PUF-ENHANCED SCALABLE RFID SECURITY AND PRIVACY

by

Işıl Kurt

B.S., Computer Engineering, Boğaziçi University, 2017

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

Graduate Program in Computer Engineering

Boğaziçi University

2021

ACKNOWLEDGEMENTS

I am deeply grateful to my supervisor Prof. Fatih Alagöz and my co-supervisor Assist. Prof. Mete Akgün.

ABSTRACT

PUF-ENHANCED SCALABLE RFID SECURITY AND PRIVACY

Radio Frequency Identification (RFID) is a very promising technology that enables the automatic identification of objects. However, it has some challenging issues such as scalability. Almost all of the existing solutions require the back end server to work linear in the number of tags in order to identify a single tag. There are some proposals providing O(1) or O(log n) identification complexity, yet, most of them are susceptible to serious attacks including RFID tag corruption attacks. Besides, only a few of them take attacks into consideration for the reader side. Nevertheless, they do not have the desired level of privacy to provide resistance against compromising attacks on both the tag side and the reader side.

In this research, we analyze the existing RFID protocols and specify the open problems that cause scalability and privacy concerns. We extend the predefined privacy model of Vaudenay by considering reader side attacks, and then propose a privacy-preserving RFID authentication protocol that does not require any search operation in the back end. It provides resistance against tag and reader corruption attacks by using Physically Unclonable Functions (PUFs) as secure storage to keep secrets of the system. Our protocol provides destructive privacy for tag holders in case of reader corruption attacks without any conditions. Additionally, our protocol allows readers to work offline by transferring the necessary database records to them and still provides destructive privacy in case of corruption of offline readers. To the best of our knowledge, it is the first protocol providing such a high privacy level without lookup property.

ÖZET

PUF TABANLI ÖLÇEKLENEBİLİR RFID SİSTEMLERİNİN GÜVENLİĞİ VE MAHREMİYETİ

Radyo Frekansı ile Tanımlama (RFID), nesnelerin otomatik olarak tanımlanmasını sağlayan umut verici bir teknolojidir. Ancak, ölçeklenebilirlik gibi bazı zorlu sorunları vardır. Mevcut çözümlerin neredeyse tamamı, tek bir etiketi tanımlamak için sunucunun etiket sayısında doğrusal çalışmasını gerektirir. O(1) veya O(log n) tanımlama karmaşıklığı sağlayan bazı çalışmalar vardır. Ancak bunların çoğu, RFID etiketi bozma saldırıları da dahil olmak üzere ciddi saldırılara açıktır. Ayrıca, sadece birkaç çalışmada okuyucu tarafına yönelik saldırılar düşünülmüş. Yine de hem etiket tarafında hem de okuyucu tarafında ele geçirme saldırılara karşı direnç sağlamak için istenilen gizlilik düzeyine sahip değiller.

Bu araştırmada, mevcut RFID protokollerini analiz ederek ölçeklenebilirlik ve gizlilik endişelerine neden olan açık noktaları belirlendi. Vaudenay'ın önceden tanımlanmış gizlilik modelini, okuyucu tarafı saldırılarını dikkate alarak genişletildi ve ardından arkayüzde herhangi bir arama işlemi gerektirmeyen, gizliliği koruyan bir RFID kimlik doğrulama protokolü önerildi. Sistemin sırlarını saklamak için Fiziksel klonlanamayan fonksiyonlar (PUF'ler) güvenli depolama olarak kullanılarak, etiket ve okuyucu bozma saldırılarına karşı direnç sağlar. Protokolümüz, herhangi bir koşul olmaksızın okuyucu bozma saldırıları durumunda etiket sahipleri için yıkıcı gizlilik sağlar. Ek olarak, protokolümüz okuyucuların gerekli veritabanı kayıtlarını onlara aktararak çevrimdışı çalışmasına izin verir ve çevrimdışı okuyucuların bozulması durumunda yine de yıkıcı gizlilik sağlar. Bildiğimiz kadarıyla, protokolümüz, arama özelliği olmadan bu kadar yüksek bir gizlilik seviyesi sağlayan ilk protokoldür.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

$\mathbf{A}$	An adversary
a_j	One of the secret values of a reader R_j to generate S_1
$\mathcal{B}$	A blinder
b_j	One of the secret values of a reader R_j to generate S_1
$\mathcal{C}$	A challenger
c	A challenge that input and condition are given to a PUF
c_j	One of the secret values of a reader R_j to generate S_1
$DATA_i$	Information about a tag T_i
d_j	One of the secret values of a reader R_j to generate S_2
e_j	One of the secret values of a reader R_j to generate S_2
f_j	One of the secret values of a reader R_j to generate S_2
H	A hash function
ID_i	The identifier of a tag T_i
k_1, k_2	Temporary values that are used in the protocol
K_P	Public key
K_S	Private key
l	The length of the variable
m	A message
M_1	A message of a reader to send to a tag
M_2, M_3, M_4	A message set of a tag to send to a reader
$\mathbf{N}$	A set of natural numbers
P_i	The PUF of a tag T_i
P_j	The PUF of a tag R_j
r	A response that a PUF generates
r_1	A nonce of a reader R_j
r_2	A nonce of a reader R_j
r_3	A nonce of a tag T_i
s	A session

S_1	The master secret 1
S_2	The master secret 2
$vtag$	The identifier of a drawn tag
v_i	One of the secret values of a tag T_i to generate S_1
u_i	One of the secret values of a tag T_i to generate S_1
w_i	One of the secret values of a tag T_i to generate S_1
x_i	One of the secret values of a tag T_i to generate S_2
y_i	One of the secret values of a tag T_i to generate S_2
z_i	One of the secret values of a tag T_i to generate S_2
Q_{max}	The maximum number of queries
$\mathcal{G}$	A game
$\mathcal{O}()$	The notation for the complexity
π	A protocol instance
$\perp$	The truth value 'false'
$\oplus$	XOR operator
$\in$	Random choice operator

LIST OF ACRONYMS/ABBREVIATIONS

DoS	Denial of Service
MUX	Multiplexer
P	An ideal PUF
PKC	Public-key cryptography
POK	Physical Obfuscated Key
Pr	Probability
PUF	Physically Unclonable Function
RFID	Radio Frequency Identification
RFID IC	Radio Frequency Identification Integrated Circuit
SRAM PUF	SRAM memory-based PUF

1. INTRODUCTION

Radio Frequency Identification (RFID) technology that is increasingly widespread offers effective solutions in many sectors. RFID allows the identification of objects by using radio signals. Tags, readers, and a back end server are the basic components of an RFID system. General working principle of this technology are as follows: electromagnetic waves sent by the reader are taken by the tag as energy. In this way, the tag is activated, and data transfer from the tag to the reader is carried out.

With the expansion of RFID-based applications, many people carry RFID-enabled devices such as travel cards, credit cards, etc. Malicious people can make use of these devices in order to track other people. The only thing they need is a reader which is able to detect RFID-enabled devices. Even when a single tag is tracked, this violates the user's privacy. Therefore, the most important issue to be solved is private identification. Private identification can be achieved by using Public-key cryptography (PKC). However, RFID tags do not have sufficient computational capabilities for PKC. Therefore, symmetric-key cryptography has to be used to design identification protocols.

In the RFID identification protocol, RFID tags should randomize their responses in order to provide privacy. The reader searches on its database in order to find the owner of the randomized response. When there are more tags in the system, the search process increases the duration of identification. As a result, the scalability problem arises in large-scale RFID systems. Many studies in the literature require a linear search process.

There are several important solutions to solve the scalability problem. One of them is the tree-based protocol proposed by Molnar and Wagner [1]. It reduces identification complexity from O(N) to O(log N). However, it is vulnerable to tag compromising attacks [2].

Bringer *et al.* [3] modified the tree-based protocol in [1] by using Physical Obfuscated Keys (POKs), and they increased the resistance of tags against corruption. Thus, an adversary that is corrupting a tag cannot learn the secrets of the tag. Avoine *et al.* [2] proposed a time-memory trade-off approach that reduces identification complexity to $O(N^{2/3})$. Wu and Stinson [4] proposed a scalable RFID protocol. The proposed protocol provides security and privacy by using the difficulty of reconstructing a polynomial with noisy data. In this protocol, the maximum number of queries that a tag will answer correctly is limited to Q_{max}. That means an adversary querying a tag for Q_{max} times repeatedly can trace the tag. Alomair *et al.* [5] proposed an RFID protocol with constant-time identification. They designed a special database infrastructure at the back end server. In [6], a traceability attack on Alomair *et al.*'s protocol was presented. Akgün *et al.* [7] proposed the first Physically Unclonable Functions (PUFs) based RFID authentication protocol that provides $\mathsf{NARROW-DESTRUCTIVE}$ privacy with $\mathcal{O}(1)$ identification complexity. However, Tiplea and Hristea [8] have shown that the last interaction of a corrupted tag is defined by an adversary in the proposed attack. Therefore, Akgün *et al.* do not provide the claimed level of privacy.

Compromising attacks on the reader side is another problem of the RFID protocols. If a protocol does not have any security mechanism against the compromising attacks on the reader side, the adversary can get important information like secret keys and then impersonate the readers and tags. Most of the current studies emphasize the attacks against the tag side. Only Kardaş *et al.* [9] considered compromising reader side attacks on their protocols. Kardaş *et al.* categorized RFID systems into two groups in terms of the connection of the reader and back end. If all readers of the system are connected to the back end, it is an online RFID system. If the readers connect to the back end only for synchronization of tag and reader information, it is an offline RFID system. Their protocol was secure against the corruption of offline readers under two conditions that are very difficult to meet. Indeed, this protocol does not solve reader side compromising, it only proposed a mechanism in which compromised offline readers are no longer a threat to the security of the entire system after certain conditions are met. On the other hand, the other solutions do not even consider compromising reader side attacks.

1.1. Contribution and Outline

In this paper, we examine the previous studies in terms of their strengths and weaknesses, and we find three problems to improve and get a secure and scalable RFID protocol. These three problems are as follows :

- Avoine *et al.* [6] present an open question of whether it is possible to design such a protocol without any loss of security or privacy because they stated that a protocol using only one master key has constant-time identification, but it does not provide privacy and security as soon as one tag is compromised. Akgün *et al.* [7] addresses the open question in [6]. However, it does not provide privacy against reader side compromising attacks and proposed attacks of Tiplea and Hristea [8]. Besides, Hristea and Ţiplea [10] claimed that the stateful RFID schemes with constant tag identifiers do not provide any of kind privacy in Vaudenay's model. We addressed the open question in [6]. Our protocol does not require the search process to identify tags by utilizing these master keys thus, it provides constant-time identification and addresses the open question in [6]. Contrary to Hristea and Ţiplea's claim, we proposed a secure protocol in Vaudenay's model with constant-time identification.

- As we mentioned before, Kardaş *et al.* [9] considered compromising reader side attacks on their protocols for only offline RFID systems. Therefore, their protocol has partial reader side privacy. No other protocol has addressed this issue. We consider the reader and back end server together and call them reader side. Considering this definition, we propose a destructive private RFID authentication protocol that is secure against both the reader side and tag-side compromise attacks. Our protocol utilizes Physically Unclonable Functions (PUFs) on both the reader side and tag-side in order to provide security to master keys that are shared by all tags and readers. In that way, it provides security against reader side compromise attacks using only PUFs without having to meet any conditions.

Besides, our protocol can be used for all RFID systems, regardless of the type of the RFID system. By transferring database records to readers, offline RFID systems can provide the same privacy level.

- RFID protocols require tag authentication on the reader side and reader authentication on the tag side to maintain key or data synchronization between the tag and the reader. In the protocol we offer, there is no need for key synchronization thanks to the master key shared by the whole system. Therefore, there is no need for reader authentication and verification message from the reader to the tag. It means that our protocol does not have an extra communication step for reader authentication on the tag side. Therefore, the proposed attack on the last interaction step in Kardaş *et al.*'s protocol [11] and Akgün *et al.*'s protocol [7] that is mentioned in Tiplea and Hristea [8] is not valid for our proposal.

To the best of our knowledge, our protocol is the first protocol that is secure against both tag and reader side compromise attacks and provides destructive privacy without lookup property for large scale RFID systems.

The rest of the paper is organized as follows: Chapter 2 gives some preliminaries about Radio Frequency Identification (RFID), Physically Unclonable Functions (PUFs), and Hash functions. In Chapter 3, we review some of the previous works with novel approaches and problems. We state open problems of RFID systems to be solved. Chapter 4 gives privacy and security definitions that are used to analyze our protocol. Then, Chapter 5 introduces our new proposed protocol and its analysis. In Section 5.1 we give notations that are used in describing the protocol. Section 5.2 describes the proposed protocol. In Section 5.3, we give a security and privacy analysis of our proposal. In Section 6, we compare proposed protocols. Finally, in Section 7, we conclude the paper. It summarizes the results we obtained in this study.

2. BACKGROUND INFORMATION AND PRELIMINERIES

2.1. Radio Frequency Identification (RFID)

Radio-Frequency Identification (RFID) is a technology that uses radio waves to identify and track objects. This technology offers effective solutions in many areas such as the military, healthcare, security. Industries use RFID systems to perform such tasks as road payment [12], billing systems in malls, ticketing systems [13], personal access control, and other tasks.

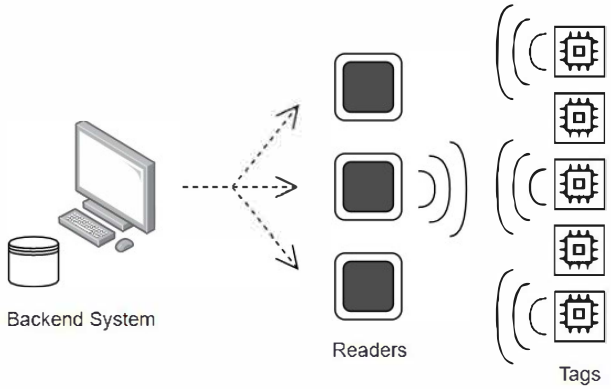

Figure 2.1. An RFID System.

An RFID system consists of three components: a back end server, readers, and tags (see Figure 2.1). When an RFID reader sends out radio waves, the RFID tag gets these waves and is activated. The RFID tags have their own data such as unique identifiers. It transacts and sends the answer back to the reader.

There are two types of RFID systems in terms of the connection of reader and back end. If all readers of the system are connected to the back end, it is an online RFID system. In online RFID systems, readers do not have an additional database. The back end server stores all information on its database.

Since readers do not have an additional database, they transfer the information to the back end server. It means that the back end server always is kept updated. If readers connect to the back end only for synchronization of tag and reader information, it is an offline RFID system. In this system, readers have additional databases besides the back end server. The synchronization is done at regular intervals. Until the synchronization is finished, readers store and use their own database. At the synchronization time, the back end server and all readers' databases are updated.

RFID tags are mainly classified into two groups: passive and active tags. Whereas the former ones are powered by energy that is generated by the reader, the latter ones are powered by a battery. Certain characteristics such as the range, costs, memory capacity, security issues of RFID tags affects its the extensive usage. To illustrate, passive tags are more costly-effective than the active ones, therefore, they are widely used for the debit card system of public transit uses and so as to the process, when a passenger scans the debit card on the post machine, the card is powered and as a result, the payment occurs. As for the active tags, they have much greater range than the passive tags and that's why, they are extensively used for items where accurate location tracking is necessary, for example, cargo containers.

The first device that uses the first state of RFID technology was invented in 1948 by Harry Stockman [14] and it was used to recognize friendly and hostile aircraft by Germany in World War II by using transponders. Development on RFID technology continued until the invention of the initial device that uses the first ancestor of modern RFID [15]. Mario Cardullo's device [15] was the passive radio transponder with 16-bit memory. It was convenient to use this device in many areas such as transportation. Afterward, many improvements are done to RFID systems: active and semi-passive tags were invented and came into use, the memory and the speed of communication were improved, online and offline RFID systems were built, and RFID systems evolved, as a result, to solve many problems in some areas such as manifactoring, transaction systems, and people management.

2.2. Physically Unclonable Functions (PUFs)

Physically Unclonable Function (PUF) includes an unclonable noisy function that is embedded into a device physically [16]. A PUF is a physical object that creates responses based on physical properties of the circuit such as gate and wire delays. It behaves as a unique identifier for the given input and conditions. It maps a challenge c to a response r.

It is impossible to duplicate a PUF because it uses randomness coming from the manufacturing process. Basically, for the same challenge c, PUF generates slightly different responses that are used by a small circuit, called Fuzzy Extractor, to map to a unique value r (see Figure 2.2). On the other hand, two different PUFs generate different responses for the same challenge with overwhelming probability (see Figure 2.2) which means that PUFs having the same logical circuit design produce different responses. A PUF has the following characteristics [17]:

- It is not possible to build two PUFs with the same challenge-response behavior. (see Figure 2.2)
- It is difficult to guess the response of a PUF for a given challenge.
- It has random outputs.

There are several PUF implementations in the literature. The studies about the usage of systems having different physical properties were started in the 1980s. The first study that uses the term PUFs is [18]. The most important ones are delay-based PUFs [19–21], memory-based PUFs [22–24], and coating PUFs [25]. The properties and the most basic usage of delay-based PUFs and memory-based PUFs were analyzed in [26,27]. Katzenbeisser *et al.* [27] stated that SRAM PUFs seem to achieve all desired properties of a PUF.

PUFs are promising functions that can be used in the design of secure and low-cost authentication protocols for RFID systems. PUFs can be used to solve the concerns of the RFID solution by acting as a unique identifier.

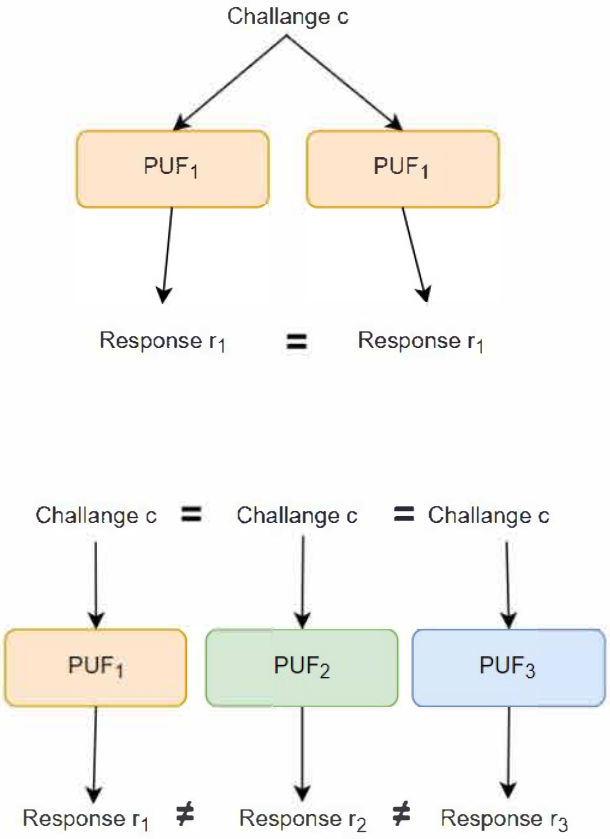

Figure 2.2. Challenge-response behaviour of different PUFs.

It provides to build more secure RFID solutions. Several studies have already been made to implement PUFs on RFID tags, just as done by Devadas *et al.* [28]. They designed and fabricated RFID ICs with the silicon PUF circuit based on MUXes and an arbiter. Furthermore, Devadas *et al.* founded Verayo Inc. that provides PUF-based security products for authentication of products and anti-counterfeiting and developed the first commercial PUF embedded RFID tag.

Definition 2.2.1 (Physically Unclonable Function (PUF) [29]). *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* $\{0,1\}^{\gamma} \rightarrow \{0,1\}^{\kappa}$ *that has the following parameters:*

(i) *For all* $c \in \{0,1\}^{\gamma}$ *and all pairs* $(r_i, r_j) \in [P(c)]^2$*, it holds that probability* $Pr[r_i = r_j] = 1$.

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

(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.*

2.3. Hash Function

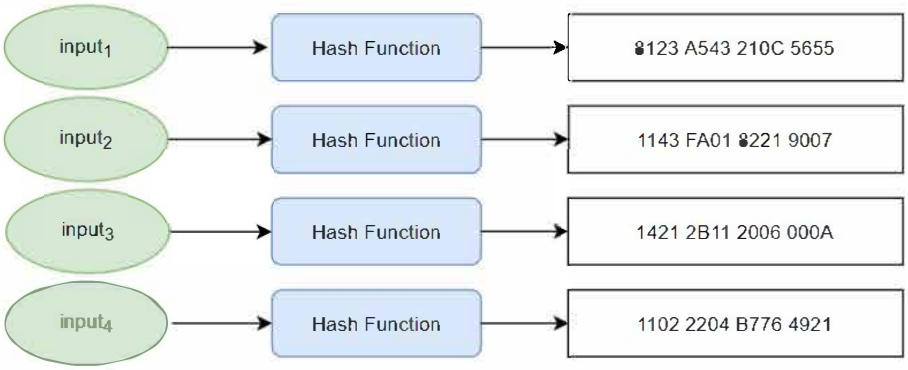

Figure 2.3. Hash Function Behaviour Example.

A Hash function is a one-way cryptographic algorithm that converts inputs that are keys for this function to numeric outputs. The main aim of the hash function which does not have one-to-one property is to create a map between input and output. The hash function acts as an algorithm to index the output. Therefore, the value returned by a hash function is not unique for each key, and this set of values is called a Hash Table. A hash table which has a fixed size generates the same output for the same input. It uses certain bit operations inside to generate output like OR, XOR, or ADD. Briefly, a hash function maps inputs with the hash table.

The hash functions take many forms that come up with their advantages and disadvantages. While many study areas use the hash function to benefit, there are also several studies concentrated to develop more efficient hash functions. A hash function reduces the amount of memory that should be stored since the hash function has a fixed size. A useful hash function can provide fast computation and reduce the number of duplication of output values. In that way, the hash function allows a fast search for the application that's why, hash functions are used in data storage and data retrieval applications. It is very commonly used in cryptography, RFID solutions, and authentication solutions.

It is possible to use hash functions in RFID tags since there are special cryptographic hash functions that are designed for RFID tags [30] [31]. Most of the RFID protocols that we mention in the upcoming chapters use the hash functions in different ways. For example, Akgün *et al.* [7] use hash function for the tag authentication, Alomair *et al.* [5] and Kardaş *et al.* [11] use successive hash functions to generate temporary value to keep secure their secrets.

Definition 2.3.1 (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 $\{0,1\}^{\gamma} \rightarrow \{0,1\}^{\kappa}$ with the following basic requirements:*

(i) For a given output y_i, it is computationally infeasible to find an 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.

2.4. Symmetric Key and Asymmetric Key Cryptography

2.4.1. Symmetric Key Cryptography

Symmetric-key cryptography, also called Private-Key Cryptography, is one of the old and simple encryption schemes. It uses only one secret key for both the encryption and decryption of messages. This key should be kept protected by users. Encrypted message format cannot be inspected without this secret key, thus the messages are prevented from any compromising attack.

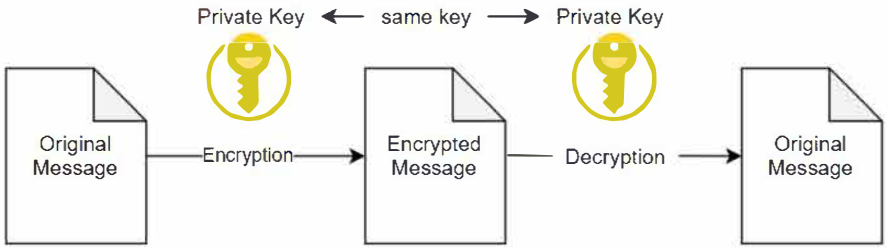

Figure 2.4. Symmetric Key Cryptography.

2.4.2. Asymmetric Key Cryptography

Asymmetric cryptography, also called Public-Key Cryptography (PKC), is an encryption scheme that uses two keys: public and private. The public key is used for the encryption of messages and it can be shared by users whereas, the private key is used for decryption and it should be withheld by its owner. Encrypted message format cannot be inspected without the private key, thus the messages are prevented from any compromising attack.

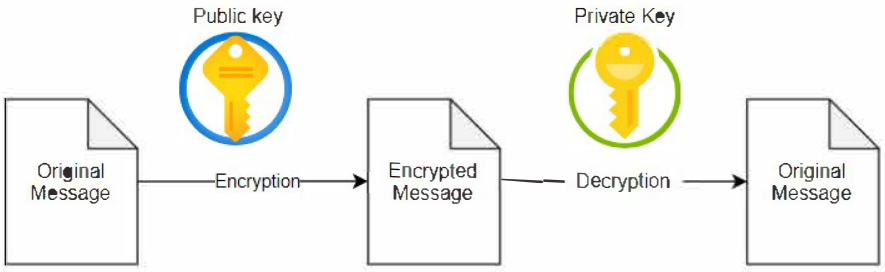

Figure 2.5. Asymmetric Key Cryptography.

2.4.3. The Differences Between Symmetric Key and Asymmetric Key Cryptography

Symmetric-key cryptography is much simpler than asymmetric-key cryptography. A massive amount of data can be transferred and take less time to encrypt and decrypt the messages with symmetric-key cryptography. One drawback of symmetric-key cryptography is that it has to be used in a closed system to achieve secure authentication because all users should be trusted to share a secret key.

On the other side, an encryption scheme can be more secure with asymmetric cryptography through its two keys feature. However, it has some disadvantages in terms of its complexity and time consumption. For instance, it is a more complex and time-consuming process. The length of the keys of asymmetric encryption is greater than symmetric encryption's, therefore asymmetric encryption can cause some CPU problems.

There are examples of both being used to provide secure authentication in RFID solutions. Avoine *et al.* [6] claim that public-key cryptography is expensive and time-consuming to perform transactions in RFID tags due to the insufficient computational capabilities of RFID tags.

Therefore, Avoine *et al*. analyzed the protocols that used symmetric-key cryptography, for example [32–34], in this paper. On the contrary, Hein *et al*. [35] claim that some cryptography methods can be preferred in the RFID tags that have an asymmetric approach. For example, they proposed a new processor based on elliptic curve cryptography for tag authentication in RFID. Hutter *et al*. [36] and Lee *et al*. [37] also studied on elliptic curve cryptography to present more effective solutions.

3. RELATED WORK

There are several proposed RFID systems in the literature to produce effective solutions. While some of them have scability problems to identify a tag, there are also certain solutions to decrease that identification time but at the same time, they have some security vulnerabilities to attacks such as replay attacks or impersonation attacks. To increase the security of solutions, PUFs are utilized to keep keys as secrets in some studies. This chapter aims to introduce those previously studied protocols together with their novel approaches problems and vulnerabilities, besides, in order to improve the protocol, some open problems will be specified in two sections in terms of the usage of PUFs.

3.1. Scalable RFID Solutions

Molnar and Wagner [1] proposed a privacy-preserving RFID protocol to solve privacy issues related to RFID systems in libraries. Their new protocol uses a balanced tree structure to store keys to reduce identification complexity. When d is the depth of the tree with branching factor α, each tag has $d+1$ keys and should store only that number of keys. For authentication, the reader needs $\alpha \cdot d$ keys in the tree. For each tag, consecutive calculations for the challenge-response protocol are done to get the main key of the tag. In that way, tag authentication is completed according to the acquired key in a consequence of confirmation or error. The identification complexity of this protocol is $\mathcal{O}(\log N)$ if there are N tags in the system.

In the example Figure 3.1, the branching factor is 4, the depth of the tree is 3 and the number of tags is 16. Considering this example, the tag T_5 has three keys: k_0, $k_{1,2}$, and $k_{1,2,1}$. For reader authentication, three consecutive calculations should be done for each tag.

Avoine *et al.* [2] presented an attack to show the weakness of the protocol. In this attack, an adversary can trace only one tag with a tampered tag.

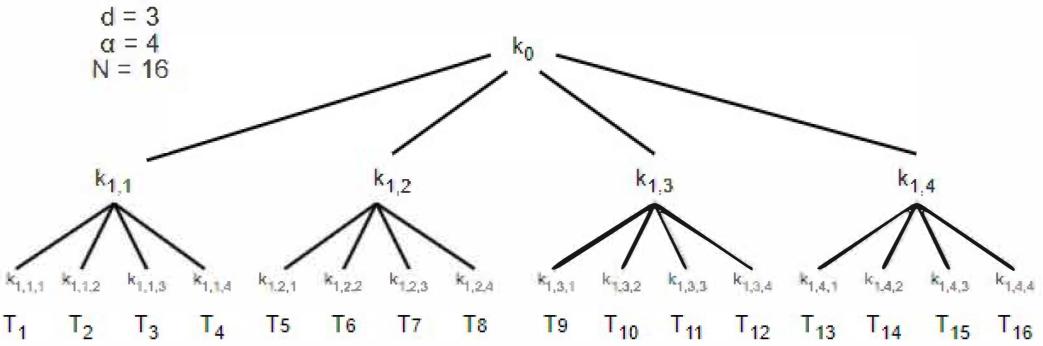

Figure 3.1. Balanced Tree Example for Molnar and Wagner's Protocol.

It is mean that a tag can be traceable in this protocol. Avoine shows that this protocol provides $\mathsf{NARROW} - \mathsf{WEAK}$ privacy.

Table 3.1. Okhubo-Suzuki-Kinoshita Original Protocol [2].

System		**Tag**
	request $\longrightarrow$	
	$\longleftarrow$ $G(s_i{}^k)$	$s_i{}^{k+1} = H(s_i{}^k)$

Avoine *et al.* [2] applied a variant of Hellman's time-memory trace-off technique in [38] in order to reduce the time complexity of the Okhubo-Suzuki-Kinoshita (OSK) protocol from [33] and they reduced it from $\mathcal{O}(N)$ to $\mathcal{O}(N^{2/3})$. They show that Okhubo-Suzuki-Kinoshita (OSK) protocol (see in Table 3.1) is vulnerable against replay attacks which they also made some modifications afterwards to avoid.

In this protocol (see modified version in Table 3.2), an adversary which tampers with a tag cannot trace its past interactions however, the adversary can impersonate the tag and can trace its future interactions. Avoine *et al.*'s protocol is $\mathsf{NARROW-FORWARD}$ private.

Table 3.2. Okhubo-Suzuki-Kinoshita Modified Protocol by Avonie *et al.* [2].

System		**Tag**
	$\xrightarrow{\text{r}}$	
	$\xleftarrow{G(s_i{}^k) \oplus r}$	$s_i{}^{k+1} = H(s_i{}^k)$

Avonie *et al.* [6] also proposed another protocol that has constant time identification by using only one master key. In this protocol, they tried but could not prevent DoS attacks. Due to this vulnerability to DoS attacks, they mention their open question of whether it is possible to design a protocol that has constant time identification without any loss of security, and they were successful at showing that many previous protocols have vulnerabilities to different kinds of attacks such as traceability.

Wu and Stinson [4] proposed an RFID protocol to solve the scalability problem, which provides security and privacy thanks to the difficulty of reconstructing a polynomial with noisy data. The time complexity of identifying a tag is to solve mb 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 [4]. A server performs 128 polynomial operations to identify a tag, therefore it has the same calculation burden as a tree-based system with 2^{128} tags [5]. This protocol has $\mathcal{O}(1)$ identification complexity. However, an adversary which is repeatedly querying a tag Q_{max} times can trace the tag since the maximum number of queries that the tag will answer correctly is limited to Q_{max}, which means this protocol does not provide privacy.

Alomair *et al.* [5] proposed an RFID protocol that addresses the private identification problem with constant time identification. They constructed a three-layered database structure. It works like a hashing mechanism but they truncate the inputs according to their most significant bits. After three truncations, they get the actual information about the tag. By means of this database structure, readers are able to perform both retrieving data and identifying a tag in an extremely short time. They have mutual authentication and they obtain $\mathcal{O}(1)$ identification complexity (see protocol in the Table 3.3).

Table 3.3. Alomair's Proposed Protocol [5].

Database	**Reader**	**Tag**
		k, c, p_i
	$r \in_R \{0,1\}^L$ →	
	← $h(p_i, c), r' = h(0, p_i, c, k, r)$	
← $h(p_i, c)$		
→ p_i, k, p_i'		
	→ $h(1, p_i, k, r') \oplus p_i', h(2, p_i', k, r')$	

Generally, Alomair's protocol has the following variables:

- Each tag has a secret pseudonym p_i that acts as a unique identifier. However, in the system, the number of pseudonyms should be more than the number of tags because the pseudonyms of the tags are updated after each authentication. The main reason to update the pseudonyms is to prevent tags from any compromising attacks.

- Each tag has a secret key k. Similar to the pseudonyms, the secret key is also updated after each authentication to provide privacy against tag compromising attacks.

- Each tag has a counter c. At first, the counter is 0, and after every reader query, the counter is increased gradually. If the authentication is successful, the counter value is reset to 0. The counter is used to mitigate traceability attacks.

- There is a maximum counter number as C.

- For each pseudonym, the hash value $h(p_i, c)$ is computed for all c (maximum C) and they are stored in the database for tag authentication.

However, Avonie *et al.* [6] showed that an adversary could identify a tag by querying C consecutive times as well as track its past interactions. Although increasing C number makes this attack harder, it still has a high probability to trace the tag, especially for rarely used systems like ticket systems.

3.2. PUF-Based Solutions

Damith *et al.* [39] proposed a PUF-based protocol in which precomputed challenge-response pairs are stored in the back end database. This protocol is vulnerable to replay attacks. Precomputed challenge-response pairs can be used to trace a specific tag, nonetheless, encryption of challenges and responses eliminates this problem.

Lenoid and Gabriel [40] proposed PUF based RFID identification protocol. In the proposed protocol, PUFs are used to update identifiers of tags. The reader stores k future identifiers of tags. This protocol is vulnerable to DoS attacks in case the tag is forced to update its identifier more than k.

Bringer *et al.* [3] use the tree-based protocol in [1] using Physical Obfuscated Keys (POKs) and increase the resistance of tags against any probability of corruption of the tags. Thus, an adversary which corrupts a tag cannot learn the secrets of the tag. This protocol provides $\mathsf{NARROW-DESTRUCTIVE}$ privacy with $\mathcal{O}(\log N)$ identification time.

Sadeghi *et al.* [29] utilized PUFs to develop the first destructive private RFID protocol. The proposed protocol is based on the weak private protocol in [41]. The identification complexity of this protocol is $\mathcal{O}(N)$. Kardaş *et al.* [9] showed that the secret key K is revealed by applying a cold boot attack [42]. Thus, this protocol is $\mathsf{NARROW-WEAK}$ private.

Mauv and Piramuthu [43] utilized PUFs in order to provide a solution for the ticket-switching problem in retail stores. This protocol accomplishes authentication of the tags thus it provides only one-way authentication.

Kardaş *et al.* [11] proposed an RFID mutual authentication protocol (in Table 3.4) which utilized PUFs to provide unique identities to the tags and to obtain resistance against side-channel attacks. Kardaş *et al.* claims this protocol provides $\mathsf{NARROW-DESTRUCTIVE}$ privacy with $\mathcal{O}(N)$ identification complexity against the attacks to both of a reader and a tag.

However, Tiplea and Hristea [8] have shown that this protocol does not provide the claimed level of privacy with their proposed attacks to the last interaction for the reader authentication used in Kardaş *et al.*'s protocol and in that way, an adversary can define a corrupted tag by tracing that interaction which means that this protocol is vulnerable to traceability attacks.

In other respects, Kardaş *et al.* [11] focused on the compromising attacks on the reader side in this protocol. They categorized RFID systems into two groups: online and ofline RFID systems which is also mentioned in Section 2.1.

Table 3.4. Kardaş *et al*. Protocol [11].

Reader		**Tag**
ID_R, c_R		ID_i, G_i, c_i
$DB = [(ID_1, {K_1}^1, {K_1}^2),$		
$\quad \ldots\ , (ID_i, {K_i}^1, {K_i}^2)]$		
Pick $n_R \in_R \{0,1\}^\alpha$		
	$\xrightarrow{ID_R, c_R, n_R}$	
		Pick $n_T \in_R \{0,1\}^\alpha$
		If $c_R \geq c_i$ then
		$\quad {S_i}^1 = P_i(G_i)$
		$\quad {K_i}^1 = H({S_i}^1, ID_R, c_R)$
		$\quad \text{temp} = H({K_i}^1, n_R, n_T)$
		$\quad$ delete ${S_i}^1, {K_i}^1$
		$\quad {S_i}^2 = P_i(G_i \oplus ID_i)$
		$\quad {K_i}^2 = H({S_i}^2, ID_R, c_R)$
		$\quad v_1, v_2 = H({K_i}^2, temp)$
		$\quad$ delete ${S_i}^2, {K_i}^2$
		else $v_1 \in_R \{0,1\}^\gamma$
	$\xleftarrow{n_T, v_1}$	
If $\exists(ID_i, {K_i}^1, {K_i}^2 \in DB)$		
$\quad$ s.t		
$\quad v_1', v_2' = H({K_i}^2, H({K_i}^1, n_R, n_T))$		
$\quad v_1' = v_1$		
then Send v_2'		
else Send $v_2' \in_R \{0,1\}^\gamma$		
	$\xrightarrow{v_2'}$	
		If $v_2' = v_2$ && $c_R > c_i$ then
		$\quad c_T = c_R$

However, they have designed their protocol only for offline RFID systems and have set two main conditions which are highly difficult to meet. In fact, it does not solve the vulnerability of reader side compromising, instead they only have a mechanism in which compromised offline readers are no longer a threat to the security of the entire system as long as certain conditions are met. Therefore, it can be stated that this protocol cannot provide reader side privacy strictly which is still an open problem to be solved in RFID systems.

Akgün *et al.* [7] proposed the first PUF-based RFID authentication protocol that provides $\mathsf{NARROW-DESTRUCTIVE}$ privacy with $\mathcal{O}(1)$ identification complexity (see the protocol in Table 3.5). They ensure the secure storage and use of the shared master key between tags and readers by utilizing PUFs and their design is secure against tag side compromising attacks. However, corruption of a single reader in the system causes master keys to be revealed and as a result all tags to be compromised. Although the result of reader corruption affects all tags in the system, no solution has been proposed to prevent from this occurence.

In addition to this vulnerability, Tiplea and Hristea [8] have shown that this protocol does not provide the claimed level of privacy. Akgün *et al.* has mutual authentication in their protocol and just as Kardaş *et al.* have done in their protocol, they added the last interaction for the reader authentication. However, in their paper, Tiplea and Hristea have stated that in the proposed attack, the last interaction of a corrupted tag can be defined by an adversary, yet, it neither can identify the tag's previous interactions nor impersonate the tag. Despite a tiny impact of this attack on the security of the solution, it is indicated that Akgün *et al.*'s protocol does not provide claimed privacy and this proposed attack is another open problem to be solved.

Hristea and Ţiplea [10] proposed a stateful (PUF based) scheme with constant identifiers (see in Table 3.6). The proposed protocol updates the tag identifier after tag identification. The key purpose of constant tag identifiers is to enable efficient tag identification. Because of constant tag identifiers, tag identification in the database can take only $\mathcal{O}(\log N)$ time.

Table 3.5. Akgün *et al*. Protocol [7].

Reader		**Tag**
$S, [ID_i, a_i, b_i, DATA_i]$		ID_i, a_i, b_i, c_i
$r_1 \in \{0.1\}^l$		
	$\xrightarrow{r_1}$	
		$r_2 \in \{0.1\}^l$
		$M_1 \leftarrow H(r_1, r_2, a_i)$
		$M_2 \leftarrow H(r_2, r_1, 1) \oplus ID_i$
		$h \leftarrow H(r_2, 1, 2)$
		$k \leftarrow P_i(a_i) \oplus r_2$
		delete $P_i(a_i)$ and r_2
		$k \leftarrow k \oplus P_i(b_i) \oplus c_i$
		delete $P_i(b_i)$
	$\xleftarrow{M_1, M_2, k}$	
$r_3 \in \{0.1\}^l$		
$r_2' \leftarrow S \oplus k$		
$ID_i' \leftarrow M_2 \oplus H(r_2', r_1, 1)$		
if $(M_1 = H(r_1, r_2', a_i))$		
$\quad M_3 \leftarrow H(H(r_2', 1, 2), r_3, b_i)$		
else		
$\quad \perp$		
	$\xrightarrow{r_3, M_3}$	
		if $(M_3 \neq H(h, r_3, b_i))$

The authors claimed that the stateful RFID schemes with constant tag identifiers do not provide any of kind privacy in Vaudenay's model. Therefore, they proposed new protocols that are WEAK private and DESTRUCTIVE private in randomized Vaudenay's model.

However, their protocol is still vulnerable to traceability and impersonation attacks. Although RFID protocols require random messages to provide security against traceability attacks, both or the readers and tags do not randomize its response in this protocol for each challenge of the reader. For example, as it can be seen in Table 3.6, until successful reader authentication step on the tag side, the tag always returns z for the first message. Besides, the reader always response same message w, which means that the tag is traceable by a passive adversary between the two successful identifications. This protocol allows adversaries to impersonate tags which is not possible to do with the same tag consecutively because there is a mechanism that restores synchronization between the tag and the reader. However, one-time tag impersonation is a major security vulnerability in RFID applications such as access control. To illustrate, an attacker could enter a room to which the attacker does not have an access right or an attacker could use someone else's train ticket. More importantly, an adversary that corrupts a tag can identify all past interactions of the tag. Updating tags' key in a reversible manner causes the adversary to identify tags' past interactions.

Therefore, it can be understood that randomized messages have crucial roles to provide privacy against traceability attacks. It is another issue we pay attention to while defining our protocol.

Table 3.6. Hristea and Ţiplea's Protocol [10].

Reader		**Tag**
DB, F		K, x
		$z = F_K(0, 0, x)$
	$\xleftarrow{z}$	
If $\exists(ID, K, x) \in DB$ and $i \in \{0, 1\}$ s.t $z = F_K(0, 0, x + i)$ then $x = x + i$ and $w = F_K(0, 1, x + i)$ else $w \leftarrow \{0, 1\}^{l_2}$		
	$\xrightarrow{w}$	
		$w' = F_K(0, 1, x)$ If $w = w'$ then output OK $x = x + 1, w' = F_K(1, 2, x)$ else else output $\perp$ and $w' = F_K(1, 0, x)$
	$\xleftarrow{w'}$	
If $w' = F_K(1, 2, x + 1)$ the output $ID, x = x + 1$ else output $\perp$		

4. SECURITY AND PRIVACY MODEL

In the literature, there are many privacy models presented to define the privacy of the RFID solutions. Avoine *et al.*'s [44] model formalizes privacy in terms of distinguishing two tags. Afterward, Juels *et al.* [45] extended. Another example is Damgård's study [46] focusing on RFID systems with symmetric-key cryptography. However, comparison of protocols by using these models could be hard because they present their privacy classes according to their protocols or solutions. Establishing a common model was a concern in order to present a common measure for the performance of the RFID systems. For this purpose, Vaudenay [41] presented a new privacy model that can be used for all protocols, it includes privacy and security definitions, oracle definitions for adversaries, and analyzes.

In this chapter, we describe the RFID security and privacy model presented by Vaudenay [41] and our extensions. In our model, there are a back end database, a set of readers R_j, and a set of tags T_i. The connection between readers and back end database is secure. The main aim of using readers is to identify tags and specify unknown tags. Tags have unique identifiers. Each tag T_i and reader R_j have computational abilities like PUF evaluation, hashing, and random number generation. The back end database stores tag identifiers and their information. K_P is a public key in this scheme and a tag can be set up with this key.

4.1. System Model

The first step to describe an RFID scheme is to set up readers and back end database. Then, tags with unique IDs are set up. In the end, the protocol is run between a reader and a tag for the identification step, as a result, it returns an output that is correct except with a negligible probability. Following procedures can formalize the definition of an RFID scheme:

- SetupReader(1^s): generates the back end database. Besides, it generates a private/public key pair (K_S, K_P) with the given security parameter s. The K_P represents a public key which is publicly released whereas K_S represents a private key which the back end database stores.

- $\mathsf{SetupTag}_{\mathsf{K_P}}(\mathsf{ID})$: generates tag instances by using K_P. They have unique identifiers ID. This procedure also generates specific secret K_S and the initial state of the tag.

- IdentTag: is a procedure between tags and readers. If the reader identifies the tag, then the output is the tag identifier. Otherwise, the output is $\perp$.

4.2. Adversary Model

In this section, we describe oracles presented by Vaudenay [41] which are executed by adversaries only for getting access to the current state of the tag and the reader. Besides, we define our additional oracle which is executed by the adversary to corrupt a reader. For each oracle, a tag can be free or drawn. Drawn tags, called 'virtual tag', are the ones which an adversary contacts. A virtual tag has a temporary identifier as vtag.

- CreateTag(ID): uses $SetupTag_{K_P}$ algorithm to create a tag with a unique identifier only for legitimate tags and it updates the database to store information of this new tag.

- DrawTag(distr): chooses a set of free tags as drawn tags at random with distribution probability $distr$. It assigns new identifiers to the selected tags. Then, this oracle returns this set of tag identifiers and their bits telling whether they are legitimate or not $(vtag_1, b_1, vtag_2, b_2, ..., vtag_n, b_n)$.

- Free(vtag): changes the status of the drawn tag with the identifier ($vtag$) to free. It makes $vtag$ unreachable from oracles.

- Launch: enables the reader to start a new $IdentTag$ protocol instance π.

- SendReader(m, π): sends a message m to a protocol instance π for the reader. Then, it receives the message m' as an answer.

- SendTag(m, T_i): sends a message m to a tag T_i. Then, it receives the message m' as an answer.

- Result(π): returns 1 in case the reader identifies a legitimate tag, and 0 otherwise at the end of the protocol session.

- Corrupt(T_i): corrupts tag T_i and gets the internal states of that tag.

Because security issues of the reader side are ignored in Vaudenay's model and the readers can also be corrupted by adversaries like tags in our protocol, we added a new oracle for readers to the privacy model. Our new oracle is:

- CorruptReader(R_j): corrupts reader R_j and gets internal states of it.

4.3. Security

Definition 4.3.1 (Security [41]). *A scheme provides security if it provides secure tag authentication.*

- *Tag authentication is secure if there exists a polynomial-time adversary such that at least one protocol session π on the reader identified an uncorrupted legitimate tag ID but π and ID do not have any matching conversation, with non-negligible probability.*

4.4. Privacy

Vaudenay [41] defines the classes of adversaries. We extend it by considering reader side corruption.

Definition 4.4.1 (Adversary Classes [41])**.** *Weak, Forward, Destructive, Strong and Narrow adversary:*

(i) WEAK*: Adversaries cannot execute CorruptReader and CorruptTag oracles which means that corruption is not allowed.*

(ii) FORWARD*: Adversary can execute CorruptReader and CorruptTag oracles after other executed CorruptReader or CorruptTag oracles.*

(iii) DESTRUCTIVE*: By adversaries, executing CorruptReader and CorruptTag oracles destroys the reader or tag.*

(iv) STRONG*: Adversaries can execute all oracles which means that corruption is allowed.*

(v) NARROW*: Adversaries cannot learn whether the reader identifies a tag or not.*

When we want to sort the classes in terms of corruption, we consider the first four classes and get this order:

$$WEAK \subseteq FORWARD \subseteq DESTRUCTIVE \subseteq STRONG. \tag{4.1}$$

However, the class NARROW represents a highly different property of privacy, which is evaluated according to whether adversaries can learn the reader identifies a tag or not. A protocol can have both NARROW and one of the first four privacy classes. For example, a protocol that does not allow an adversary to learn the result of identification and corruption has both of NARROW and WEAK privacy level.

Definition 4.4.2 (Privacy)**.** *RFID scheme is P-private if all such adversaries which belong to class P cannot distinguish a real RFID system from a simulated RFID system with non-negligible probability.*

5. SCALABLE RFID AUTHENTICATION PROTOCOL

This chapter introduces a new protocol that addresses three open problems that are stated in previous chapters: a private protocol which runs with a constant identification time; the proposed attacks presented by Hristea and Ţiplea [8]; and reader side compromising attacks. For the first open problem, Avonie *et al.* [6] questions whether a private protocol may run with a constant identification time or not, actually, many protocols have constant identification time yet, they have distinct vulnerabilities against attacks such as impersonation attack, traceability or side-channel attack. Another open problem that is indicated by Hristea and Ţiplea put forward that many protocols have a vulnerability for the proposed attacks. As for the last but not least open problem, there is not any strict privacy protocol against those attacks except for one protocol [11] which partially considers reader side comrpomising attacks and applies a certain mechanism. The new protocol addresses to these three problems without losing any other features.

Before the detailed explanation, we can briefly mention our protocol as follows: In our proposed protocol, two master keys shared by readers and tags are used and they speed up the authentication steps. In order to provide security to master keys against compromising attacks, PUFs are utilized in both tag and reader sides. We also use hash functions to perform tag authentication on the reader side. Additionally, we randomize the messages between the tag and reader due to the widely-known fact that RFID protocols require random messages to provide security against traceability attacks. Our random variables provide these randomized messages for protocol, and by means of master keys and random variables, we get constant identification time and desired level of privacy for our protocol.

In section 5.1 there is a table that shows the notations used in the proposed protocol below. Section 5.2 describes our proposed protocol and its initialization and authentication steps in details and finally, in the following part 5.3, the privacy and security of the protocol are analyzed extensively.

5.1. Notations

Table 5.1 gives the notations used in describing the proposed protocol.

Table 5.1. Notations.

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
$(a, b, c, d, e, f)_j$	Secret values of a reader R_j
$(u, v, w, x, y, z)_i$	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
P_j	The PUF $\{0,1\}^l \rightarrow \{0,1\}^l$ of a reader R_j
$\oplus$	XOR operator
r_1, r_2, r_3	Random numbers
$\in$	Random choice operator

5.2. Protocol Description

We describe here the proposed authentication protocol in great detail. The proposed protocol has two phases: initialization and authentication.

5.2.1. Initialization Phase

The back end server generates two random keys S_1 and S_2 which are common keys shared by all tags and readers.

Four random unique keys a, b, d, e generated for each reader. Then, keys $c = S_1 \oplus P(a) \oplus P(b)$ and $f = S_2 \oplus P(d) \oplus P(e)$ are computed for each reader. Each reader uses its own embedded PUF $P(.)$ for the calculation of c and f.

Four random unique keys u, v, x, y generated for each tag. Then, keys $w = S_1 \oplus P(u) \oplus P(v)$ and $z = S_2 \oplus P(x) \oplus P(y)$ are computed for each tag. Each tag uses its own embedded PUF $P(.)$ for the calculation of w and z. The back end server stores $[ID, DATA]$ for each tag.

5.2.2. Authentication Phase

(i) The reader creates two nonces $r_1, r_2 \in \{0,1\}^l$ and calculates $M_1 \leftarrow P_j(a_j) \oplus P_j(r_1)$. After this calculation, $P_j(a_j)$ and $P_j(r_1)$ are deleted from the volatile memory. The reader continues to calculate $M_1 \leftarrow P_j(b_j) \oplus P_j(r_2) \oplus M_1$. Then $P_j(b_j)$ and $P_j(r_2)$ are deleted from the volatile memory. Finally, the reader calculates $M_1 \leftarrow c_j \oplus M_1$ and sends M_1 to tags.

(ii) Upon receiving M_1, a tag T_i creates a nonce $r_3 \in \{0,1\}^l$ and calculates $M_4 = r_3 \oplus P_i(x_i)$ and $k_1 = r_3 \oplus P_i(u_i)$. After these calculations, r_3, $P_i(x_i)$ and $P_i(u_i)$ are deleted from the volatile memory. It XORes k_1 with $P_i(v_i)$. $P_i(v_i)$ is deleted from the volatile memory. The tag calculates $k_1 \leftarrow k_1 \oplus w$, $k_2 \leftarrow H(k_1, 1, 2)$, $M_2 \leftarrow H(k_1, ID_i, M_1)$ and $M_3 \leftarrow k_2 \oplus ID_i$. Then, the tag continues to calculate M_4 as follows: It XORes M_4 with $P_i(y_i)$. $P_i(y_i)$ is deleted from the volatile memory. Finally, it computes M_4 by XORing with $z_i \oplus M_1$ and sends M_2, M_3 and M_4 to the reader.

(iii) The reader calculates $M_4 \leftarrow P_j(d_j) \oplus P_j(r_1) \oplus M_4$. After this calculation, the reader deletes $P_j(d_j)$ and $P_j(r_1)$. The reader continues to calculate $M_4 \leftarrow P_j(e_j) \oplus P_j(r_2) \oplus M_4$. Then $P_j(e_j)$ and $P_j(r_2)$ are deleted from the volatile memory. Then, it obtains $k_1 \leftarrow f_j \oplus M_4$ and $ID_i \leftarrow M_3 \oplus H(k_1, 1, 2)$. It checks the validity of M_2. If M_2 is not valid, the reader stops the session.

Table 5.2. The Proposed Protocol.

Reader R_j		**Tag** T_i
$a_j, b_j, c_j, d_j, e_j, f_j, [ID_i, DATA_i]$		$ID_i, u_i, v_i, w_i, x_i, y_i, z_i$
$r_1, r_2 \in \{0,1\}^l$		
$M_1 \leftarrow P_j(a_j) \oplus P_j(r_1)$		
delete $P_j(a_j)$ and $P_j(r_1)$		
$M_1 \leftarrow P_j(b_j) \oplus P_j(r_2) \oplus M_1$		
delete $P_j(b_j)$ and $P_j(r_2)$		
$M_1 \leftarrow c_j \oplus M_1$		
	$\xrightarrow{M_1}$	
		$r_3 \in \{0,1\}^l$
		$M_4 \leftarrow r_3 \oplus \bar{P}_i(x_i)$
		$k_1 \leftarrow r_3 \oplus \bar{P}_i(u_i)$
		delete r_3, $P_i(x_i)$ and $P_i(u_i)$
		$k_1 \leftarrow k_1 \oplus P_i(v_i)$
		delete $P_i(v_i)$
		$k_1 \leftarrow k_1 \oplus w$
		$k_2 \leftarrow H(k_1, 1, 2)$
		$M_2 \leftarrow H(k_1, ID_i, M_1)$
		$M_3 \leftarrow k_2 \oplus ID_i$
		$M_4 \leftarrow M_4 \oplus \bar{P}_i(y_i)$
		delete $P_i(y_i)$
		$M_4 \leftarrow M_4 \oplus M_1 \oplus z_i$
	$\xleftarrow{M_2, M_3, M_4}$	
$M_4 \leftarrow P_j(d_j) \oplus P_j(r_1) \oplus M_4$		
delete $P_j(d_j)$ and $P_j(r_1)$		
$M_4 \leftarrow P_j(e_j) \oplus P_j(r_2) \oplus M_4$		
delete $P_j(e_j)$ and $P_j(r_2)$		
$k_1 \leftarrow f_j \oplus M_4$		
$ID_i \leftarrow M_3 \oplus H(k_1, 1, 2)$		
if $(M_2 \neq H(k_1, ID_i, M_1))$ then $\perp$		

This protocol is designed for online RFID systems. It means that the readers are connected to the database at the back end. However, our protocol can be easily integrated into offline RFID systems. The only difference is that the readers do not connect to the back end continuously. Therefore, the readers have their own database. Without changing the protocol, we can provide the same privacy for offline RFID systems.

5.3. Analysis

In this section, we analyze the security and privacy level of our proposed protocol with lemmas and theorems.

Lemma 5.1. *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. We assume that there is an adversary $\mathcal{A}$ that can learn the secrets S_1 and S_2 by corrupting a tag. If $\mathcal{A}$ corrupt a tag T_i that is offline, $\mathcal{A}$ does not learn anything because the volatile memory is empty and $\mathcal{A}$ has to simulate $P_i()$ to compute S_1 and S_2. $\mathcal{A}$ can corrupt the tag while interacting with the reader. In this case, what the attacker can obtain changes depending on the time of the corruption because three deletions are carried out during the protocol execution. We consider each of the deletions one by one to determine $\mathcal{A}$'s advantage. We assume that $\mathcal{A}$ eavesdrops messages between R_j and T_i. This means $\mathcal{A}$ knows M_1 during the corruption. Let assume $\mathcal{A}$ corrupts the tag T_i before the first deletion. $\mathcal{A}$ obtains M_1, u_i, v_i, w_i, x_i, y_i, z_i, r_3, $P_i(x_i)$ and $P_i(u_i)$. In order to obtain the master key S_1 from M_1, $\mathcal{A}$ has to know $P_j(r_1)$ and $P_j(r_2)$. $P_j(r_1)$ and $P_j(r_2)$ are created by a reader R_j from random values r_1 and r_2. It is impossible for $\mathcal{A}$ to obtain these values by corrupting T_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 M_1, u_i, v_i, w_i, x_i, y_i, z_i, $r_3 \oplus P_i(x_i)$, and $P_i(v_i)$. In order to obtain the master key S_1, $\mathcal{A}$ has to simulate $P_i(.)$ or has to know $P_j(r_1)$ and $P_j(r_2)$. Let assume $\mathcal{A}$ corrupts the tag T_i before the third deletion.

$\mathcal{A}$ obtains M_1, M_2, M_3, u_i, v_i, w_i, x_i, y_i, z_i, $\bar{P}_i(y_i)$, $r_3 \oplus \bar{P}_i(x_i)$ and $r_3 \oplus S_1$. In order to obtain the master key S_2, $\mathcal{A}$ has to simulate $P_i(.)$ or has to obtain $P_i(x_i)$ from $r_3 \oplus P_i(x_i)$. $\mathcal{A}$ can obtain one secret value created with $P_i(.)$ at a time. Thus, $\mathcal{A}$ has to simulate $P_i(.)$. This contradicts with security definition of the PUF (Definition 2.2.1). As a result, $\mathcal{A}$ can learn S_1 and S_2 by corrupting the tag T_i with negligible probability. □

Lemma 5.2. *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 reader is negligible.*

Proof. We assume that there is an adversary $\mathcal{A}$ that can learn the secrets S_1 and S_2 by corrupting a reader. $\mathcal{A}$ cannot learn S_1 and S_2 by corrupting a reader R_j that is not interacting with any tag. The attacker has to simulate $P_j()$ to compute S_1 and S_2. $\mathcal{A}$ can corrupt R_j while interacting with a tag T_i. In this case, what the attacker can obtain changes depending on the time of the corruption because four deletions are carried out during the protocol execution. We consider each of deletions one by one to determine $\mathcal{A}$'s advantage. Let assume $\mathcal{A}$ corrupts R_j before the first deletion. $\mathcal{A}$ obtains r_1, r_2, a_j, b_j, c_j, d_j, e_j, f_j, $P_j(r_1)$ and $P_j(a_j)$. This shows that $\mathcal{A}$ has to simulate $P_j(.)$ In order to obtain the master keys S_1 and S_2. Let assume $\mathcal{A}$ corrupts the tag R_j before the second deletion. $\mathcal{A}$ obtains r_1, r_2, a_j, b_j, c_j, d_j, e_j, f_j, $\bar{P}_j(r_2)$, $\bar{P}_j(b_j)$ and $P_j(r_1) \oplus P_j(a_j)$. In order to obtain the master key S_1, $\mathcal{A}$ has to simulate $P_j(.)$ or has to obtain $P_j(a_j)$ from $P_j(r_1) \oplus P_j(a_j)$. In order to obtain the master key S_2, $\mathcal{A}$ has to simulate $P_j(.)$. Let assume $\mathcal{A}$ corrupts a reader R_j before the third deletion. $\mathcal{A}$ obtains r_1, r_2, a_j, b_j, c_j, d_j, e_j, f_j, M_1, M_2, M_3, M_4, $P_j(d_j)$ and $P_j(r_1)$. $\mathcal{A}$ has to simulate $P_j(.)$ in order to obtain the master keys S_1 and S_2 using these values. Let assume $\mathcal{A}$ corrupts the tag R_j before the fourth deletion. $\mathcal{A}$ obtains r_1, r_2, a_j, b_j, c_j, d_j, e_j, f_j, M_1, M_2, M_3, M_4, $P_j(e_j)$ and $P_j(r_2)$. $\mathcal{A}$ can compute the $S_1 \oplus S_2$ in this step. However, $\mathcal{A}$ cannot use this information to get master keys. In order to obtain the master key S_1 from M_1, $\mathcal{A}$ has to simulate $P_j(.)$. In order to obtain the master key S_2 from M_4, $\mathcal{A}$ also has to simulate $P_j(.)$. $\mathcal{A}$ can obtain one secret value created with $P_j(.)$ at a time. Thus, $\mathcal{A}$ has to simulate $P_j(.)$.

This contradicts with security definition of the PUF (Definition 2.2.1). As a result, $\mathcal{A}$ can learn S_1 or S_2 by corrupting the reader R_j with negligible probability. □

Theorem 5.3.1. *The proposed protocol provides tag authentication if H is a hash function (Definition 2.3.1) and P is a PUF (Definition 2.2.1).*

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

$$\begin{aligned} M_2 &= H(k_1, ID_i, M_1) \\ M_3 &= H(k_1, 1, 2) \oplus ID_i \\ M_4 &= M_1 \oplus r_3 \oplus P_i(x_i) \oplus P_i(y_i) \oplus z_i. \end{aligned} \tag{5.1}$$

We know that $\mathcal{A}$ cannot learn master secrets S_1 and S_2 by corrupting tags (Lemma 5.1). $\mathcal{A}$ can corrupt the tag T_i and learn ID_i, u_i, v_i, w_i, x_i, y_i, z_i. The adversary 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 $r_3^{s+1} = r_3^s$. $\mathcal{A}$ can easily generate $M_4^{s+1} = M_4^s \oplus M_1^{s+1} \oplus M_1^s$. In order to generate M_2^{s+1}, $\mathcal{A}$ needs to know r_3^s. M_2 is generated by computing $H(k_1, ID_i, M_1)$ where M_1 comes from the reader and k_1 is not known by the adversary. We know that $H(.)$ is a random-like function (Definition 2.3.1). 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 5.3.2. *The proposed protocol achieves narrow-destructive privacy if the protocol achieves tag authentication, P is PUF (Definition 2.2.1) and H is hash function (Definition 2.3.1).*

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. To prove Theorem 5.3.2, we create a blinder $\mathcal{B}$ and then show that any destructive adversary $\mathcal{A}$ can distinguish the blinder $\mathcal{B}$ from the real environment with at most negligible probability. We first show how $\mathcal{B}$ simulates oracles.

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

Let assume that there is a game $\mathcal{G}_0$ where $\mathcal{A}$ interacts with real oracles. We construct a new game $\mathcal{G}_1$ from $\mathcal{G}_0$. In the game $\mathcal{G}_1$, the states of all tags are simulated with randomly chosen values. For example, uniformly random values are assigned to following values $P_i(u_i) \in \{0,1\}^l$, $P_i(v_i) \in \{0,1\}^l$, $w_i \in \{0,1\}^l$, $P_i(x_i) \in \{0,1\}^l$, $P_i(y_i) \in \{0,1\}^l$, $z_i \in \{0,1\}^l$, and $ID_i \in \{0,1\}^l$ of a tag T_i. The same randomization is done for the readers. Their state and keys are simulated with randomly chosen values. During the attack, the challenger $\mathcal{C}$ responds to $\mathsf{SendReader}(\pi)$ oracle query with either $M_1 = S_1 \oplus P_j(r_1) \oplus P_j(r_2)$ as in the game $\mathcal{G}_0$ or $M_1 \in \{0,1\}^l$ as in the game $\mathcal{G}_1$. The challenger $\mathcal{C}$ responds to $\mathsf{SendTag}(\mathsf{M}_1, \mathsf{vtag})$ oracle query with $M_2 = H(S_1 \oplus r_3, ID_i, M_1)$, $M_3 = H(S_1 \oplus r_3, 1, 2)$, and $M_4 = S_2 \oplus r_3 \oplus M_1$ values generated as in the game $\mathcal{G}_0$ or $M_2 = H(S_1' \oplus r_3, ID_i', M_1)$, $M_3 = H(S_1' \oplus r_3, 1, 2)$, and $M_4 = S_2' \oplus r_3 \oplus M_1$ values generated as in the game $\mathcal{G}_1$ where $S_1' \in \{0,1\}^l$, $S_2' \in \{0,1\}^l$, and $ID_i' \in \{0,1\}^l$. $\mathcal{A}$ tries to distinguish $\mathcal{G}_1$ from $\mathcal{G}_0$. $\mathcal{A}$ can corrupt tags and readers. However, $\mathcal{A}$ cannot obtain any secret (Lemma 5.1) and corrupted tags and readers cannot be used any more (Definition 2.2.1).

After a polynomial times of queries, $\mathcal{A}$ can distinguish $\mathcal{G}_1$ from $\mathcal{G}_0$. 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 2.2.1). As a result, the success probability of $\mathcal{A}$ is negligible.

We construct a new game $\mathcal{G}_2$ from $\mathcal{G}_1$. Differently, in $\mathcal{G}_2$, SendReader and SendTag oracles are simulated by the blinder $\mathcal{B}$ as given above. $\mathcal{A}$ tries to distinguish $\mathcal{G}_2$ from $\mathcal{G}_1$. After a polynomial times of queries, $\mathcal{A}$ can distinguish $\mathcal{G}_2$ from $\mathcal{G}_1$. 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 2.3.1). As a result, the success probability of $\mathcal{A}$ is negligible.

We construct a new system $\mathcal{G}_3$ from $\mathcal{G}_2$. The only difference of $\mathcal{G}_3$ from $\mathcal{G}_2$ is that Result oracle is simulated by the blinder $\mathcal{B}$ as described above. $\mathcal{A}$ tries to distinguish $\mathcal{G}_3$ from $\mathcal{G}_2$. After a polynomial times of queries, $\mathcal{A}$ can distinguish $\mathcal{G}_3$ from $\mathcal{G}_2$. That means $\mathcal{A}$ runs a protocol instance π and $\mathcal{G}_3$ returns a different output than $\mathcal{G}_2$. We know that the simulation in $\mathcal{G}_3$ is perfect and this can only happen when $\mathcal{A}$ generates a protocol transcript that makes the system $\mathcal{G}_2$ returns 1. Theorem 5.3.1 says that this can happen with negligible probability. As a result the success probability of $\mathcal{A}$ is negligible.

Full proof shows that the game $\mathcal{G}_3$ equals to the set of all oracles simulated by the blinder $\mathcal{B}$ described above. The game $\mathcal{G}_0$ equals to the game where $\mathcal{A}$ interacts with real oracles. This means $\mathcal{A}$ cannot distinguish the full blinder $\mathcal{B}$ from the real system with non-negligible probability.

□

6. COMPARISON

We choose only some proposed protocols that have good results from related works in Chapter 3 for the comparison part. Then we compare our protocol with these protocols in terms of search time in the database, communication cost, computational cost in the tag side, privacy on the reader side, mutual authentication, and privacy level in Table 6.1. The main purpose of this comparison is to show what are open problems in literature, missing parts, and what points we emphasize and improve.

- Search time: The time it takes to find a tag in the database for tag identification. It indicates the speed of the algorithm. As shown in Table 6.1, the search time of Wu and Stinson [4] and Alomair *et al.* [5] are $\mathcal{O}(1)$. They are as fast as our proposal. However, when we look at their privacy levels, both of them are vulnerable to traceability attacks by querying a tag repeatedly. Therefore, both of them have no privacy. The search time of Akgün *et al.* [7] is also $\mathcal{O}(1)$ but it is not secure against reader side compromising attacks. The only protocol that has high-level privacy and less search time is our protocol.

- Communication Cost: It describes the interaction number between reader and tag. Especially, protocols that have mutual authentication can have more than 2 interactions. Since we do not see any need for mutual authentication between the tag and reader in our protocol, 2 communication steps are enough for us. Avoine *et al.* [2] has only two communication steps as our protocol. The others have more than two communication steps. It means that our protocol and Avoine *et al.*'s keep the communication cost low.

- Mutual Authentication: Some protocols require mutual authentication because they need to update their states and secret keys after the authentication step or maintain synchronization between readers, tags, and the back end server. For example, Akgün *et al.*'s protocol [7] has mutual authentication.

However, Tiplea and Hristea [8] have shown that the last communication of the tag and reader can be defined by an adversary and the tag can be traced. Therefore, Akgün does not have claimed privacy level. Similarly, Kardaş *et al.*'s protocol [11] is also vulnerable to traceability attacks because of its last interaction for reader authentication. Our protocol does not need mutual authentication because the scheme has static variables and does not have any steps to maintain synchronization. It decreases communication costs. Besides, it protects our protocol from the attacks that are mentioned in [8].

- Computational Cost: The computational cost of a tag in the system to do calculations and keep the system synchronized. Only Bringer *et al.*'s [3] and our protocols have $\mathsf{NARROW-DESTRUCTIVE}$ privacy. However, when we look at their computational costs, Bringer *et al.*'s protocol has a huge number of calculations, depending on the tag number N. On the other side, Avoine *et al.*'s protocol [2] has $\mathsf{NARROW-FORWARD}$ privacy but at the same time, its search time is very high. The other protocols do not have any privacy. On the contrary, we tried to keep computational cost low while offering high-level privacy with low search time.

- Reader Side Privacy: It means that an adversary can get important information like secret keys by compromising attack on the reader side. For example, Akgün *et al.* [7] is vulnerable to reader compromising attacks and an adversary can corrupt and compromise readers. Kardaş *et al.* [11] claims that their protocol has privacy to compromising attack on the reader for only offline RFID systems. However, this protocol has some conditions to provide this privacy. Let's assume an adversary corrupts a reader and gets current state information. If the system does not update the remaining readers and the tags do not have an interaction with one of the updated readers then the adversary can impersonate this reader. It means that Kardaş *et al.*'s protocol [11] has partial reader side privacy. However, our protocol utilizes PUFs in both the reader side and tag-side in order to provide security to master keys that are shared by all tags and readers.

In that way, it provides NARROW − DESTRUCTIVE privacy for both offline and online RFID systems without any conditions that must be met. On the other hand, the other solutions do not even consider compromising reader side attacks.

- Privacy: the privacy level that is mentioned (Definition 4.4.1). Our proposal has NARROW − DESTRUCTIVE privacy that is the best destructive privacy level in Table 6.1. Bringer *et al.* [3] and Avoine *et al.* [2] have also high privacy level. However, as we mentioned, our protocol has comprehensive privacy on the reader side and tag side unlike them.

At the end of the comparison of proposals in terms of these factors, our proposal gives the highest privacy level with the lower communication cost and search time on the reader side.

Table 6.1. Comparison of Protocols.

	Search Time	Communication Cost	Computational Cost*	Reader Side Privacy	Mutual Authentication	Privacy Level
Bringer *et al.* [3]	$\mathcal{O}(\log N)$	log2N + 2	(log2N+1) nonces + (log2N+1) hashes + 2(log2N+1) PUFs	no	no	NARROW-DESTRUCTIVE
Avoine *et al.* [2]	$\mathcal{O}(N^{2/3})$	3	3 hashes	no	yes	NARROW-FORWARD
Wu and Stinson [4]	$\mathcal{O}(1)$	2	(2b-1) nonces + 1 hash + 1 polynomial †	no	no	NO − PRIVACY
Kardaş *et al.* [11]	$\mathcal{O}(N)$	3	1 nonce + 4 hashes + 2 PUFs	yes §	yes	NO − PRIVACY
Alomair *et al.* [5]	$\mathcal{O}(1)$	3	5 hashes	no	yes	NO − PRIVACY
Hristea and Ţiplea [10]	$\mathcal{O}(\log N)$	3	3 PRFs + 2 PUFs	no	yes	NO − PRIVACY
Akgün *et al.* [7]	$\mathcal{O}(1)$	3	1 nonce + 4 hashes + 2 PUFs	no	yes	NO − PRIVACY
Proposed	$\mathcal{O}(1)$	2	1 nonce + 2 hashes + 4 PUFs	yes	no	NARROW-DESTRUCTIVE

* Computational cost of a tag in the system with N tags

† b is a predefined parameter

§ Partial reader side Privacy

7. CONCLUSION

RFID technology offers effective solutions, therefore, it is being widely used for many purposes such as personal tracking in business life, product or cargo tracking, and also, to give examples from our daily life experiences, using a credit card to shop and paying the toll can be counted for RFID's usage. This increase also comes up with some problems in the fields of security and privacy because tags and readers that are parts of RFID systems are quite vulnerable to attacks. Similarly, studies in this domain are also increasing and new suggestions are constantly being offered however, there is still no scalable solution that has high-level privacy. Therefore, we study to propose a scalable and privacy-preserving RFID protocol in this thesis.

In Chapter 3, we analyze the existing RFID protocols and specify the open problems that cause scalability and privacy concerns: a private protocol that runs with a constant identification time; the proposed attacks presented by Hristea and Ţiplea [8]; and reader side compromising attacks.

In Chapter 4, Vaudenay's security and privacy model [41] is described, which does not include reader side privacy and security. We introduce one additional oracle for the corruption of the reader side because our protocol addresses reader side security issues and provides privacy for the reader side attacks. Apart from this new oracle, we extend Vaudenay's adversary classes by considering also reader side corruption in order to evaluate our protocol privacy and security level with regard to the other protocols.

In Chapter 5, we introduce a scalable and privacy-preserving RFID protocol that addresses three issues that is already defined in details in Chapter 3. For the first open problem, we address the Avonie *et al.*'s question which is about whether a private protocol may run with a constant identification time or not and we propose a protocol that provides $\mathsf{NARROW-DESTRUCTIVE}$ privacy with $\mathcal{O}(1)$ identification complexity.

For the second issue, our protocol has only tag authentication but not an extra communication step for the reader authentication, in that way, the proposed attack presented by Hristea and Ţiplea [8] that cause a vulnerability is not valid for our protocol. For the last issue, there is a proposed protocol that has partially reader side privacy, which has some conditions to have this privacy mechanism and no other protocols in the literature solve this problem. We propose a protocol which utilizes Physically Unclonable Functions (PUFs) on both the reader side and tag side in order to provide security to master keys that are shared by all tags and readers. Therefore, it provides security against reader side compromise attacks only by using PUF without any conditions that must be met for all RFID systems. We present our protocol for online RFID systems, however, it can work regardless of the type of the RFID system. By transferring database records to readers, offline RFID systems can provide the same level of privacy.

In Chapter 6, we compare our protocol with the other existing protocols in terms of some important criteria: search time in the database, communication cost, computational cost in the tag side, privacy on the reader side, mutual authentication, and privacy level. As a consequence, our proposal gives the highest privacy level with the lower communication cost and search time on the reader side for large scale RFID systems.

REFERENCES

1. Molnar, D. and D. Wagner, "Privacy and security in library RFID: issues, practices, and architectures", *Proceedings of the 11th ACM Conference on Computer and Communications Security*, CCS '04, pp. 210–219, ACM, New York, NY, USA, 2004.

2. Avoine, G., E. Dysli and P. Oechslin, "Reducing time complexity in RFID systems", *Proceedings of the 12th International Conference on Selected Areas in Cryptography*, SAC'05, pp. 291–306, Springer-Verlag, Berlin, Heidelberg, 2006.

3. Bringer, J., H. Chabanne and T. Icart, "Improved privacy of the tree-based hash protocols using physically unclonable function", *Proceedings of the 6th International Conference on Security and Cryptography for Networks*, SCN '08, pp. 77–91, Springer-Verlag, Berlin, Heidelberg, 2008.

4. 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.

5. 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 - TPDS*, Vol. 23, pp. 1–10, 06 2010.

6. Avoine, G., M. Bingol, X. Carpent and S. Yalcin, "Privacy-friendly authentication in RFID systems: On sublinear protocols based on symmetric-key cryptography", *IEEE Transactions on Mobile Computing*, Vol. 12, No. 10, pp. 2037–2049, 2013.

7. 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.

8. Hristea, C. and F. L. Tiplea, "Destructive privacy and mutual authentication in

Vaudenay's RFID model", Cryptology ePrint Archive, Report 2019/073, 2019, https://eprint.iacr.org/2019/073.

9. Kardaş, S., M. S. Kiraz, M. A. Bingöl and H. Demirci, "A novel RFID distance bounding protocol based on physically unclonable functions", *Proceedings of the 7th International Conference on RFID Security and Privacy*, RFIDSec'11, pp. 78–93, Springer-Verlag, Berlin, Heidelberg, 2012.

10. Hristea, C. and F. L. Ţiplea, "Privacy of stateful RFID systems with constant tag identifiers", *IEEE Transactions on Information Forensics and Security*, Vol. 15, pp. 1920–1934, 2019.

11. Kardaş, S., S. Çelik, M. Yıldız and A. Levi, "PUF-enhanced offline RFID security and privacy", *Journal of Network and Computer Applications*, Vol. 35, No. 6, pp. 2059–2067, Nov. 2012.

12. Ahsan, K., S. Hanifa and P. Kingston, "RFID applications: An introductory and exploratory study", *International Journal of Computer Science Issues*, Vol. 7, 02 2010.

13. Hasan, M. F. M., G. Tangim, M. K. Islam, M. R. H. Khandokar and A. U. Alam, "RFID-based ticketing for public transport system: Perspective megacity Dhaka", *2010 3rd International Conference on Computer Science and Information Technology*, Vol. 6, pp. 459–462, 2010.

14. Stockman, H., "Communication by means of reflected power", *Proceedings of the IRE*, Vol. 36, No. 10, pp. 1196–1204, 1948.

15. Cardullo, M. W. and W. L. Parks, "Transponder apparatus and system", *United State Patent*, Vol. 3713148, 1970.

16. 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.

17. Busch, H., S. Katzenbeisser and P. Baecher, "Information security applications", chap. PUF-Based Authentication Protocols — Revisited, pp. 296–308, Springer-Verlag, Berlin, Heidelberg, 2009.

18. 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*, pp. 148–160, 2002.

19. 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", *2004 Symposium on VLSI Circuits. Digest of Technical Papers (IEEE Cat. No. 04CH37525)*, pp. 176–179, June 2004.

20. Suh, G. and S. Devadas, "Physical unclonable functions for device authentication and secret key generation", *2007 44th ACM/IEEE Design Automation Conference*, pp. 9–14, June 2007.

21. Ozturk, E., G. Hammouri and B. Sunar, "Towards robust low cost authentication for pervasive devices", *2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 170–178, March 2008.

22. Guajardo, J., S. S. Kumar, G.-J. Schrijen and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection", *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 63–80, Springer, 2007.

23. Su, Y., J. Holleman and B. Otis, "A digital 1.6 pJ/bit chip identification circuit using process variations", *IEEE Journal of Solid-State Circuits*, Vol. 43, No. 1, pp. 69–77, Jan 2008.

24. Van der Leest, V., G.-J. Schrijen, H. Handschuh and P. Tuyls, "Hardware intrinsic security from D flip-flops", *Proceedings of the Fifth ACM Workshop on Scalable*

Trusted Computing, pp. 53–62, 2010.

25. Tuyls, P. and L. Batina, "RFID-tags for anti-counterfeiting", *Cryptographers' Track at the RSA Conference*, pp. 115–131, Springer, 2006.

26. 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.

27. Katzenbeisser, S., Ü. Kocabaş, 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", *International Workshop on Cryptographic Hardware and Embedded Systems*, pp. 283–301, Springer, 2012.

28. 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", *2008 IEEE International Conference on RFID*, pp. 58–64, April 2008.

29. Sadeghi, A.-R., I. Visconti and C. Wachsmann, "PUF-enhanced RFID security and privacy", *Workshop on Secure Component and System Identification (SECSI)*, Vol. 110, 2010.

30. Shamir, A., "SQUASH–A new MAC with provable security properties for highly constrained devices such as RFID tags", *International Workshop on Fast Software Encryption*, pp. 144–157, Springer, 2008.

31. O'Neill, M. *et al.*, "Low-cost SHA-1 hash function architecture for RFID tags", *RFIDSec*, Vol. 8, pp. 41–51, 2008.

32. Cheon, J. H., J. Hong and G. Tsudik, "Reducing RFID reader load with the meet-in-the-middle strategy", *Journal of Communications and Networks*, Vol. 14, No. 1,

pp. 10–14, 2012.

33. Ohkubo, M., K. Suzuki and S. Kinoshita, "Cryptographic approach to "Privacy-Friendly" tags", *RFID Privacy Workshop*, MIT, Massachusetts, USA, November 2003.

34. Burmester, M., T. Van Le, B. De Medeiros and G. Tsudik, "Universally composable RFID identification and authentication protocols", *ACM Transactions on Information and System Security (TISSEC)*, Vol. 12, No. 4, pp. 1–33, 2009.

35. Hein, D., J. Wolkerstorfer and N. Felber, "ECC is ready for RFID–A proof in silicon", *International Workshop on Selected Areas in Cryptography*, pp. 401–413, 2008.

36. Hutter, M., M. Feldhofer and T. Plos, "An ECDSA processor for RFID authentication", *International Workshop on Radio Frequency Identification: Security and Privacy Issues*, pp. 189–202, 2010.

37. Lee, Y. K., K. Sakiyama, L. Batina and I. Verbauwhede, "Elliptic-curve-based security processor for RFID", *IEEE Transactions on Computers*, Vol. 57, No. 11, pp. 1514–1527, 2008.

38. Hellman, M., "A cryptanalytic time-memory trade-off", *IEEE transactions on Information Theory*, Vol. 26, No. 4, pp. 401–406, 1980.

39. Ranasinghe, D., D. Engels, P. Cole *et al.*, "Security and privacy: Modest proposals for low-cost RFID systems", *Auto-ID labs Research Workshop, Zurich, Switzerland*, Citeseer, 2004.

40. Bolotnyy, L. and G. Robins, "Physically unclonable function-based security and privacy in RFID systems", *Fifth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom'07)*, pp. 211–220, IEEE, 2007.

41. Vaudenay, S., "On privacy models for RFID", *Proceedings of the Advances in Cryptology 13th International Conference on Theory and Application of Cryptology and Information Security*, ASIACRYPT'07, pp. 68–87, Springer-Verlag, Berlin, Heidelberg, 2007.

42. Halderman, J. A., S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum and E. W. Felten, "Lest we remember: cold-boot attacks on encryption keys", *Communications of the ACM*, Vol. 52, No. 5, pp. 91–98, May 2009.

43. Mauw, S. and S. Piramuthu, "A PUF-based authentication protocol to address ticket-switching of RFID-tagged items", *Security and Trust Management*, Vol. 7783 of *Lecture Notes in Computer Science*, pp. 209–224, Springer Berlin Heidelberg, 2013.

44. Avoine, G., *Cryptography in radio frequency identification and fair exchange protocols*, Ph.D. Thesis, Citeseer, 2005.

45. Juels, A. and S. A. Weis, "Defining strong privacy for RFID", *ACM Transactions on Information and System Security (TISSEC)*, Vol. 13, No. 1, pp. 1–23, 2009.

46. Damgård, I. and M. Ø. Pedersen, "RFID security: Tradeoffs between security and efficiency", *Cryptographers' Track at the RSA Conference*, pp. 318–332, 2008.