

158. Cao, T., E. Bertino and H. Lei, "Security Analysis of the SASI Protocol", *Dependable and Secure Computing, IEEE Transactions on*, Vol. 6, No. 1, pp. 73 –77, 2009.

159. Phan, R. C.-W., "Cryptanalysis of a New Ultralightweight RFID Authentication Protocol - SASI", *IEEE Transactions on Dependable and Secure Computing*, Vol. 99, No. 1, 2008.

160. Sun, H., W. Ting and K. Wang, "On the Security of Chien's Ultralightweight RFID Authentication Protocol", *Dependable and Secure Computing, IEEE Transactions on*, 2010.

161. Li, T. and G. Wan, "SLMAP - A Secure ultra-Lightweight RFID Mutual Authentication Protocol", *Advances in Cryptology – CHINACRYPT'07*, Lecture Notes in Computer Science, pp. 19–22, Springer, Cheng Du, China, October 2007.

162. Hernandez-Castro, J. C., J. E. Tapiador, P. Peris-Lopez, J. A. Clark and E.-G.

Talbi, "Metaheuristic Traceability Attack against SLMAP, an RFID Lightweight Authentication Protocol", *Proceedings of the 23rd IEEE International Parallel and Distributed Processing Symposium – IPDPS 2009*, IEEE, IEEE Computer Society, Rome, Italy, May 2009.

163. Peris-Lopez, P., J. C. Hernandez-Castro, J. M. Estevez-Tapiador and A. Ribagorda, "Advances in Ultralightweight Cryptography for Low-cost RFID Tags: Gossamer Protocol", K.-I. Chung, K. Sohn and M. Yung (Editors), *Workshop on Information Security Applications – WISA'08*, Vol. 5379 of *Lecture Notes in Computer Science*, pp. 56–68, Springer, Jeju Island, Korea, September 2008.

164. Zeeshan Bilal, A. M. and F. Kausar, "Security Analysis of Ultra-lightweight Cryptographic Protocol for Low-cost RFID Tags: Gossamer Protocol", *International Conference on Network-Based Information Systems – NBIS'09*, pp. 260–267, IEEE, IEEE Computer Society, Indianapolis, Indiana, USA, August 2009.

165. Billet, O., J. Etrog and H. Gilbert, "Lightweight Privacy Preserving Authentication for RFID Using a Stream Cipher", S. Hong and T. Iwata (Editors), *Fast Software Encryption – FSE'10*, Vol. 6147 of *Lecture Notes in Computer Science*, pp. 55–74, Springer, Seoul, Korea, February 2010.

166. Hopper, N. J. and M. Blum, "Secure Human Identification Protocols", *ASIACRYPT*, pp. 52–66, 2001.

167. Gilbert, H., M. Robshaw and H. Sibert, "An Active Attack Against HB^+ – A provably Secure Lightweight Authentication Protocol", *IET Electronics Letters*, Vol. 41, No. 21, pp. 1169–1170, October 2005.

168. Nguyen Duc, D. and K. Kim, "Securing HB^+ against GRS Man-in-the-Middle Attack", *Proceedings of the Symposium on Cryptography and Information Security (SCIS2007)*, 2007.

169. Piramuthu, S. and Y. Tu, "Modified HB Authentication Protocol", *Western Eu-*

ropean Workshop on Research in Cryptology, Germany, July 2007.

170. Munilla, J. and A. Peinado, "HB-MP: A further step in the HB-family of lightweight authentication protocols", *Computer Networks*, Vol. 51, No. 9, pp. 2262 – 2267, 2007.

171. Gilbert, H., M. J. B. Robshaw and Y. Seurin, "HB$^{\#}$: Increasing the Security and Efficiency of HB^{+}", *EUROCRYPT*, pp. 361–378, 2008.

172. Ouafi, K., R. Overbeck and S. Vaudenay, "On the Security of HB# against a Man-in-the-Middle Attack", J. Pieprzyk (Editor), *Advances in Cryptology – Asiacrypt 2008*, Vol. 5350 of *Lecture Notes in Computer Science*, pp. 108–124, Springer, Melbourne, Australia, December 2008.

173. Hammouri, G. and B. Sunar, "PUF-HB: A Tamper-Resilient HB Based Authentication Protocol", *ACNS*, pp. 346–365, 2008.

174. Leng, X., K. Mayes and K. Markantonakis, "HB-MP+ Protocol: An Improvement on the HB-MP Protocol", *IEEE International Conference on RFID – IEEE RFID 2008*, pp. 118–124, April 2008.

175. Godor, G., M. Antal and S. Imre, "Mutual Authentication Protocol for Low Computational Capacity RFID Systems", *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*, pp. 1–5, Nov 2008.

176. Go´dor, G. and S. Imre, "Security Analysis of the Simple Lightweight Authentication Protocol", *Networks (ICN), 2010 Ninth International Conference on*, pp. 231–236, April 2010.

177. Gong, L., R. M. Needham and R. Yahalom, "Reasoning about Belief in Cryptographic Protocols", *IEEE Symposium on Security and Privacy*, pp. 234–248, 1990.

178. Luo, Z., T. Chan and J. Li, "A lightweight mutual authentication protocol for

RFID networks", *e-Business Engineering, 2005. ICEBE 2005. IEEE International Conference on*, pp. 620–625, Oct 2005.

179. Gódor, G. and M. Antal, "Improved Lightweight Mutual Authentication Protocol for RFID Systems", Z. Mammeri (Editor), *Wireless and Mobile Networking*, Vol. 284 of *IFIP International Federation for Information Processing*, pp. 471–482, Springer US, 2008.

180. Gao, L., M. Ma, Y. Shu and Y. Wei, "An ultralightweight RFID authentication protocol with CRC and permutation", *Journal of Network and Computer Applications*, , No. 0, pp. –, 2013, http://www.sciencedirect.com/science/article/pii/S1084804513002269.

181. Pang, L., H. Li, L. He, A. Alramadhan and Y. Wang, "Secure and efficient lightweight RFID authentication protocol based on fast tag indexing", *International Journal of Communication Systems*, 2013, http://dx.doi.org/10.1002/dac.2538.

182. Masoumeh Safkhani, N. B., "For an EPC-C1 G2 RFID compliant Protocol, CRC with Concatenation : No; PRNG with Concatenation : Yes", Cryptology ePrint Archive, Report 2013/490, 2013, http://eprint.iacr.org/.

183. Han, D. and D. Kwon, "Vulnerability of an RFID Authentication Protocol Conforming to EPC Class 1 Generation 2 Standards", *Comput. Stand. Interfaces*, Vol. 31, No. 4, pp. 648–652, Jun. 2009.

184. Peris-Lopez, P., J. C. Hernandez-Castro, J. M. Estevez-Tapiador and A. Ribagorda, "Cryptanalysis of a Novel Authentication Protocol Conforming to EPC-C1G2 Standard", *Comput. Stand. Interfaces*, Vol. 31, No. 2, pp. 372–380, Feb. 2009.

185. Yeh, T.-C., Y.-J. Wang, T.-C. Kuo and S.-S. Wang, "Securing RFID systems conforming to {EPC} Class 1 Generation 2 standard", *Expert Systems with Ap-*

plications, Vol. 37, No. 12, pp. 7678 – 7683, 2010.

186. Yoon, E.-J., "Improvement of the securing RFID systems conforming to {EPC} Class 1 Generation 2 standard", *Expert Systems with Applications*, Vol. 39, No. 1, pp. 1589 – 1594, 2012.

187. Safkhani, M., N. Bagheri, S. S. Kumar and M. Naderi, "Cryptanalysis of improved Yeh et al.'s authentication Protocol: An EPC Class-1 Generation-2 standard compliant protocol", Cryptology ePrint Archive, Report 2011/426, 2011.

188. Peris-Lopez, P., A. Orfila, A. Mitrokotsa and J. C. van der Lubbe, "A Comprehensive RFID Solution to Enhance Inpatient Medication Safety", *International Journal of Medical Informatics*, October 2010.

189. Safkhani, M., N. Bagheri and M. Naderi, "A note on the security of IS-RFID, an inpatient medication safety", *I. J. Medical Informatics*, Vol. 83, No. 1, pp. 82–85, 2014, `http://dx.doi.org/10.1016/j.ijmedinf.2013.04.003`.

190. Cheng, Z.-Y., Y. Liu, C.-C. Chang and S.-C. Chang, "Authenticated RFID security mechanism based on chaotic maps", *Security and Communication Networks*, Vol. 6, No. 2, pp. 247–256, 2013, `http://dx.doi.org/10.1002/sec.709`.

191. Benssalah, M., M. Djeddou and K. Drouiche, "Security enhancement of the authenticated RFID security mechanism based on chaotic maps", *Security and Communication Networks*, 2014, `http://dx.doi.org/10.1002/sec.946`.

192. Wang, X. and J. Zhao, "An improved key agreement protocol based on chaos", *Communications in Nonlinear Science and Numerical Simulation*, Vol. 15, No. 12, pp. 4052 – 4057, 2010, `http://www.sciencedirect.com/science/article/pii/S1007570410001103`.

193. Burrows, M., M. Abadi and R. Needham, "A Logic of Authentication", *ACM Trans. Comput. Syst.*, Vol. 8, No. 1, pp. 18–36, Feb. 1990.

194. Berbain, C., O. Billet, J. Etrog and H. Gilbert, "An Efficient Forward Private RFID Protocol", *Proceedings of the 16th ACM Conference on Computer and Communications Security*, CCS '09, pp. 43–53, ACM, New York, NY, USA, 2009, http://doi.acm.org/10.1145/1653662.1653669.

195. Yoon, E.-J. and I.-S. Jeon, "An efficient and secure Diffie–Hellman key agreement protocol based on Chebyshev chaotic map", *Communications in Nonlinear Science and Numerical Simulation*, Vol. 16, No. 6, pp. 2383 – 2389, 2011.

196. Guo, C. and C.-C. Chang, "Chaotic maps-based password-authenticated key agreement using smart cards", *Communications in Nonlinear Science and Numerical Simulation*, Vol. 18, No. 6, pp. 1433 – 1440, 2013.

197. Fateman, R. J., "Lookup Tables, Recurrences and Complexity", *Proceedings of the ACM-SIGSAM 1989 International Symposium on Symbolic and Algebraic Computation*, ISSAC '89, pp. 68–73, ACM, New York, NY, USA, 1989, http://doi.acm.org/10.1145/74540.74549.

198. Sadoudi, S., M. Azzaz, C. Tanougast and A. Dandache, "Real time hardware implementation of a new Duffing's chaotic attractor", *Electronics, Circuits, and Systems, 2009. ICECS 2009. 16th IEEE International Conference on*, pp. 559–562, Dec 2009.

199. Chung, H. L. H., *Chaos Based RFID Authentication Protocol*, Master thesis, University of Ottawa, Ottawa, Canada, 2013.

200. Bringer, J., H. Chabanne and T. Icart, "Improved Privacy of the Tree-Based Hash Protocols Using Physically Unclonable Function", R. Ostrovsky, R. D. Prisco and I. Visconti (Editors), *Proceedings of the 6th International Conference on Security and Cryptography for Networks – SCN'08*, Vol. 5229 of *Lecture Notes in Computer Science*, pp. 77–91, Springer, Amalfi, Italy, August 2008.

201. Sadeghi, A.-R., I. Visconti and C. Wachsmann, "PUF-Enhanced RFID Security

and Privacy", *Secure Component and System Identification – SECSI'10*, Cologne, Germany, April 2010.

202. Avoine, G., M. A. Bingol, X. Carpent and S. B. Ors Yalcin, "Privacy-friendly Authentication in RFID Systems: On Sub-linear Protocols based on Symmetric-key Cryptography", *IEEE Transactions on Mobile Computing*, Vol. 99, September 2012.

203. Ravikanth, P. S., *Physical one-way functions*, Ph.D. Thesis, Massachusetts Institute of Technology, 2001.

204. Armknecht, F., R. Maes, A. Sadeghi, O.-X. Standaert and C. Wachsmann, "A Formalization of the Security Features of Physical Functions", *Security and Privacy (SP), 2011 IEEE Symposium on*, pp. 397–412, May 2011.

205. Tuyls, P., B. Skoric and T. Kevenaar, *Security with Noisy Data: Private Biometrics, Secure Key Storage and Anti-Counterfeiting*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

206. Katzenbeisser, S., U. Kocabas, V. Rožić, A.-R. Sadeghi, I. Verbauwhede and C. Wachsmann, "PUFs: Myth, Fact or Busted? A Security Evaluation of Physically Unclonable Functions (PUFs) Cast in Silicon", E. Prouff and P. Schaumont (Editors), *Cryptographic Hardware and Embedded Systems – CHES 2012*, Vol. 7428 of *Lecture Notes in Computer Science*, pp. 283–301, Springer Berlin Heidelberg, 2012.

207. Gassend, B., D. Clarke, M. van Dijk and S. Devadas, "Silicon Physical Random Functions", *Proceedings of the 9th ACM Conference on Computer and Communications Security*, CCS '02, pp. 148–160, ACM, New York, NY, USA, 2002.

208. Dodis, Y., L. Reyzin and A. Smith, "Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data", C. Cachin and J. Camenisch (Editors), *Advances in Cryptology - EUROCRYPT 2004*, Vol. 3027 of *Lecture Notes*

in Computer Science, pp. 523–540, Springer Berlin Heidelberg, 2004.

209. Verbauwhede, I. and R. Maes, "Physically Unclonable Functions: Manufacturing Variability As an Unclonable Device Identifier", *Proceedings of the 21st Edition of the Great Lakes Symposium on Great Lakes Symposium on VLSI*, GLSVLSI '11, pp. 455–460, ACM, New York, NY, USA, 2011.

210. Lee, J., D. Lim, B. Gassend, G. Suh, M. van Dijk and S. Devadas, "A technique to build a secret key in integrated circuits for identification and authentication applications", *VLSI Circuits, 2004. Digest of Technical Papers. 2004 Symposium on*, pp. 176–179, June 2004.

211. Suh, G. and S. Devadas, "Physical Unclonable Functions for Device Authentication and Secret Key Generation", *Design Automation Conference, 2007. DAC '07. 44th ACM/IEEE*, pp. 9–14, June 2007.

212. Ozturk, E., G. Hammouri and B. Sunar, "Towards Robust Low Cost Authentication for Pervasive Devices", *Pervasive Computing and Communications, 2008. PerCom 2008. Sixth Annual IEEE International Conference on*, pp. 170–178, March 2008.

213. Guajardo, J., S. Kumar, G.-J. Schrijen and P. Tuyls, "FPGA Intrinsic PUFs and Their Use for IP Protection", P. Paillier and I. Verbauwhede (Editors), *Cryptographic Hardware and Embedded Systems - CHES 2007*, Vol. 4727 of *Lecture Notes in Computer Science*, pp. 63–80, Springer Berlin Heidelberg, 2007.

214. Su, Y., J. Holleman and B. Otis, "A Digital 1.6 pJ/bit Chip Identification Circuit Using Process Variations", *Solid-State Circuits, IEEE Journal of*, Vol. 43, No. 1, pp. 69–77, Jan 2008.

215. van der Leest, V., G.-J. Schrijen, H. Handschuh and P. Tuyls.

216. Tuyls, P. and L. Batina, "RFID-Tags for Anti-counterfeiting", D. Pointcheval

(Editor), *Topics in Cryptology – CT-RSA 2006*, Vol. 3860 of *Lecture Notes in Computer Science*, pp. 115–131, Springer Berlin Heidelberg, 2006.

217. Skoric, B., G.-J. Schrijen, P. Tuyls, T. Ignatenko and F. Willems, "Secure Key Storage with PUFs", P. Tuyls, B. Skoric and T. Kevenaar (Editors), *Security with Noisy Data*, pp. 269–292, Springer London, 2007.

218. Busch, H., S. Katzenbeisser and P. Baecher, "Information Security Applications", pp. 296–308, Springer-Verlag, Berlin, Heidelberg, 2009.

219. Guajardo, J., S. Kumar, G.-J. Schrijen and P. Tuyls, "FPGA Intrinsic PUFs and Their Use for IP Protection", P. Paillier and I. Verbauwhede (Editors), *Cryptographic Hardware and Embedded Systems - CHES 2007*, Vol. 4727 of *Lecture Notes in Computer Science*, pp. 63–80, Springer Berlin Heidelberg, 2007.

220. Herder, C., M.-D. Yu, F. Koushanfar and S. Devadas, "Physical Unclonable Functions and Applications: A Tutorial", *Proceedings of the IEEE*, Vol. 102, No. 8, pp. 1126–1141, Aug 2014.

221. "Verayo, Inc.", `http://verayo.com/solutions.php`, accessed: 2014-09-30.

222. Avoine, G., B. Martin and T. Martin, "Tree-Based RFID Authentication Protocols Are Definitively Not Privacy-Friendly", S. O. Yalcin (Editor), *Workshop on RFID Security – RFIDSec'10*, Vol. 6370 of *Lecture Notes in Computer Science*, pp. 103–122, Springer, Istanbul, Turkey, June 2010.