Protecting Privacy and Ensuring Security of RFID Systems Using Private Authentication Protocols

Md. Endadul Hoque
Marquette University

PROTECTING PRIVACY AND ENSURING SECURITY OF RFID SYSTEMS USING PRIVATE AUTHENTICATION PROTOCOLS

by

Md. Endadul Hoque

A Thesis submitted to the Faculty of the Graduate School,
Marquette University,
in Partial Fulfillment of the Requirements for
the Degree of Master of Science

Milwaukee, Wisconsin

August 2010

ABSTRACT
PROTECTING PRIVACY AND ENSURING SECURITY OF RFID SYSTEMS USING PRIVATE AUTHENTICATION PROTOCOLS

Md. Endadul Hoque

Marquette University, 2010

Radio Frequency IDentification (RFID) systems have been studied as an emerging technology for automatic identification of objects and assets in various applications ranging from inventory tracking to point of sale applications and from healthcare applications to e-passport. The expansion of RFID technology, however, gives rise to severe security and privacy concerns. To ensure the widespread deployment of this technology, the security and privacy threats must be addressed. However, providing solutions to the security and privacy threats has been a challenge due to extremely inadequate resources of typical RFID tags. Authentication protocols can be a possible solution to secure RFID communications.

In this thesis, we consider RFID authentication protocols based on symmetric key cryptography. We identify the security and privacy requirements for an RFID system. We present four protocols in this thesis. First, we propose a lightweight authentication protocol for typical tags that can perform symmetric key operations. This protocol makes use of pseudo random number generators (PRNG) and one way hash functions to ensure the security and privacy requirements of RFID systems. Second, we define the desynchronizing attack and describe the vulnerabilities of this attack in RFID systems. We propose a robust authentication protocol that can prevent the desynchronizing attack. This protocol can recover the disabled tags that are desynchronized with the reader because of this attack. Third, we introduce a novel authentication protocol based on elliptic curve cryptography (ECC) to avoid the counterfeiting problem of RFID systems. This protocol is appropriate for the RFID tags that can perform the operations of ECC. Finally, to address the tradeoff between scalability and privacy of RFID systems, we propose an efficient anonymous authentication protocol. We characterize the privacy of RFID systems and prove that our protocol preserves the privacy of RFID tags and achieves better scalability as well.

ACKNOWLEDGEMENTS

Md. Endadul Hoque

I am sincerely thankful to the people who contributed directly and indirectly to this thesis. I highly appreciate the invaluable advice, the fruitful guidance, and the endless efforts of Dr. Sheikh Iqbal Ahamed, Director of Ubicomp Research Lab. His futuristic ideas and thoughtful views encouraged me to chase the research challenges during my Master’s study at Marquette University. I am thankful to Dr. Ahamed for giving me the opportunity to work with him. I am also grateful to the thesis committee members Dr. Douglas Harris, Dr. Praveen Madiraju, and Dr. Rong Ge for their invaluable comments and patience during the preparation of this thesis. I would like to thank my lab members for their help and advice during the last two years. My heartiest gratitude goes to my wife, Farzana Rahman, for her eternal love and never-ending support for the last eight years. I highly appreciate her precious help in my research endeavors. My special thanks go to my friends in Milwaukee for their encouragement, support and incredible sense of humor that have made my period of study at Marquette extremely enjoyable and fruitful. Finally, my gratitude goes to my family and my in-laws for their love, trust and support in every aspect of my life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1: Introduction

Radio Frequency IDentification (RFID) is identified as emerging technology that is utilized by various applications to automatically identify objects such as objects and/or assets. The main benefits of RFID systems are that they can provide automated contactless identification of a range of physical entities. Usually, an RFID tag is a wireless transponder. Information stored on a tag can be read by special transceivers called RFID readers without requiring line-of-sight.

The first known application of RFID was the “friend or foe” identification system used in fighter planes in World War II [Royal40]. After a few decades, RFID technology gains the attention because of its inherent capability of being used as a replacement for bar codes in supply chain and inventory management [Juels06]. Nowadays, because of its low cost and ease of use, RFID technology has become widespread, including in point of sale applications [Juels06], product tracking in supply chain management [Juels06, Kerschbaum09], automated fare collection in public transportation [Garfinkel05], animal tracking to child supervision, healthcare applications [Juels06].

1.1 Overview of an RFID system

An RFID system has three major components: RFID tags, RFID readers and a backend server. We now briefly describe each of these components.

RFID tags: An RFID tag is an identification device which usually has an identifier and which transmits data wirelessly using radio frequency (RF) in response to interrogation by an RFID reader. A tag is a tiny chip which consists of an antenna and an integrated circuit. This IC is used for storage, signal modulation, and signal demodulation. The antenna is used for the communication with readers via radio frequency. When a tag receives a specific radio signal, it automatically transmits a reply. An RFID tag is shown in Figure 1.1.

Depending on the power supply, tags are categorized into three types: *passive*, *active* and *semi-passive* tags. Passive tags have no power source on board. They draw power from an RF signal sent by the readers. Therefore, in the absence of a reader, they cannot communicate, not even compute. The second type is active tags which have on board power source. They do not have to rely on the readers (or on other devices) for computation and communication with the readers. Semi-passive tags are the combination of both the active and passive RFID technology. They have their own power source for only computation purpose, not for communication. They rely on the RF signals sent by the readers to send or receive data.

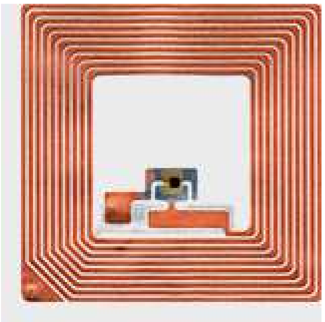


Figure 1.1: An RFID tag (source: http://www.barcode-solutions.com)

Depending on the processing capability, RFID tags can be categorized in two types: *dumb* and *smart*. Dumb tags have no significant processing power expect the communication capability for which they rely on the readers' RF signal. They contain fixed length unique identifiers which they reply in response to interrogation by readers. Even their memory capacity is fairly small – from few hundred bytes to maximum 2kBytes [Laurie07]. On the other hand, smart tags have on board processers that can capable of performing cryptographic operations [Laurie07]. Their memory capacity is much larger than that of dumb tags. They usually have 32kBytes or more memory [Laurie07]. They are capable of performing authentication before sending the stored data.

RFID readers: An RFID reader is a transceiver that interrogates and read data from tags. A reader uses its antenna to broadcast an RF signal which is used to start the interrogation.

Typically readers are connected to a backend server which has a database of tag information. Readers forward responses of tags to the backend server for further processing. An RFID reader is shown in Figure 1.2.

Figure 1.2: An RFID reader (source: http://www.barcode-solutions.com)

Backend server: A backend server is equipped with a database of tag information. It can retrieve the detailed information of a tag by using the tag's response as a key. Depending on the applications, the communication can be started by the backend server or by readers. Usually, when the backend server wants to identify one or more tags, a reader broadcasts an RF signal. Any tag within the range of the signal responds to the reader. The reader forwards the tag's response to the backend server. Then the backend server processes the response to identify the tag. If the server identifies the tag successfully, it can retrieve the detailed information of the tag. Figure 1.3 shows a typical RFID system.

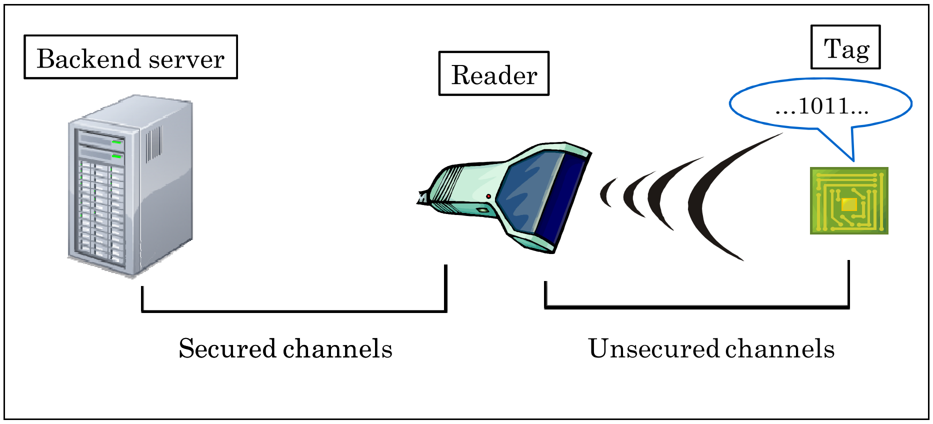


Figure 1.3: An RFID system

1.2 Motivation

The expansion of RFID technology is limited because it gives rise to serious security and privacy concerns, such as eavesdropping, cloning, impersonations, and tracking of end users. Since a reader and a tag communicate via a wireless radio communication channels, their interactions are susceptible to eavesdropping and/or manipulation. If a tag replies its unique identifier to authenticate itself to the reader, this fixed response becomes a signature of the tag which opens the possibility of being tracked. Thus the privacy of the tag holder can be invaded. So these security and privacy issues must be addressed before the widespread deployment of RFID tags.

Conventional security primitives cannot be integrated in RFID tags since they have inadequate computation capabilities with extremely limited resources. That is why research community devote themselves in search of appropriate solutions that will ensure RFID privacy and security without compromising the cost. Authentication can be one approach to address such security and privacy threats. An RFID system can use a tag authentication scheme in which a tag can be identified and verified by the backend server without disclosing the tag's identifier to the eavesdropper. In addition, authentication also ensures that only authorized reader can access a tag. Since tags are under heavy threats of adversaries, it is also mandatory to make sure that the tag's reply is accurate.

Public key cryptography is infeasible for the current RFID tags because of their limited resources. However, the current tags can perform symmetric key cryptography such as hash functions and symmetric encryption algorithms. So far various authentication protocols to address the security and privacy threats in RFID system by using symmetric key cryptography have been proposed in literature. We discuss some relevant protocols in the chapter 2 but they all have some limitations in terms privacy, security and/or performance. This motivates us to develop new authentication protocols that can perform better as well as address the security and privacy challenges. In this thesis, we focus on the solutions to the security and privacy threats at the protocol level, though some hardware based solutions have been proposed to ensure the security of RFID systems.

1.3 Main contributions

The main contributions of this thesis are as follows:

1. We identify the security and privacy requirements for an RFID system.
2. Our contributions include the assessment of some prior authentication protocols against the identified security and privacy requirements.
3. We present a lightweight authentication protocol for typical RFID tags that can perform symmetric key cryptography such as hash functions. This protocol addresses the identified security and privacy requirements.
4. We define desynchronizing attack and propose a robust authentication protocol that supports recovery against this attack. We analyze this protocol against the identified security and privacy requirements for an RFID system.
5. We introduce a novel authentication protocol based on Elliptic Curve Cryptography (ECC) to avoid the counterfeiting problem in RFID systems. This protocol is appropriate for an RFID system where tags are capable of performing the operations of elliptic curve cryptography.
6. Finally, we characterize the privacy of RFID systems. We present an efficient anonymous authentication protocol that addresses the tradeoff between scalability and privacy of RFID systems. This protocol improves the scalability of an RFID system. However, it not only prevents the security attacks but also preserves the privacy of tags. In fact, we prove that our protocol preserves the privacy of RFID tags.

1.4 Thesis Organization

The remainder of this thesis is organized as follows:

1. In chapter 2, we present the security and privacy requirements for an RFID system along with the assessment of some prior works against the identified requirements.
2. We present our lightweight authentication protocol in chapter 3. In addition to the protocol description, we describe an attack model and analyze the protocol with respect to this attack model.
3. In chapter 4, we define the desynchronizing attack and describe how this attack can disable tags so that the tags cannot be identified by the valid reader in future. We propose a robust authentication protocol that not only recovers the RFID system from this disabled state to the normal operating state but also supports the identified security and privacy requirements.
4. In chapter 5, we introduce an ECC based authentication protocol for RFID systems where tags can perform ECC based operations. We present the security and privacy analysis of the protocol with respect to a proposed attack model.
5. In chapter 6, we propose an efficient anonymous authentication protocol as a solution to the tradeoff between the scalability and privacy problem of RFID systems. We define the characteristics of RFID privacy. Based on this notion of RFID privacy, we prove that our protocol protects privacy of RFID tags and thereby the privacy of tag holders.
6. Finally in chapter 7 we summarize the contributions of the thesis and present some future research directions.

1.5 Publications

Some materials of this thesis have been published in [Ahamed08A, Ahamed08B, Hoque09A, and Hoque09B]. The protocol of chapter 3 has been first published in [Ahamed08A]. A extended version of this work can be found in [Hoque09A]. The contents of [Hoque09B] form the basis of chapter 4. The protocol of chapter 5 can be found in [Ahamed08B].

Chapter 2: Related Work

To design a solution to security and privacy threats, the goals or requirements need to be identified. In this chapter, we first present our identified security and privacy requirements for an RFID system. Then we assess some prior works against the identified requirements. Though a considerable volume of papers have been published so far, here we present some prior works that are quite relevant to our proposals.

2.1 Security and privacy requirements for an RFID system

Several real life applications of RFID systems require them to be secure and protective against security and privacy related attacks. Considering those applications and analyzing their security requirements, we identify the following security and privacy goals for an RFID system. A safety ring composed of all these security and privacy goals is depicted in Figure 2.1. An RFID system ensuring all the six elements of this safety ring is considered to be secured and protected against all major attacks. The elements of the safety ring are explained in the following.

1) Protect Privacy: RFID technology raises privacy concerns in some situations. For example, consumer privacy is invaded when the use of RFID enables unauthorized parties to obtain personally identifiable information, including location information of the tag holder. So it should be guaranteed that a tag or its secret data cannot be distinguished without tampering it.

2) Prevent Tracking: Consumer community never wants to be tracked. Therefore, preventing tracking is another major goal of authentication protocol. If an adversary does not have any identifiable information of a tag, then she cannot track the tag. But if a tag replies with a constant response each time it is queried, then it becomes a signature of the tag. As a result, this signature allows the adversary to track the tag. So it should be ensured by the protocol that a tag neither reveals its unique identifier nor replies with a constant response.

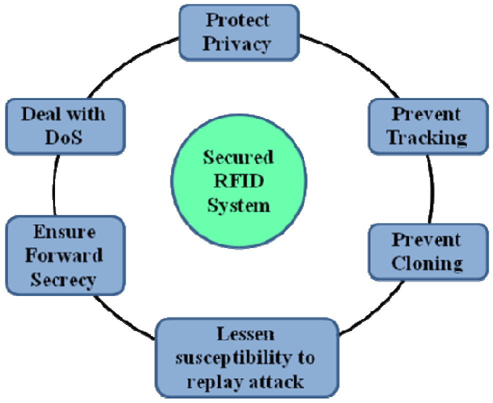


Figure 2.1: Six elements of the "Safety Ring"

3) Deal with Denial of Service (DoS) attack: DoS attack means that an entity is prevented from accessing its authorized entities. Therefore, the availability of an RFID system mainly depends on the assurance of this goal. An RFID system should continue running and provide service to its authorized users even if an adversary launches DoS attack. Since it is not possible to detect all kinds of DoS attack, authentication protocols should at least provide a way to deal with them. Protocols should be able to take measure against vulnerable actions of the adversaries and recover the system that is under the DoS attacks.

4) Ensure Forward Secrecy: Forward secrecy means that if an adversary compromises a tag and learns the secret key shared between the tag and the reader, she will be unable to identify the previous responses originated from the tag. In order to maintain the security of RFID systems, forward secrecy should be ensured by authentication protocol.

5) Lessen susceptibility to replay attack: An adversary can launch a replay attack by just replying a message that she eavesdrops from an earlier authentication session between a reader and a tag. Under this attack, the adversary can fool the reader as well as the tag. She can impersonate the tag by replying the tag's response to the reader. An authentication protocol must ensure that an attacker cannot impersonate a legitimate tag by just replaying an eavesdropped message.

6) Prevent Cloning: One important application of RFID systems is to detect counterfeit products. In order to avoid counterfeiting, RFID tags need to be designed in such a way that an adversary cannot clone the tags. Since this is a hardware based solution to the cloning attack, we are interested in the solutions to attacks at the protocol level. The adversary can clone a tag if she knows the secret key shared by the tag with the authorized reader. So, to be secured against cloning attack, protocols should never reveal the shared secret key.

2.2 Relevant authentication protocols

We categorized the prior works in two classes: *symmetric key based schemes and public key based schemes*. Although some schemes focus on providing the security solutions integrated in basic RFID tags (e.g., blocker tags [Juels03], RFID Guardian [Rieback05]), here we consider only the protocol level solutions.

2.2.1 Symmetric key schemes in RFID

In this subsection, we describe the most significant symmetric-key protocols proposed for RFID security and privacy.

Deterministic hash locks protocol. Weis et al. [Weis03] first proposed a cryptographically controlled access for RFID systems using *hash locks*. This scheme controls access to a tag by locking or unlocking the tag using a one way hash function h. Each tag stores an identifier and the hash value of a secret key. A tag replies with the hash value in response to a reader's query. The valid reader can look up the secret key of the tag in a database of key-hash pairs. The valid reader proves itself by sending the tag's secret key so that the tag can verifies the key using the hash function by comparing with the stored hash value. Finally, the tag releases its data if the correct key is given by the reader. This protocol can protect the actual data on the tag, but the static hash value would still be traceable.

Randomized hash locks protocol. To address the problem of the deterministic hash locks protocol, Weis et al. [Weis03] proposed randomized hash locks protocol, a modified version of the above protocol. This protocol makes use of a nonce[1] to introduce randomness in tags' responses. A tag generates a nonce r and sends the pair $(r, h(ID \parallel r))$ to the server each time the tag is queried. This protocol addresses the traceability problem, but prompts the reader to brute-force its inventory for any ID that matches the given hash if concatenated with r. In addition, this protocol is susceptible to replay attack, where an adversary can reply with the tag's response that she learned by eavesdropping in an earlier authentication session. The reader accepts this replay response as legitimate.

Improved Randomized hash locks protocol. Juels and Weis [Juels07] proposed an improved randomized hash locks protocol for RFID systems that protects the privacy of RFID tags as well as prevent replay attack. In this protocol, the server generates a nonce r_1 and queries the tag with this nonce. The tag produces another nonce r_2 and sends the pair $(r_2, h(ID \parallel r_1 \parallel r_2))$ as the response. Therefore the server has to perform a linear search in the database for any ID that matches the given hash if concatenated with r_1 and r_2. This protocol offers strong tag privacy in front of eavesdroppers. However, the limitation of this protocol is poor scalability.

OSK protocol. Ohkubo et al. [Ohkubo03] proposed an RFID privacy protection scheme designed to protect against tracking and provide forward security[2]. The protocol makes use of *hash-chains*. The server stores a secret key s_i^0 and a identifier ID_i for each tag T_i. The tag T_i initially stores the secret key s_i as s_i^0. The tag replies with a hash value of the key $h_1(s_i)$ to the server and updates the key as $s_i \leftarrow h_2(s_i)$. The server then identifies the tag via an exhaustive search, computing $h_1\left(h_2^j(s_i^0)\right)$ for each tag in the database until it finds a match with $h_1(s_i)$, where h_2^j means the j iterations of the function h_2. This protocol is susceptible to replay attack

[1] Nonce is a random or pseudo random number that never repeats its value.

[2] Forward security means that if an adversary compromises a tag, she will not able to trace back the entire history of the tag's responses.

where an adversary can impersonate a tag with knowing the secret key. In addition, this protocol also has the scalability problem.

Dimitriou protocol. Dimitriou [Dimitriou05] proposed a mutual authentication of both tags and the server. The general idea is that the server updates a tag's identifier if the tag proves its identity to the server and the tag updates its own identifier only when the server proves its validity to the tag. Thus this protocol keeps both the server and the tag always in perfect synchronization. Though this protocol protection against tag cloning, it is subject to tracking and denial-of-service attack. The response of the tag is static between two valid sessions, and thus it makes the system susceptible to tracking. In addition, if the server's response (that the server sends to the tag to prove its validity) does not reach the tag in a session, the tag becomes desynchronized with the server.

Molnar-Wagner protocol. A tree based protocol first proposed by Molnar and Wagner [Molnar04] reduces the reader's complexity from linear search to logarithmic search. This is a tree based approach where keys are arranged in tree and tags are assigned to the leaves. Instead of a single secret key, each tag stores a series of keys along the path of the tree from the root to the leaf assigned to the tag. After receiving a challenge from the server, the tag replies its response using all its keys. The server, then, finds the tree from the root to the leaves to verify whether the tag has a valid key at each level of the tree. According to this scheme, the server needs $O(\log_k N)$ operations to identify a tag, where N is the number of tags in the system and k is the branching factor of the key tree. Though this scheme provides scalability, however, it sacrifices some privacy of tags when any tag is compromised by the adversary.

YA-TRAP protocol. A trivial RFID tag identification protocol proposed by Tsudik [Tsudik06] uses timestamps to improve scalability and provide protection against tracking. The server queries a tag with the current timestamp ($t_i{}'$). The tag replies a random number if $t_i' \leq t_i$ or $t_i' > t_m$, where t_i is the timestamp stored on the tag and t_m is the maximum possible timestamp. If the server is valid, the server's timestamp would be $t_i \leq t_i' < t_m$. Upon receiving a query from the valid server, the tag updates its $t_i \leftarrow t_i{}'$ and replies with a keyed hash value $f_{k_i}(t_i)$, where k_i is the unique secret key of the tag. The server then identifies the tag by finding the response in its lookup table. This protocol needs $O(1)$ operations to identify a tag and thereby improves scalability. However, this scheme is susceptible to denial-of-service attack.

HB$^+$-protocol. To lower the requirements of cryptographic functions for RFID tags, Juels and Weis [Juels05] proposed a lightweight authentication protocol based on the famous human-to-computer authentication, Hopper and Blum (HB) protocol. This simple symmetric authentication protocol requires low-cost implementation. The hardness of this protocol is same as the Learning Parity with Noise (LPN) problem. However, this protocol is vulnerable to an active attack like message manipulation.

Tan-Sheng-Li protocol. The server of an RFID system can become a single point-of-failure if the server is compromised by an adversary. A serverless protocol first proposed by Tan et al. [Tan07] removes the vulnerability of a single point-of-failure in RFID systems. In this protocol, the reader, instead of the server, maintains a list of secret information of the tags that the reader has access to. Contrary, each tag has a secret t. The tag shares this secret information with no one of the system, not even with the reader. However, both the reader and the tag know $f(r,t)$, where r is reader identifier and $f(\cdot,\cdot)$ is a hash function. In response to the query from the reader, tag replies with some of the bits of $h(f(r,t) \parallel n_i \parallel n_j))$, where n_i and n_j are two random numbers generated by the reader and the tag, respectively and $h(.)$ is a one way hash function. Along with the response, the tag queries the reader with a question string. Only the legitimate reader can reply with the valid answer string which proves the reader's legitimacy to the tag. The tag releases its data after checking that the reader is valid. But this protocol suffers because of poor scalability. Even the protocol 2 [Tan07] is not completely anonymous as the tag replies with some bits of its identifier.

2.2.2 Public key schemes in RFID

Public key cryptography is not feasible because of the inadequate computation and storage complexity of RFID tags. Typical RFID tags are not capable of performing the expensive operations of public key cryptography. On the other hand, strong privacy must be achieved before the widespread deployment of RFID systems. Public key cryptography seems to the best solution to protect consumer privacy. Therefore, a lot of works has been done to analyze and adapt the public key protocols for RFID systems. In [Tuyls06], the author proposed that the Elliptic Curve Cryptography (ECC) can be implemented on RFID tags using less than 5000 gates. The feasibility of ECC implementation on RFID tags is also proposed in [Wolkerstorfer05, Hein09]. These public key cryptography based techniques for identification are mainly based on Elliptic Curves Discrete Logarithmic Problem (ECDLP) [Okamoto92]. Some of the significant protocols for RFID systems based on public key cryptography are proposed in [Batina06A, Batina06B, Bringer08, Deursen09, Lee08A, Lee08B, and Lee09]. We will not describe these protocols here, since our focus is on the symmetric key protocols for RFID security and privacy.

Chapter 3: A Lightweight Serverless RFID Authentication Protocol

3.1 Introduction

So far all the security and privacy requirements of RFID systems were ensured by central databases. This server based model has drawn much consideration and some of the outcomes are reflected in [Avoine05, Burmester06, Conti07, Cui07, Seo06A, Seo07B, Tsudik06 and Vajda03]. According to the fundamental architecture of an RFID system, a reader scans a tag and relays the information to the backend server. Other than the back end database, no other component of this system (not even a reader) is able to infer any information from the tag's response since it is encrypted. The server returns an acknowledgement (sometimes the tag's data) to the reader only after verifying both the tag and the reader. Typically only the central server can authenticate the involved tag and the reader. In the server based system the central server indeed plays an essential role of checking the validity of the tag and the reader, which is very important for privacy protection and security issues. Consequently a malicious reader could hardly obtain precious information from the tags of this server based system.

The major drawback of such system is that the readers always have to be connected to the server, which limits usage of RFID systems in remote locations where the connectivity with the server cannot always be ensured. Besides, having a single database makes the whole system more vulnerable to privacy attacks. Having the knowledge of all tag secrets and tag information, the central server becomes a single point-of-failure. As a result, if the server is captured by an adversary, privacy of the entire user community is jeopardized. Therefore a serverless RFID system is proposed in [Tan07] addressing the shortcomings of the server based system. Tan *et al.* paper introduces an RFID system to the world where a gigantic central server is not a single point-of-failure anymore.

An alternative solution, but analogous to the central server system, is to transfer the information of the tags from the central server to the reader that has access to these tags. In such serverless system, a reader has to identify legitimate tags all by itself because of the absence of server. At the same time, in order to receive the tag's data the reader has to prove himself as legitimate to the tag. But because of the mobile nature of the readers, they can be stolen like tags. An adversary with a stolen reader can have access to all the information of the tags that the stolen reader has access to. This information may include the *ID* and the *tag secret* number of the tags. Any single pair can be loaded into a blank tag by the adversary. This fake tag can now impersonate the legitimate tag and as a result a valid reader cannot distinguish between these two tags. This is a severe breach in the security of an RFID system.

Existing solutions cannot be applied to the security problem of the RFID systems because of the limited resources of the tags. Due to the lack of security, the use of RFID technology has been restricted to a closed set of pervasive applications. However, the number of applications will be increased if secure serverless systems are introduced. One of the major advantages of the serverless RFID system is that it reduces the cost of system deployment in large application areas. On the other hand, to ensure security and privacy of the system, lightweight solutions are required because of the resource limitation of the tags. In this chapter we propose a lightweight authentication protocol that can provide security and privacy protection similar to the central server model without having persistent connection with the server. A version of this protocol has been presented in [Ahamed08A, Hoque09A].

3.2 System architecture

The RFID system generally consists of tags, the reader, and the backend server. However, ours is a serverless system. The serverless RFID system primarily consists of two entities the reader and a set of tags. A certification authority is involved in the system to certify the readers and authorize them to particular tags. Next we discuss components of an RFID system.

Tag: Each tag T is comprised of an IC chip and an antenna. The tag sends information to the RFID reader in response through wireless medium. We focus on passive tags which are expected to be the most common type of RFID tags. In our system, each tag is able to communicate with one reader at a time.

Reader: A reader R is a device that sends some query using the radio frequency signal to a tag, receives the response from the tag, and performs some important computation on those responses.

Certification authority: The certification authority CA is a trusted party that plays the crucial role during the deployment. CA initializes each tag by writing the *tag secret* into the tag's memory. Moreover the certification authority certifies each reader in the system and authorizes each reader's access to a particular set of tags. The communication channel between the reader and the tag is assumed to be vulnerable to several attacks. However, the channel between the reader and the CA is assumed to be secure.

3.3 The lightweight authentication protocol

3.3.1 Notations and assumptions

All the notations of this protocol are presented in Table 3.1. Each tag and each reader have the knowledge of two functions $\mathcal{P}(.)$ and $h(.)$. The function $\mathcal{P}(.)$ is a fairly simple random number generator that can be implemented at low cost. $\mathcal{P}(.)$ takes a seed as an argument and outputs a pseudorandom number according to its distribution. $h(.)$ is used by all the readers and the tags to generate the next seed of the pseudorandom number generator by passing the current seed as input. $h(.)$ is an irreversible one way hash function. Therefore a current seed cannot be linked with the previous one.

Table 3.1: Notations for the lightweight authentication protocol

Symbol	**Meaning**
R_i	ith reader of the system
T_j	jth tag of the system
$\mathcal{P}(.)$	Pseudo random number generator
$h(.)$	One way hash function
r_i	the identifier of the ith reader
$\mathcal{L}_i$	the contact list of R_i
id_j	the identifier of the jth tag
t_j	the secret of the jth tag
$seed_j$	R_i receives this seed from CA for T_j
$seed_{T_j}$	T_j receives this seed for the R_i from CA
$\hat{A}$	the adversary

A reader R_i has a unique identifier r_i and a contact list $\mathcal{L}_i$. R_i obtains r_i and $\mathcal{L}_i$ from the CA during the system deployment. On the other hand, the CA initializes each tag T_j with a unique identifier id_j and a unique secret t_j by writing in his nonvolatile memory. The contact list $\mathcal{L}_i$ contains information about the tags that R_i has access to. If R_i is authorized to access a set of tags, say $T_1, \cdots, T_n$, $\mathcal{L}_i$ becomes,

$$\mathcal{L}_i = \{< seed_j, id_j > | id_j = \text{identifier of } T_j \text{ and } seed_j = \text{next seed of } T_j \text{ and } 1 \leq j \leq n\}.$$

In other words,

$$\mathcal{L}_i = \{< seed_1, id_1 >, \ldots, < seed_n, id_n >\}.$$

Now we discuss the seed mentioned above. Each tag T_j contains only one seed for the one authorized reader R_i. While the tag T_j is deployed by the CA, T_j receives $seed_{T_j} = h(r_i \parallel t_j)$ from the CA where $h(.)$ is one way hash function and $\parallel$ represents concatenation. T_j stores the $seed_{T_j}$ in the tag's nonvolatile memory. On the other hand, the reader R_i receives the contact list $\mathcal{L}_i$ during the deployment where $seed_j = h(r_i \parallel t_j), \forall j \in [1, n]$. The reader R_i uses these seeds to communicate with the tags. Note that R_i does not know the tag secret t_j. The reader only knows the outcome of the function $h(r_i \parallel t_j)$ as $seed_j$. Since the initial $seed_j$ is computed by the CA, the reader R_i can never learn t_j from the received $seed_j$. We also assume that CA cannot be compromised by any adversary. In this chapter we denote an adversary by $\hat{A}$.

```
$R_i \rightarrow T_j$ :  $request, rand_i$
$T_j$             :  generate $rand_j$
                     set $n_j = \mathcal{P}\left(seed_{T_j} \oplus \left(rand_i \parallel rand_j\right)\right)$
$R_i \leftarrow T_j$ :  $n_j, rand_j$
$R_i$             :  set $n_i = rand_i$
                     for each $< seed_m, id_m > \in \mathcal{L}_i$
                         set $n_m = \mathcal{P}\left(seed_m \oplus \left(rand_i \parallel rand_j\right)\right)$
                         if ( $n_m == n_j$ ) then
                             set $s = h(seed_m)$
                             set $n_i = \mathcal{P}(s)$
                             set $seed_m = h(s)$
                             break
                         end if
                     end for
                     $R_i \rightarrow T_j : n_i$
$T_j$             :  set $k = h\left(seed_{T_j}\right)$
                     set $a = \mathcal{P}(k)$
                     if ($a == n_i$) then
                         $seed_{T_j} = h(k)$
                     else
                         $Reader\ is\ not\ authorized$
                         $Or\ it\ is\ an\ adversary$
```

Figure 3.1: The lightweight authentication protocol

3.3.2 Protocol description

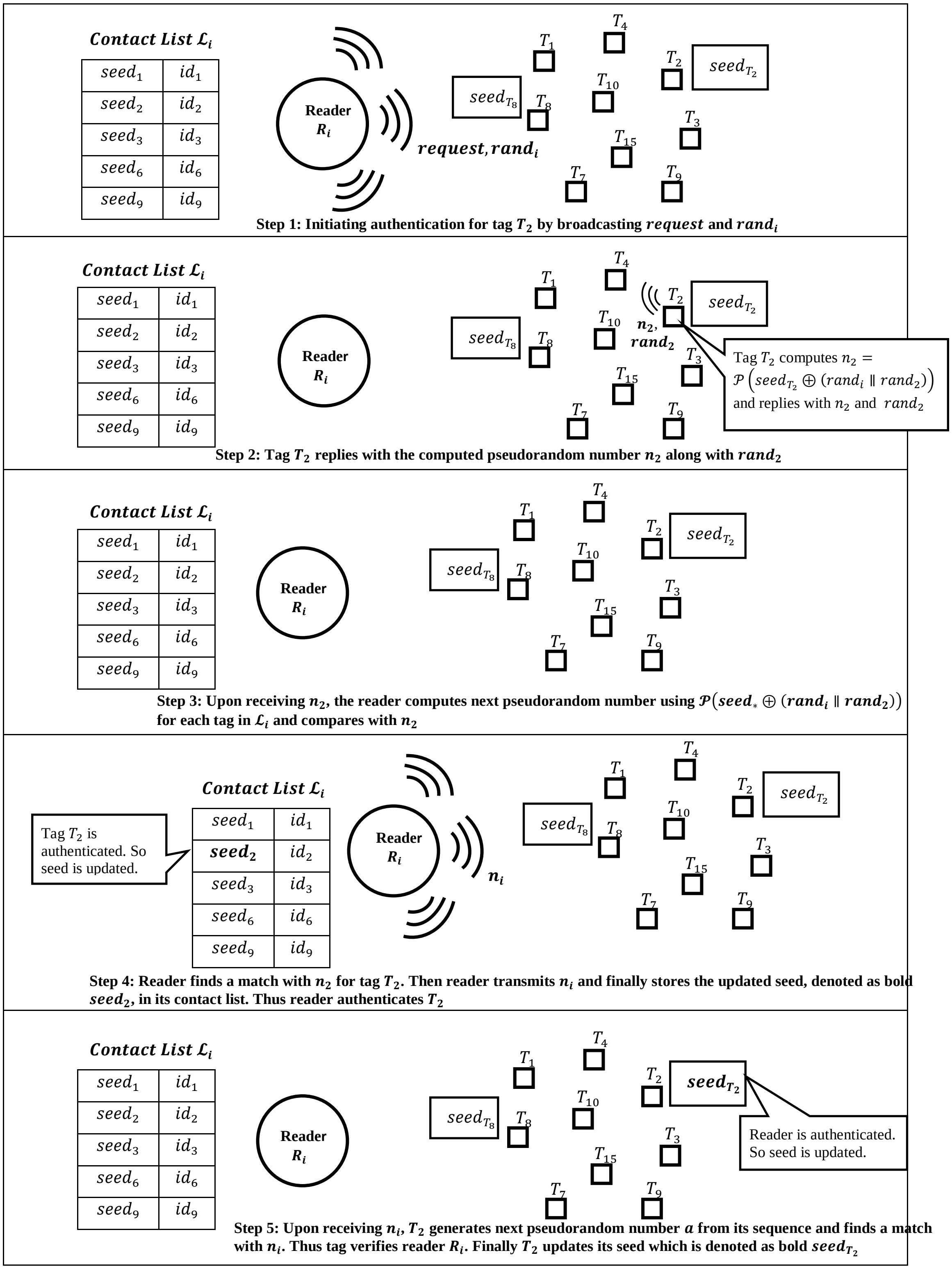
The protocol is shown in Figure 3.1. At the beginning, the reader R_i transmits a $request$ and a random number $rand_i$ to the tag T_j. Upon the reception of the $request$ and the $rand_i$, T_j generates a random number $rand_j$ and computes $n_j = \mathcal{P}\left(seed_{T_j} \oplus \left(rand_i \parallel rand_j\right)\right)$. To prove own legitimacy, the tag replies with n_j and $rand_j$. Now the reader has to verify the legitimacy of the tag. The reader calculates $n_m = \mathcal{P}\left(seed_m \oplus \left(rand_i \parallel rand_j\right)\right)$ for each $< seed_m, id_m > \in \mathcal{L}_i$ and searches for a match between the received n_j and the produced n_m. If the reader finds a match, the reader becomes sure about the validity of the tag. Since it is a mutual authentication, the reader has to prove itself to the queried tag. After identifying the tag, the reader will produce a pseudo random number (n_i) after generating the next seed from the seed of the particular tag using the hash function and will update the new seed after hashing the current seed (produced to compute n_i) using the same function. If the reader fails to find a match, it generates a random number ($n_i = rand_i$) and concludes T_j is a fake tag. Then, the reader replies n_i to the tag. To verify the validity of the reader, the tag produces a pseudo random number (a) after generating the next seed from its current seed $seed_{T_j}$. If the tag finds a match between a and n_i, the tag also updates its own seed using the same hash function and concludes R_i as the authorized reader. Otherwise, T_j decides that the reader is not an authorized one or this is an adversary. In this protocol, both the reader and the tag update their seeds after they confirm the validity of the opposite party.

3.3.3 Interaction diagram

Figure 3.2 shows a detailed interaction diagram of our authentication protocol where the reader R_i is communicating with the tag T_2.

3.4 Attack model

The major goal of an adversary in any RFID system is to counterfeit a real tag such that the fake tag can only be distinguished from the real one with a small probability. As a result, the fake tag (the fake product as well) will be identified as a legitimate one. In this chapter, an adversary is denoted as $\hat{A}$. This adversary can control a number of readers and tags. Each reader and tag controlled by the adversary are denoted as $\hat{R}$ and $\hat{T}$, respectively. $\hat{R}$ is unauthorized to have access to any real tags as this adversarial reader fails to prove own identity to the CA. Similarly, $\hat{T}$ is not valid since it has no idea about the seed and the tag secret. We presume that the certification authority cannot be compromised. Otherwise, the adversary would get total control over all the tags. We also assume that all the entities such as tags, readers, adversaries, adversarial tags and adversarial readers have polynomially bounded resources.



Step 1: Initiating authentication for tag T_2 by broadcasting $request$ and $rand_i$

Tag T_2 computes $n_2 = \mathcal{P}\left(seed_{T_2} \oplus (rand_i \parallel rand_2)\right)$ and replies with n_2 and $rand_2$

Step 2: Tag T_2 replies with the computed pseudorandom number n_2 along with $rand_2$

Step 3: Upon receiving n_2, the reader computes next pseudorandom number using $\mathcal{P}(seed_* \oplus (rand_i \parallel rand_2))$ for each tag in $\mathcal{L}_i$ and compares with n_2

Tag T_2 is authenticated. So seed is updated.

Step 4: Reader finds a match with n_2 for tag T_2. Then reader transmits n_i and finally stores the updated seed, denoted as bold $\boldsymbol{seed_2}$, in its contact list. Thus reader authenticates T_2

Reader is authenticated. So seed is updated.

Step 5: Upon receiving n_i, T_2 generates next pseudorandom number a from its sequence and finds a match with n_i. Thus tag verifies reader R_i. Finally T_2 updates its seed which is denoted as bold $\boldsymbol{seed_{T_2}}$

Figure 3.2: Detail interactions of the authentication protocol between the reader R_i and the tag T_2

Our assumption includes that $\hat{A}$ is more powerful than a passive attacker. Like a passive attacker, she can eavesdrop on the both the channels between a valid reader and a valid tag. However, like an active attacker, $\hat{A}$ can install a rouge reader $\hat{R}$ that can communicate with a valid tag. In addition, the adversary can also install a fake tag $\hat{T}$ to communicate with a legitimate reader. In both cases, the ultimate goal of the adversary is to counterfeit a tag with the learned information. In addition to these attacks, $\hat{A}$ can launch hardware based physical attacks, while these hardware based physical attacks are beyond the scope of this chapter. We will address the attacks on the protocol layer.

3.5 Analysis of the protocol

We analyze our protocol in two steps. First we present the security and privacy analysis followed by the cost analysis.

3.5.1 Security and privacy analysis

In this subsection, we analyze our proposed authentication protocol against different types of attacks. For every attack, we first describe how the attack is launched by an adversary. Then we explain how our protocol protects the system against the attack. R_i and T_j are referred to as a legitimate reader and tag.

Privacy protection: If an adversary $\hat{A}$ comes across any private information of the tag by querying it, she may cause several vulnerabilities to the owner's day to day life. We assume that $\hat{A}$ may target a list of tags and attack each tag for a fixed number of times. $\hat{A}$ queries the tags to discover the id of the tag and thereby some private information of the product or the tag owner. But our protocol strongly protects user privacy because a tag never sends its own id to anyone, not even to the authorized reader. Each tag sends its reply in disguise so that only an authorized reader can identify the tag. Moreover, no one is able to infer or learn the id of the tag by simply looking at the tag's responses or by simply querying the tag. Under this attack, the adversary has a list of targeted RFID tags. The adversary queries each tag within the group to determine which tags of her list exist within this group. According to our protocol, each time a reader queries a tag T_j, it replies to the reader with a new response $\mathcal{P}\left(seed_{T_j} \oplus (rand_i \parallel rand_j)\right)$. Therefore $\hat{A}$ fails to link any two responses and thereby fails to identify which one of the tags is replying. Thus our protocol protects the privacy of the tag.

Tracking: The adversary $\hat{A}$ tries to track T_j over time. $\hat{A}$ succeeds if she is able to distinguish T_j from other tags. Under this attack, $\hat{A}$ repeatedly queries T_j with a value which yields a consistent reply. This consistent reply becomes a signature of T_j. To launch this attack, $\hat{A}$ can reuse the same $rand_i$ learned from any previous session of our protocol. By incorporating $rand_j$, our protocol becomes secured against tracking since $\hat{A}$ cannot predict $rand_j$ ahead of time. Consequently, T_j will reply a new pseudo random number each time it is queried. Moreover, when $\hat{A}$ learns the $rand_i$ from any previous successful authentication session, $seed_{T_j}$ has been updated to a new value. Thus, $\hat{A}$ fails to get any consistent reply from T_j. As a result, she cannot track T_j afterwards.

Cloning: Under this attack, $\hat{A}$ queries T_j and places the response in a fake tag. Let this fake tag be $\widehat{T_j}$. Now, $\hat{A}$ wants to pass off the forged tag as legitimate and she becomes successful if she can fool a legitimate reader R_i. According to our protocol, whenever the adversary queries T_j, she receives a different response each time because of $rand_i$ and $rand_j$. Now, if $\hat{A}$ places this response in $\widehat{T_j}$, she will never be able to fool a valid R_i. When $\widehat{T_j}$ is queried by R_i, $\widehat{T_j}$ cannot generate the actual response. $\widehat{T_j}$ fails because for each query, R_i transmits a new $rand_i$ that $\hat{A}$ cannot predict at the time of producing the fake tag. On the other hand, since $\widehat{T_j}$ does not know the current seed stored on T_j, the fake tag cannot generate the actual response while queried by R_i.

Denial of Service (DoS): In this attack, $\hat{A}$ does not want to derive any information or impersonate the reader or the tag. The main target of the adversary is to ensure that a reader cannot access its authorized tags. To launch a DoS attack, $\hat{A}$ places many requests to the backend server so that the readers are unable to communicate with the backend server. This problem becomes severe when the backend database shares a secret key with the tag that has to be synchronized after each successful authentication. Our protocol eliminates the need of a backend server. So, synchronization between the server and the tag is not required any more. Moreover, in our scheme, a reader communicates with the backend server only at the time of deployment.

Physical attack: Physical attack means $\hat{A}$ can compromise either a tag or a reader. We will consider both the cases.

A. ***$\hat{A}$ compromises R_i:*** When $\hat{A}$ compromises a reader R_i, the adversary will know the reader's contact list $\mathcal{L}_i$ and the id r_i. She can now impersonate R_i and prove herself to T_j, if the reader has been authorized to access T_j. Eventually, she is able to access the tags $T_1, T_2, \dots, T_n$, if R_i has access to these tags. Now, the security goal is to prevent $\hat{A}$ from using this knowledge to create any counterfeit tag. Let T_j resides in the contact list $\mathcal{L}_i$, and $\hat{A}$ wishes to counterfeit the tag T_j which we name $\widehat{T_j}$. The adversary will be successful if $\widehat{T_j}$ can fool another legitimate reader R_x. But under our scheme, only one reader is authorized to access T_j and that reader is R_i. So, $\widehat{T_j}$ cannot fool R_x by learning $seed_j$ and id_j from $\mathcal{L}_i$.

B. ***$\hat{A}$ compromises T_j:*** In this case, the adversary compromises a tag T_j. $\hat{A}$ is able to create a fake tag $\widehat{T_j}$ that can fool an honest reader R_i. We want to prevent $\hat{A}$ from counterfeiting another valid tag that can fool R_i. Since the adversary has compromised T_j, we assume that the adversary knows all the private information of T_j. With this information, $\hat{A}$ wants to clone a valid tag T_x. With this cloned tag, $\hat{A}$ wants to spoof an honest reader R_i that is authorized to access T_x. Since each RFID tag shares a seed with its authorized reader, T_x shares a different seed with R_i which is not known to T_j. Though $\hat{A}$ knows $seed_{T_j}$, however, she cannot derive the shared seed between R_i and T_x. Therefore, the adversary cannot create the fake tag to fool R_i.

Eavesdropping: $\hat{A}$ eavesdrops the communication between R_i and T_j and later uses this information to launch any of the attacks mentioned above. $\hat{A}$ can learn every information exchanged between R_i and T_j such as $rand_i$, n_j, $rand_j$ and n_i. We assume that $\hat{A}$ can listen to both the tag-to-reader and reader-to-tag communication channels. According to our protocol, $\hat{A}$ cannot launch privacy attack as the protocol does not reveal any sort of private information of the tag and the reader. Even $\hat{A}$ fails to track T_j, because the tag replies with a new pseudo random number every time it is queried. Thus, $\hat{A}$ cannot figure out any signature to follow T_j.

Under our protocol, listening over the communication channels cannot help $\hat{A}$ to launch a cloning attack. $\hat{A}$ cannot create a fake tag $\widehat{T_j}$ by learning only the pseudo random numbers exchanged between R_i and T_j. Since $\hat{A}$ cannot predict $rand_i$ and have no idea about $seed_{T_j}$, it is impossible for $\hat{A}$ to clone T_j. As a result, she cannot fool a legitimate reader R_i. Suppose, $\hat{A}$ tries to impersonate the tag T_j and we name the fake tag $\widehat{T_j}$. This fake tag wants to fool an honest reader R_i with which T_j has communicated recently. Now, $\widehat{T_j}$ will not be able to deceive R_i, since the reader will definitely query with a new $\widetilde{rand_i}$. And $\widehat{T_j}$ fails to generate $\mathcal{P}\left(seed_{T_j} \oplus \left(\widetilde{rand_i} \parallel rand_j\right)\right)$ because it has no idea about the $seed_{T_j}$. Even $\widehat{T_j}$ fails to launch a *replay attack*. If $\widehat{T_j}$ replays with the same response $\mathcal{P}\left(seed_{T_j} \oplus \left(rand_i \parallel rand_j\right)\right)$ that is learned from an earlier authentication session, R_i will easily identify that it is a fake tag.

Forward secrecy: Forward secrecy means if anyhow an adversary compromises a tag, she will not be able to track down any data previously transmitted by the tag. It means that if $\hat{A}$ physically tampers T_j and learns $seed_{T_j}$shared with R_i, $\hat{A}$ will not be able to trace the data back through past events in which both the reader and the tag were involved. Our protocol ensures strong forward secrecy since the seed-update function $h(.)$ is an irreversible one way hash function. Therefore, by tampering T_j, $\hat{A}$ cannot realize the former outputs based on the former seeds since she cannot derive the previous seeds in the sequence from the current seed.

3.5.2 Cost analysis

Our authentication protocol involves one hash function, $h(.)$ and one pseudo random function $\mathcal{P}(\cdot)$. But the cost depends on the number of execution of the hash function during an authentication session between the reader and the tag. Therefore, we determine the cost of our protocol on the basis of the computations of $h(\cdot)$. From the authentication protocol described in section 3.3.2, the tag performs the $h(\cdot)$ twice, first in line 12 and second in line 15. So, the cost for our protocols is little higher than the protocols described in [Juels07, Tsudik06, and Weis03] which require the tag to perform only one hash function. The additional hash function allows our protocol to be serverless and yet avoids exposing the tag's secret to the reader. In terms of efficiency, in the worst case, the reader needs to perform $|\mathcal{L}_i|$ computations, since it has to derive $\mathcal{P}\left(seed_m \oplus \left(rand_i \parallel rand_j\right)\right)$ for each tag T_m in the contact list. Now, we consider the communication cost. Assuming that both reader and the tag ids are of the same length, the authentication protocol requires the communication of $(2 \cdot |n| + 2 \cdot |rand|)$ bits, where $|n|$ is the length of random numbers n_i and n_j, and $|rand|$ is the length of both the $rand_i$ and the $rand_j$ respectively.

3.6 Additional features

3.6.1 Ownership transfer

Ownership transfer ensures that an authorized reader renounces the authority of a tag and a new reader gets the authority to access the tag. Suppose, R_i is the current owner of the tag T_j. After transferring ownership to another reader R_x, T_j responds to R_x in the same way as it did to R_i. From now on, R_i has no rights to access T_j. Though ownership transfer issue is dealt with in

[Cui07, Tsudik07 and Molnar05], the backend server plays a significant role in those protocols. Based on our protocol, we propose two methods of ownership transfer.

A. *CA based ownership transfer:* The certification authority (CA) has all the responsibility of deploying tags and authorizing readers. A reader obtains its contact list $\mathcal{L}$ from the CA using a secure channel at the beginning of its operation. Whenever the reader R_i faces the need to transfer the ownership of a particular tag to another reader, it informs the CA about the change in the access policy along with the ownership information of the tag. Ownership information comprises the identifier and the current seed for the particular tag stored in the contact list of R_i. The CA now authenticates the new owner (another reader) and authorizes the new owner by updating its contact list with the ownership information. Then the certification authority also deletes the ownership information of the tag from the old owner's contact list. For example, the id_j and the current $seed_j$ of the tag T_j are its ownership information. The old owner R_i transmits this ownership information to the CA at the time of informing about a change in ownership of T_j. CA authorizes R_x (new owner) with this ownership information and removes this information from $\mathcal{L}_i$.

B. *Serverless ownership transfer:* The prerequisite of this method is *reader-to-reader secure communication*. At the time of the ownership transfer, the old owner R_i transmits the id_j and the current $seed_j$ of the tag T_j to the new owner R_x and then simply removes ownership information for T_j from its contact list $\mathcal{L}_i$. However, the old reader can abuse the situation by deciding not to delete $seed_j$ from his contact list. Therefore, to protect against such vulnerability, the new authorized owner R_x authenticates the tag T_j as soon as R_x receives the ownership information. This will desynchronize the seed shared between T_j and the old owner. Therefore, even if R_i does not remove ownership information for T_j, R_i will have no valid seed to access T_j thereafter. However, the shared seed between R_x and T_j will still be synchronized. Once the ownership transfer process is completed, the new owner R_x notifies the CA regarding the update in its new contact list to remain synchronized with the CA. This notification is done through a secure channel.

3.6.2 Scalability

Scalability means that a reader can find a tag's identifier with limited computational time regardless of the number of tags owned by it. According to our protocol, if the total number of tags owned by a reader is N, the time complexity of search operation is $O(N)$. Juels and Weis proved in [Juels07] that improved randomized hash lock offer strong privacy and security at the cost of poor scalability. We entirely comply with their observation and propose a more practical way of ensuring scalability with the help of ownership transfer. Our proposal is that each reader will have a threshold value Δ. Here Δ is the maximum number of tags that a reader can have access to and $N \leq \Delta$. When a reader's contact list surpasses the threshold Δ, the reader, called as a overloaded reader, wishes to reduce its burden. So, if the overloaded reader has a co-operative reader in its vicinity and if the co-operative reader has enough available memory space, the overloaded reader will transfer some of its burden to the co-operative reader.

3.7 Comparison with other protocols

In this section we compare our protocol with some renowned RFID authentication protocols. This comparison is based on the security and privacy properties required for an RFID system. The comparison is shown in Table 3.2.

3.8 Application scenarios

A. *Container recognition in off-site location*: Let us consider a case in which a company uses RFID system for employee identification, human authentication while entering into safety regions, document management, product maintenance and etc. All these services are easily ensured with central server based RFID system. But this company faces problem when they have to collect their ordered raw material containers from other companies that belong to the off-site

Table 3.2: Comparison of authentication protocols based on the security features and other additional features

Protocols	**Privacy Protection**	**Anti-Tracking**	**Anti-Cloning**	**Synchronization**	**DoS Resiliency**	**Forward Secrecy**	**Ownership Transfer**	**Scalability Assurance**
Seo-Lee-Kim [Seo06A]	Yes	Yes	Yes	Yes	Yes	Yes	External Intervention	Yes
Seo-Lee-Kim [Seo06B]	Yes	No	Yes	Yes	No	Yes	Yes	Yes
OSK [Ohkubo03]	Yes	Yes	Yes	Yes	No	Yes	No	No
YA-TRAP [Tsudik06]	Yes	Yes	Yes	No	No	No	No	Yes
Av-Oech[Avoine05]	Yes	Yes	Yes	Yes	No	Yes	No	Yes
RIPP-FS [Conti07]	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
Tan-Sheng-Li [Tan07]	Yes	Yes	Yes	Yes	Yes	Yes	No	No
Our Protocol	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**

locations. This off site location has no connection with the central server. Normally truck drivers are dispatched to the other companies to collect container deliveries. But it is a very usual case that people employed in this job does not have the capability to ensure that the supplied containers are the correct one that were ordered by his company. Moreover it is not possible to check each container individually because obviously there are enormous numbers of containers. As a result, containers being unchecked, sometimes wrong material are delivered to the warehouse. This causes a loss for the particular manufacturing company. Now this problem can be easily eliminated by using our serverless protocol provided that the containers are tagged objects. The truck driver may have with him his personal PDA, which can act as a reader. Reaching the offsite location this reader can easily authenticate the containers and find out whether they are the ordered containers or not. This can be easily done as under our protocol, the readers can authenticate and communicate with tags without the intervention of central server.

B. ***Environmental monitoring:*** The use of RFID systems in conjunction with highly miniaturized sensors will make it possible to observe diverse environmental phenomena. Environmental scientists perform diverse research on environment by attaching tags with animals and releasing them in the wild again. These attached tags together with our serverless protocol can help scientists on their research. Moreover, sometimes it becomes necessary to regain a tagged animal from the wild for research purpose. In this case our protocol can be very useful as readers can track or locate the tagged animal in the wild without the need of server.

C. *Authenticating smart objects usage at construction site:* Several research groups have been investigating applications of smart objects in outdoor working sites where regular tools are augmented for supplementary services. For example, in [Kortuem07] construction drill machines are augmented so that usage history can be monitored and usage safety can be ensured by appropriate alerts. Such augmentations have direct implications in the business and logistic processes of the companies since they use performance record of the workers. Our proposed protocol can be applied to such scenarios to authenticate the workers to use the smart tools and to enable secure logging of monitoring data locally which ensures their privacy.

3.9 Summary

In this chapter, we have suggested a serverless authentication protocol which ensures mutual authentication of both the tag and the reader. Our authentication protocol is forward secured and shielded against some major attacks, such as tracking, cloning, eavesdropping, physical tampering, and DoS. Moreover, we have also proposed ownership transfer mechanism which facilitates our protocol to be scalable. To the best of our knowledge, this is the first contribution in the literature that enables serverless protocol to perform ownership transfer. Our plan is to devise a robust authentication protocol which will be able to synchronize a tag and its legitimate reader even if the adversary desynchronizes them.

Chapter 4: A Robust RFID Private Authentication Protocol

4.1 Introduction

RFID systems have been scrutinized nowadays as one of the emerging technologies in pervasive environment. Authentication becomes indispensible in applications where security and privacy are major concerns. Besides preventing some major attacks, RFID systems need to be able to recover from unexpected conditions during operation.

Security and privacy aspects should be addressed before mass deployment of RFID tags in omnipresent environment. However, conventional security solutions cannot be integrated in RFID tags as they have inadequate computation capabilities with extremely limited resources. Consequently, research community proposed several authentication protocols [Ahamed08A, Conti07, Ohkubo03, Seo06A, and Tsudik06] that are secured against major attack models including tracking, cloning, eavesdropping etc. But the complete removal *denial of service* (DoS) attack is almost impossible. This attack causes conflict with the fundamental requirements such as *availability*. An adversary can launch this attack in several strategies against RFID systems. For instance, jamming the communication channels with noise can affect availability. However such attacks can be detected and mitigated by mechanisms at the physical layer [Sharma03]. In this chapter, we focus instead on the solutions to support availability at the protocol level (RFID application layer).

The adversary can attack a tag with numerous queries from a rogue reader that is his under control. As a result, the tag is not able to respond to any further query from the legitimate reader. In other words, a genuine reader cannot communicate with his legitimate tags. A similar attack is also possible on the reader launched by a rogue tag (controlled by the adversary). If the reader has to spend a lot of time to verify each tag’s reply, this attack can keep the reader busy all the time with some fake responses. If the adversary spends less time on generating those fake responses, then the system will eventually be flawed.

4.1.1 Desynchronizing attack

At the protocol level, another class of DoS attack can be possible. We term this attack as *desynchronizing* attack. To provide strong authentication in an RFID system, the reader (or the backend server) and the tag have to share secrets. For example, in a symmetric-key setting, both the reader and the tag may share keys and other state information. But if the protocol has to support privacy protection, the reader (or the backend server) must use some mutable information (such as a changing pseudonym) to identify the tag in the absence of fixed identification values. The dynamic information has to be synchronized between the reader and the tag to operate the system successfully. In this scenario, the adversary can launch the desynchronizing attack by breaking this synchronization. For example, due to the radio jamming of the channel between the tag and the reader, a communication failure may happen, which may eventually result in desynchronizing attack. Therefore RFID authentication protocols should at least figure out some methods to detect this attack and recover the system if it is under attack.

DoS attack is not addressed by most of the authentication protocols because it is not possible to cope with all kinds of DoS attacks. YA-TRAP [Tsudik06] is a famous authentication protocol that places little burden on the backend server and uses monotonically increasing timestamp which makes it secure against tracking but unsecure against DoS attack. Another a hash chain based RFID identification protocol is RIPP-FS [Conti07]. Here Conti et al. proposed that each tag shares a private symmetric key with the server. After each successful authentication, both the tag and the server update the symmetric key to maintain synchronization. RIPP-FS is resilient to a specific DoS attack where the adversary attempts to exhaust the hash chain. Another lightweight protocol is OSK [Ohkubo03]. Ohkubo et al. proposed that only two hash function is sufficient to provide indistinguishability and forward secrecy. But the problem of OSK is that a malicious reader may easily desynchronize a tag which results in DoS attack. In [Ahamed08A], we proposed two serverless authentication protocols. However, authentication protocol 2 (also presented in Chapter 3) is secured against almost all major attacks. But the major flaw of this protocol lacks recovery support.

In this chapter, we propose a ***R**obust RFID **P**rivate **A**uthent**i**cation P**r**otocol* (Ripair) that supports not only security and privacy, but also recovery in RFID systems. The protocol can get back the desynchronized tags and readers to their normal state and thus provides robustness. This protocol has been published in [Hoque09B].

4.2 System model

We consider an RFID system consisting of three components: tags, readers and a backend server. The tag is a wireless transponder embedded in physical objects for detection and prevention of product counterfeiting. The reader is a transceiver— they can query tags for identification of objects and/or subjects. To protect privacy all data of the tag that may be privacy sensitive is stored in the backend server. Conversely each tag contains a limited amount of data to prove his legitimacy to the backend server via the reader. The Reader not only interacts with the tag but also communicates with backend server to identify the tag. The communication channel between the reader and the backend server is assumed to be secured. For simplicity, we presume a reader and the backend server to be a single entity and refer it as the reader. An issuer, another entity of this model, initializes each tag by writing the necessary information into the tag's memory.

4.3 The robust RFID private authentication protocol (Ripair)

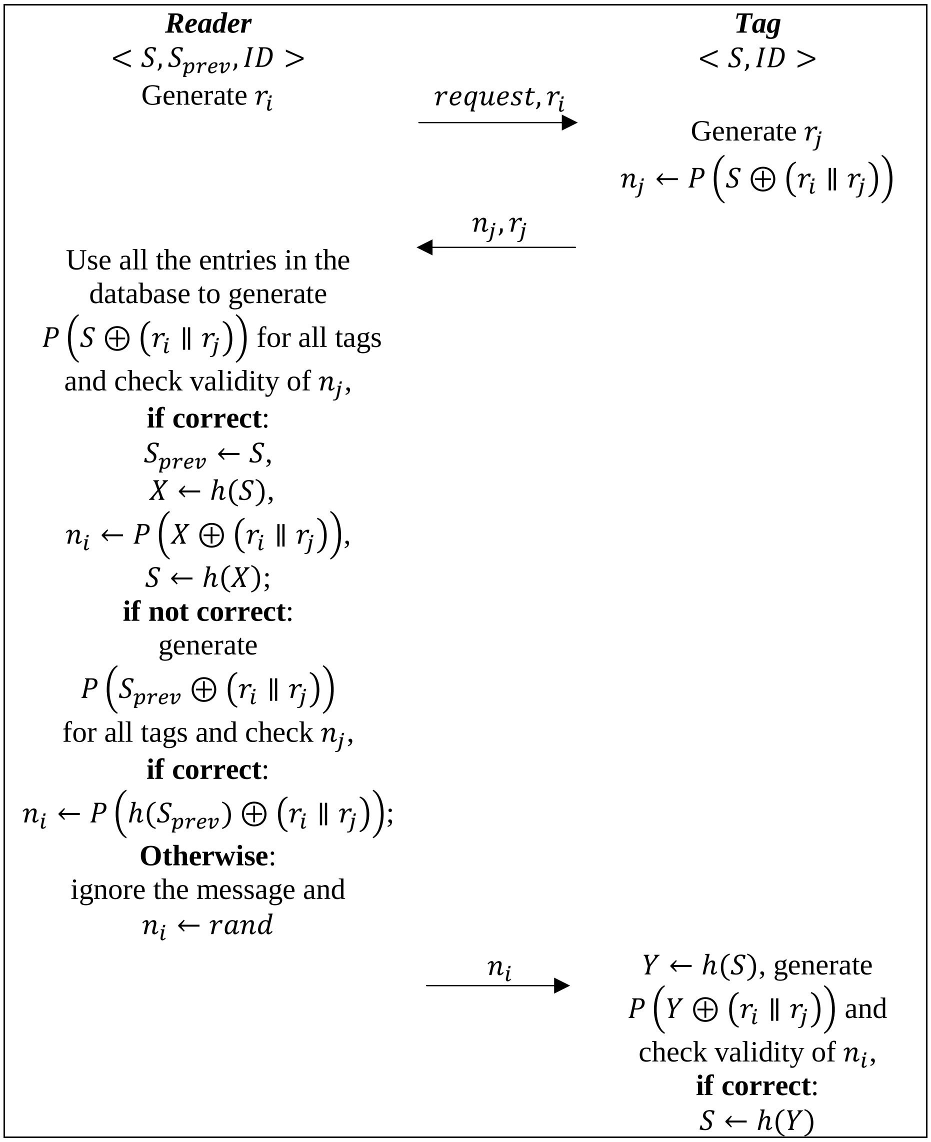
In this section we discuss our protocol. Before the description, we present the notations and the assumptions for this protocol.

4.3.1 Notations and assumptions

Suppose N is the total number of tags in the system. Each tag contains a secret pair consisting of a secret number S and an identifier ID. The tag gets this data from the issuer at the time of deployment. On the other hand, for each tag, the reader has a 3-tuple composed of the secret number S, the secret number of the last successful session S_{prev}, and the tag identifier ID. On the reader side, a tag ID points to all the data associated with the respective tag. All the entities of the system have the knowledge of a PRNG, $P(.)$ that can generate pseudorandom number based on its input (seed). In our case, this input (seed) will be the secret number S (or sometimes S_{prev}). Both the reader and the tag also knows about the one way hash function $h(.)$. Initially, the data of the tag are in sync with the data of the reader, and S_{prev} is equal to its corresponding S. We also assume that both the reader and the tag have capability of producing random numbers that have the properties of a nonce. In other words, the reader and the tag can produce nonce.

4.3.2 Protocol description

The protocol operates as shown in Figure 4.1. At first, the reader sends a request accompanied by a random number r_i (nonce). Upon receiving the request, the tag computes n_j with a self produced random number r_j (nonce). The tag replies with the n_j for authenticating itself. The r_j is attached with the response to help the reader to produce the same pseudorandom number. Now the reader checks the validity of n_j by computing $P\left(S \oplus \left(r_i \parallel r_j\right)\right)$ for each tag in the database. If the reader finds a match, it can be sure of the validity of the tag. To protect privacy, the reader has to mutate the secret number. Therefore the reader updates S_{prev} with the current S. To prove own legitimacy the reader has to generate his response that only the tag can understand. The reader produces n_i by using the next seed in the sequence that is the hashed secret number $h(S)$. Then the reader updates the seed with the next seed in the sequence (i.e., $h(h(S))$). If the reader fails to find any match in the first search strategy, he changes the scheme of search by replacing the S with the S_{prev} of each tag in the database. After a successful match, the reader has to generates n_i. In fact, this step of the protocol provides the system robustness to the desynchronizing attack. At this step the system recovers the tag that was desynchronized with the reader by some malicious actions happened before. In both the cases, the reader replies with the produced n_i. If n_j is not valid at all, the reader simply ignores the message and replies with a random number $rand$. However, this $rand$ keeps the protocol consistent by preventing an eavesdropper to acquire any knowledge (success or failure) about this session.



Reader

$< S, S_{prev}, ID >$

Generate r_i

$\xrightarrow{request, r_i}$

Tag

$< S, ID >$

Generate r_j

$n_j \leftarrow P\left(S \oplus (r_i \parallel r_j)\right)$

$\xleftarrow{n_j, r_j}$

Use all the entries in the database to generate $P\left(S \oplus (r_i \parallel r_j)\right)$ for all tags and check validity of n_j,

if correct:

$S_{prev} \leftarrow S$,

$X \leftarrow h(S)$,

$n_i \leftarrow P\left(X \oplus (r_i \parallel r_j)\right)$,

$S \leftarrow h(X)$;

if not correct:

generate

$P\left(S_{prev} \oplus (r_i \parallel r_j)\right)$

for all tags and check n_j,

if correct:

$n_i \leftarrow P\left(h(S_{prev}) \oplus (r_i \parallel r_j)\right)$;

Otherwise:

ignore the message and

$n_i \leftarrow rand$

$\xrightarrow{n_i}$

$Y \leftarrow h(S)$, generate $P\left(Y \oplus (r_i \parallel r_j)\right)$ and check validity of n_i,

if correct:

$S \leftarrow h(Y)$

Figure 4.1: The robust RFID private authentication protocol

Finally, it is the tag's turn to identify the reader by verifying the received n_i. If the n_i is valid, the tag updates his secret number accordingly as shown in Figure 4.1. Otherwise the tag discards the message.

4.4 Attack model

In this chapter, an adversary is denoted as $\hat{A}$. The adversary can control a number of readers and tags. Each reader and each tag controlled by the adversary is denoted as $\hat{R}$and $\hat{T}$, respectively. $\hat{R}$ is unauthorized to have access to any real tags as it is not connected with the backend server. Similarly, $\hat{T}$ is not valid as it has no idea about real S and real ID. We presume that the backend server cannot be compromised otherwise the adversary would get total control over the tag database then. We do not take for granted that the adversary has unlimited resources. Instead we assume that all the entities such as tags, readers, adversaries, adversarial tags and adversarial readers have polynomially bounded resources.

For this protocol, our assumptions also include that the goal of the adversary is not confined to only forging a real tag and making the fake tag (and therewith the fake product) indistinguishable from the real one. By hampering the availability (by applying desynchronizing attack) the adversary can flawed the system. The adversary $\hat{A}$ is simply more powerful than a passive attacker. Like a passive attacker he can eavesdrop on the both the forward and backward channels between a valid reader and a valid tag. However, like an active attacker, $\hat{A}$ can install a rogue reader $\hat{R}$ that can communicate with a valid tag. In addition, the adversary can install a fake tag $\hat{T}$ to communicate with a legitimate reader. In both cases the ultimate goal of the adversary is to counterfeit a tag with the learned information. Despite of these attacks, the adversary can block any channel at any time to fulfill his purpose. The adversary can launch the desynchronizing attack by blocking (or jamming) any of the channels at any step of the protocol or by scrambling any message passed from one party to another. In case of the protocol presented in Chapter 3, the adversary can successfully desynchronized the tag and the reader, if he blocks the forward channel (from the reader to the tag) at line 15 (see section 3.3.2 or Figure 3.1) where the reader sends a validation message to the tag. Denial-of-service attacks at the physical layer are out of scope of this chapter.

4.5 Analysis of the protocol

In this section we analyze our proposed protocol. First we present how our protocol provides robustness and recovers the desynchronized tags. Then we focus on security and privacy analysis of the protocol.

4.5.1 Robustness analysis

In this section, we present a detailed example to explain how Ripair provides robustness. After describing a successful tag query, we illustrate how a tag is recovered if the adversary launches the desynchronizing attack in the earlier session. In our example, to make analysis simple, we demonstrate the interaction between a single tag and the reader.

Reader	***Tag***
$< S_0, S_0, ID >$	$< S_0, ID >$

Figure 4.2: Initial states

Initially, the reader and the tag are both in sync, as shown in Figure 4.2. Now the tag responds to authenticate him upon receiving the request from the reader. Since the tag is valid, the reader finds a match with an entry in the tag database. Now the reader generates S_1 by using $h(S_0)$. To prove himself valid, the reader replies with a valid n_i. At the same time the reader also updates the S and the S_{prev} with the S_2 (=$h(S_1)$) and the S_0, respectively. The tag also finds the reader valid and updates his S from S_0 to S_2 according to the protocol. The subsequent states after this successful session are depicted in Figure 4.3.

Now the adversary can break the synchronicity in final communication pass of the protocol where the tag waits for n_i from the reader. Suppose the aforementioned tag again interacts with the reader after a while. Every message but the final one, for instance, is effectively received. The very last message containing n_i is damaged or lost due to some malicious actions of the adversary. Since the tag cannot update his secret number, he becomes desynchronized with the real reader. The internal states of the reader and the tag after this unsuccessful session are shown in Figure 4.4.

Reader	***Tag***
$< S_2, S_0, ID >$	$< S_2, ID >$

Figure 4.3: States after the successful interaction

Now if this tag again comes to vicinity of the reader, the tag starts interaction with the reader. However, the tag still has $< S_2, ID >$ as his internal data. Now the reader fails to find any match with the received response as he tries to validate with all the S's in the database where the S for this particular tag is S_4. On the contrary, the tag has used S_2 to generate his response. The reader further continues the search with the previous secret numbers, S_{prev}'s. Now the search will be fruitful since the S_{prev} for the particular tag is still S_2. Hence the reader concludes that the tag is valid that was desynchronized in some earlier session.

Reader	***Tag***
$< S_4, S_2, ID >$	$< S_2, ID >$

Figure 4.4: States after the unsuccessful session

To synchronize both the entities again, the reader takes a prominent step by sending the valid message without doing any update in the database. When the tag receives this message, he successfully verifies the originality of the reader and therewith updates the secret number (from S_2 to S_4) as well. Thus the robust protocol recovers the system from out of order state. After recovery, the states are shown in Figure 4.5.

Reader	***Tag***
$< S_4, S_2, ID >$	$< S_4, ID >$

Figure 4.5: State after recovery

4.5.2 Security and privacy analysis

In chapter 2, we have mentioned the six elements of safety ring which must be ensured by an authentication protocol in order to keep an RFID system secured and protected. In this section, we explain how Ripair defends the RFID system against those six major attacks and keeps the system within the safety ring.

Privacy Protection: Users carrying various tagged items do not want to hamper their privacy. If an adversary comes by any private information of the tag, by querying or eavesdropping, he may cause several vulnerabilities to the owner's day to day life. Our protocol protects user's privacy strongly. According to Ripair, a tag never sends his own ID to anyone, not even to the authorized reader. The tag sends his responses in disguise so that only an authorized reader can identify the tag.

Prevent Tracking: If a tag replies with a constant response (e.g., plain ID or even obfuscated message) each time he is queried, this constant response becomes a signature for the particular tag. So the potential problem is that the adversary can establish a link between the responses and the tag and therewith the owner of the tagged object which ultimately leads to tracking. In order to prevent clandestine physical tracking, each entity's response must be scrambled. Our protocol is secured against such kind of tracking attack. In Ripair, each entity replies with a distinct response in every session since random numbers (r_i and r_j) are involved within each computation of the validation messages (n_i and n_j).

Prevent Cloning: To launch this attack, an active adversary queries a real tag and obtains the response. By placing this response in a fake tag $\hat{T}$, the adversary $\hat{A}$ attempts to counterfeit the real tag. Now the attacker becomes successful in his attempts if he can deceive a legitimate reader. In other words, the real reader fails to distinguish the genuine tag from the fake one. According to Ripair, whenever an adversary $\hat{A}$ queries a real tag, he receives a distinct response each time because of the inclusion of random numbers in the response. Thus the protocol thwarts tag counterfeiting.

Ensure Forward Secrecy: Forward secrecy means that an adversary will not be able to realize any previous output transmitted by a entity even if he compromises the entity. Our protocol ensures forward secrecy. The secret number S, shared between the tag and the reader, is updated each time using irreversible one way hash function $h(.)$. After compromising a valid entity (the reader or the tag), $\hat{A}$ cannot realize the earlier responses by just learning the secret number. Because those responses are based on the former secret numbers and he cannot derive the former secret numbers from the current one.

Lessen Susceptibility to Replay Attack: In order to launch this attack, the adversary eavesdrops on both the communication channels between the tag and the reader. Thus $\hat{A}$ can learn all the challenges and the responses between the legitimate tag and the legitimate reader, and later uses these data to create a fake tag (reader) in order to deceive an honest reader (tag). But in order to deceive a legitimate reader (tag), the fake tag $\hat{T}$ (fake reader $\hat{R}$) has to generate a valid response. In case of our protocol, this is impossible since two distinct random numbers are involved in each session. Therefore Ripair is not susceptible to replay attack.

Deal with Denial of Service: In this attack, the adversary wants neither to derive any information nor to impersonate a tag or a reader. Rather the main target is to ensure that a valid reader cannot access his authorized tags. To launch a DoS attack, $\hat{A}$ can adopt several means. Though it is not possible to cope with denial of service due to all possible ways, we focus on some of those that our protocol can prevent. Jamming the channels may cause DoS. This problem exacerbates when the backend server and the tag shares a secret key that has to be synchronized after each regular query. Even distorted or damaged message may cause DoS. Certainly, our proposed protocol is vulnerable to above mentioned means. However, even after being desynchronized, the protocol can recover the RFID tags to the normal state.

4.6 Summary

Widespread deployment of RFID systems depend on the strength of security against major attacks, protection of private data, and recovery from unanticipated circumstances during the operations. Each of the elements of the “Safety Ring” has to be ensured to keep the system secured. In order to cope with these demands, we have presented a robust RFID private authentication protocol (Ripair) in this chapter. How this protocol recovers the system has been presented in the robustness analysis. In addition, security analysis establishes that the protocol keeps the system secured by ensuring the safety ring. Study of other issues of DoS and making the system more robust against those means are still open issues.

Chapter 5: ERAP: An ECC Based RFID Authentication Protocol

5.1 Introduction

Supply chain management can be referred to as a successful application of RFID technology. It is an enabling technology that has the potential of helping the retailers to provide the right product at the right place at the right time, thus maximizing sales and profit. RFID helps to uniquely identify each container, pallet, case and item being manufactured, shipped and sold. Thus RFID provides the building blocks for increased visibility throughout the supply chain.

Another important application of RFID systems is to detect counterfeit products. In supply chain, product authentication provides great opportunity to find illicit trade and business by identifying counterfeit products. Counterfeiting is a rapidly growing problem that affects a great number of industries and harms societies in many ways. Therefore new technologies must be put in place to prevent the counterfeit threat. And RFID technology has been identified as one of the major anti-counterfeiting technology.

Anti-counterfeiting problem can also be rephrased as authentication problem. In order to avoid counterfeiting, the adversary must not be able to clone any RFID tag. Moreover retrieving the tag’s secret information by attacking the authentication protocol between the reader and the tag has to be infeasible with respect to the resources of the adversary. Protection against cloning at the physical level is achieved by using physical countermeasures [Tuyls06] and protection against passive or active attack on the protocol level is provided by using cryptographic functions such as secure authentication protocols.

Public key cryptography (PKC) offers an attractive solution to the counterfeiting problem. But most of the previous works on RFID security consider only symmetric-key algorithms such as AES [Feldhofer04]. But it is still not clear whether public key algorithms can be implemented on small constrained devices such as RFID tags and can comply with memory, area, performance and power requirements for these applications. However in [Batina06A], the authors investigate PKC based identification protocols that are useful for anti counterfeiting applications.

In comparison with symmetric key based identification schemes, public key cryptography is more flexible requiring no complicated pre-distributed key and pair wise key sharing negotiation. The RSA algorithm is definitely the most popular in public key cryptography. *Elliptic curve cryptography* (ECC) is relatively a new family of public key algorithms that can provide the same level of security with shorter key lengths. Depending upon the environment and the application in which it is used, improved performance can be achieved.

It is a common belief in research community that public key cryptography (PKC), such as RSA and ECC, is not practical because the required computational complexity is prohibitive for the devices with limited computational capability and extremely constrained memory space. But some recent progress in ECC and RSA implementation shows that public key cryptography is feasible for small sensor devices and RFID tags. Recently a few papers [Tuyls06, Wolkerstorfer05, Hein09] discuss the feasibility of ECC based PKC on RFID tags. Here we adopt the belief that ECC based public key algorithms are feasible for RFID identification or authentication. In this chapter, we propose an ECC based RFID authentication protocol (ERAP) which is secure against some major passive and active attacks. Our proposed protocol is a mutual offline authentication protocol which ensures that the tag and the reader authenticate each other prior to any data exchange. Since it is a mutual authentication protocol, the tag releases own data to only an authenticated reader and the reader can access only those tags for which the reader is authorized. This protocol has been published in [Ahamed08B].

5.2 Technical preliminaries

Since our protocol is based on Elliptic Curve Cryptography (ECC), we first focus on some preliminaries of ECC.

5.2.1 ECC preliminaries

The foundation of ECC based encryption-decryption scheme and digital signature scheme is an *elliptic curve* (E). The *domain parameters* of an elliptic curve scheme describe an elliptic curve E defined over a finite field $\mathbb{F}_q$.

Definition 1. A set $D = (q, FR, S, a, b, G, n, h)$ of *domain parameters* consists of:

1. The field size q.
2. FR (field representation) is an indication of the representation used for the elements of $\mathbb{F}_q$.
3. If the elliptic curve is randomly generated in accordance with [ANSIX9.62], a s*eed* S is used. The seed length should be at least160 bits [Johnson01].
4. Two coefficients $a, b \in \mathbb{F}_q$ define the equation of E over $\mathbb{F}_q$.
5. A finite point $G = (x_G, y_G) \in E(\mathbb{F}_q)$ where $x_G, y_G \in \mathbb{F}_q$. G has prime order and is called *base point*.
6. The order n of the point G, with $n > 2^{160}$ and $n > \sqrt[4]{q}$.
7. The co-factor $h = \#E(F_q)/n$. ◊

Detail descriptions of domain parameters are provided in [ANSIX9.62, Johnson01 and Hankerson04]. There are security risks associated with multiple users sharing the same elliptic curve domain parameters [ANSIX9.62]. Detail security considerations are described in the standard X9.62 [ANSIX9.62].

Like any other public-key crypto, ECC is based on a key pair– a *private key* and a *public key*. The private key is statistically unique and unpredictable integer $d \in [1, n-1]$. And the corresponding public key is the scalar multiplication of d and G, i.e., $Q = dG$. Thus the key pair (Q, d) is associated with the domain parameters D of the elliptic curve.

5.2.2 System model

A model for our RFID authentication system has two direct components: tags T and readers R. Tags are wireless transponder embedded in physical objects for detection and prevention of product counterfeiting. Readers are transceivers— they can query tags for identification of objects and/or subjects. A Certification Authority CA is an indirect party that is trusted by all the tags and the readers. CA is also assumed not to be compromised. However, CA is not mentioned as a direct component of our RFID system since an offline authentication scheme requires no direct participation of a backend server. CA's mere function is to deploy all the tags as well as to authorize the readers. In other words, CA performs only at the time of the deployment of the system. That's why CA is not included as a direct component.

Since our protocol is based on ECC, each party (the CA, the tag and the reader) has to be capable of performing calculation based on ECC. The certification authority generates the elliptic curve domain parameters as well as the key pairs in accordance with X9.62 [ANSIX9.62]. To thwart the attacks of an adversary and to enhance the system security, we propose to select a unique elliptic curve for each RFID tag in the system.

Definition 2. A *Certification Authority* CA, the indispensable and indirect component of the RFID system, is equipped with four algorithms:

1. A *domain parameter generation algorithm* that generates a set D of domain parameters for each tag in the system. This algorithm randomly selects an elliptic curve over $\mathbb{F}_q$ according to X9.62.
2. A *domain parameter validation algorithm* that checks the validity of the set D before moving on to the next task.
3. A *key generation algorithm* that takes the set D as input and generates a key pair (Q, d), where d is the *private* key and Q is the corresponding *public* key. A key pair is generated for each elliptic curve, i.e., for each tag.

4. A *public key validation algorithm* that takes the set D and the associated public key Q as inputs and checks the validity of Q for the given set of D.◊

An RFID tag T is the smallest of all the components of our system. Each tag is capable of performing elliptic curve computation along with modular computation. During the deployment, each tag T receives a unique identifier ID, a unique set of domain parameters D and the associated private key d from CA (see Figure 5.1(a)). The certification authority writes all the data into the ROM (EEPROM) memory of the tag. These data are secret for each tag. We assume that a tag never reveals these secret data to any reader, not even to any other tag of the system. But the adversary is capable of learning these secret data by launching some hardware based physical attack. But in this chapter, we do not consider any hardware based physical attack.

The other direct component of our system is the RFID reader R. Since a reader can perform extensive computation than a tag, the reader has the major role in our authentication protocol. During the deployment, each reader R receives a *contact list* L from CA after R authenticates itself to CA (see Figure 5.1(b)). The contact list contains the identifying information of each tag T that R is authorized to access. We also assume that the communication between R and CA is performed via a secure channel.

Definition 3. If a reader R is to be authorized to access a set of tags $T_1, T_2, \dots, T_\gamma$, then after authenticating to CA the reader R receives a *contact list* L as follows:

$$L = \{(ID_i, Cert_i) | ID_i \text{ is the identifier of } T_i \text{ and}$$
$$Cert_i \text{ is the certificate of } T_i; 1 \leq i \leq \gamma\}$$

where, a *certificate* $Cert_i$ of the tag T_i is

$$Cert_i = \{(D_i, Q_i) | D_i \text{ is the set of domain parameters of } T_i;$$
$$Q_i \text{ is the public key of } T_i\} \text{ ◊}$$

ID_i	parameter D_i	d_i

(a)

ID_1	parameter D_1	Q_1
ID_2	parameter D_2	Q_2
	$\vdots$	
ID_γ	parameter D_γ	Q_γ

(b)

Figure 5.1: (a) Identifying and secret information of Tag T_i received from CA
(b) The contact list L of Reader R received from CA

In our system, a certificate of a tag has the same content like ECC based public key cryptography. In public key cryptography, the certificate of a party is available and easily accessible to other party that wants to communicate with the former. But in our system, the availability and the accessibility of a tag's certificate are restricted. Here the certificate of a tag is accessible only to the reader that has access to the tag.

5.2.3 Attack model

One of the goals of an adversary in any RFID system is to counterfeit a real tag such that the fake tag can only be distinguished from the real one with small probability. Evidently, this fake tag can let a fake product to be identified as a legitimate one just by attaching the fake tag to the fake product. In ERAP, an adversary is denoted as $\hat{A}$. The adversary can control a number of readers and tags. Each reader and tag controlled by the adversary are denoted as $\hat{R}$ and $\hat{T}$, respectively. $\hat{R}$ is unauthorized to have access to any real tags as it has no information of the contact list of the real reader R. Similarly, $\hat{T}$ is not valid as it does not have secret and identifying information of a valid tag. In addition, we assume that the adversary, the adversarial reader, and the adversarial tag have polynomially bounded resources.

We assume that $\hat{A}$ is more powerful than a passive attacker. In addition to eavesdropping (like a passive attacker) on the both channels between a valid reader and a valid tag, the adversary can install a rogue reader $\hat{R}$ (like an active attacker) that can communicate with the valid tag. Besides, the adversary can install a fake tag $\hat{T}$ to communicate with an legitimate reader. In both cases the ultimate goal of the adversary is to counterfeit a tag with the learned information. In addition, the adversary can launch physical attacks. However, the hardware based defenses against physical attacks are beyond the scope of this chapter.

5.3 Details of ERAP

A mutual authentication protocol enables communicating parties (a reader and a tag) to satisfy them mutually about each other's identity. A challenge-response based protocol offers both the parties (in our case, the reader and the tag) to generate a challenge for the other party to respond to. Each party proves own legitimacy by sending the accurate response to the other party. In an offline authentication protocol, the authentication is solely performed by a reader and a tag without any direct involvement of the backend server (like the certification authority). ERAP is a mutual challenge-response based offline authentication protocol. The protocol includes the following set of algorithms and definitions:

1. δ_R is a random number generated by the reader R. It is the challenge from the reader to the tag T.
2. δ_T is a random number generated by the tag T. It is the challenge from the tag to the reader.
3. $\delta_R, \delta_T < n$ [Hankerson04], where n is the order of the point G in ECC (see section 5.2.1).

4. **ProveTag**: $(\boldsymbol{D}, \boldsymbol{d}, \boldsymbol{\delta_R}) \rightarrow (\boldsymbol{r_T}, \boldsymbol{s_T})$. This algorithm is executed by the tag T. The arguments are the domain parameter D, the associated private key d and the challenge δ_R received from the reader R. The algorithm returns the tag's response as an ordered pair (r_T, s_T) that T transmits to R.
5. **VerifyTag**: $\left(\boldsymbol{D}, \boldsymbol{Q}, (\boldsymbol{r_T}, \boldsymbol{s_T})\right) \rightarrow \boldsymbol{\alpha}$. The reader R executes this algorithm after receiving the response (r_T, s_T) from the tag T. It takes the domain parameters D, the corresponding public key Q and the received response (r_T, s_T) as input. It verifies whether the response of the tag is accurate or not. This algorithm outputs α, where $\alpha \in \{Accept, Reject\}$.
6. **ProveReader**: $(\boldsymbol{D}, \boldsymbol{Q}, \boldsymbol{\delta_T}) \rightarrow (\boldsymbol{r_R}, \boldsymbol{s_R})$. This algorithm is executed by R to determine the reader's response (r_R, s_R) to the received challenge δ_T from the tag T, given that D is the set of domain parameters and Q is the corresponding public key.
7. **VerifyReader**: $(\boldsymbol{D}, \boldsymbol{d}, (\boldsymbol{r_R}, \boldsymbol{s_R})) \rightarrow \boldsymbol{\alpha}$. This is analogous to VerifyTag(.) algorithm except that it is executed by the tag upon receiving the response (r_R, s_R) from the reader. The output of this algorithm is α, where $\alpha \in \{Accept, Reject\}$.

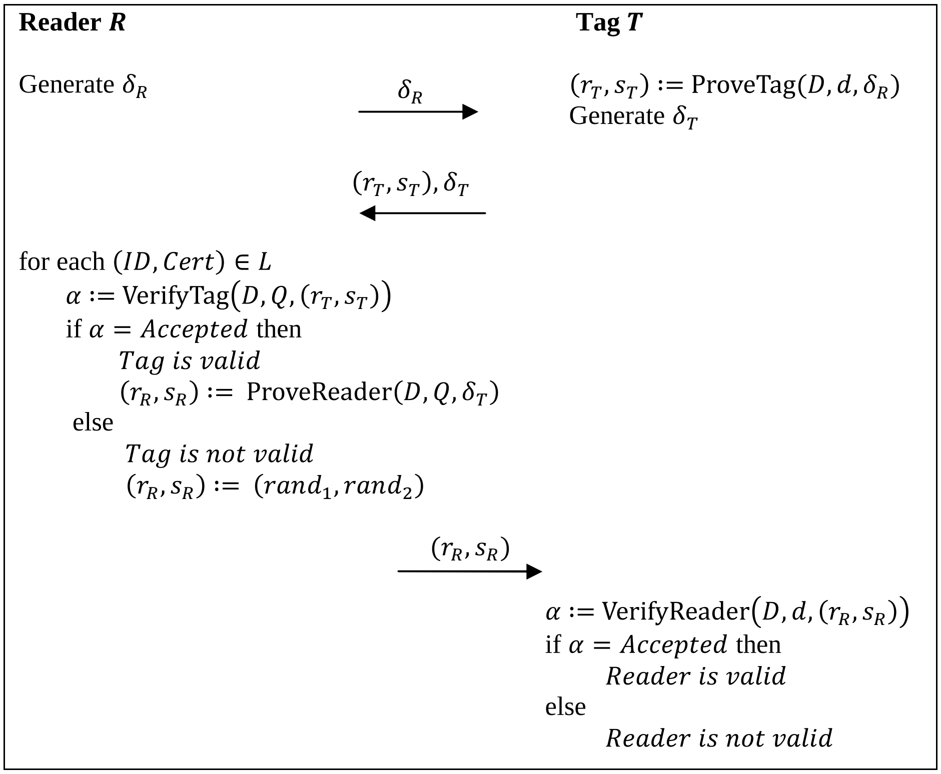
The complete authentication scheme is presented in Figure 5.2. Since it is a mutual authentication protocol, we will describe the protocol into two steps: one is "*the tag authenticates itself to the reader*" and the other is "*the reader authenticates itself to the tag*".

5.3.1 Step1:The tag authenticates itself to the reader

At the beginning of the protocol, the reader R generates a random number δ_R and transmits it as a challenge to the tag T. Now it is the time for the tag to prove its identity. The tag executes ProveTag(.) (see Figure 5.3) and replies to R with the response (r_T, s_T).

In step 1, the reader R plays the role of the verifier. The reader executes VerifyTag(.) (see Figure 5.4) to verify the received response from the tag. R accepts the tag T as valid if:

$\exists (ID, Cert) \in L$ such that
VerifyTag$(D, Q, (r_T, s_T))$ accepts the response.



```
for each $(ID, Cert) \in L$
    $\alpha :=$ VerifyTag$(D, Q, (r_T, s_T))$
    if $\alpha = Accepted$ then
        $Tag\ is\ valid$
        $(r_R, s_R) :=$ ProveReader$(D, Q, \delta_T)$
    else
        $Tag\ is\ not\ valid$
        $(r_R, s_R) := (rand_1, rand_2)$
```

```
$\alpha :=$ VerifyReader$(D, d, (r_R, s_R))$
if $\alpha = Accepted$ then
    $Reader\ is\ valid$
else
    $Reader\ is\ not\ valid$
```

Figure 5.2: Overview of the ERAP

Step 1 is based on *Digital Signature Algorithm* (DSA). In fact it is quite similar to *Elliptic Curve Digital Signature Algorithm* (ECDSA). A digital signature offers *data integrity* along with *authentication* and *non-repudiation*. But our concern is the authentication feature of a digital signature. In this way the prover (in step 1, the tag) can prove own identity to the verifier (in step 1, the reader). Here ProveTag(.) is the signature generation algorithm and VerifyTag(.) is the signature verification algorithm. Though ProveTag(.) and VerifyTag(.) are quite similar to the signature generation and verification algorithms of ECDSA, there are subtle dissimilarity. Instead of the message digest [Hankerson04], both the algorithms use the challenge generated by the reader. The proof that the verification algorithm works is given in section 5.3.3.

```
**Algorithm 1.** ProveTag
Input: Domain Parameters $D = (q, FR, S, a, b, G, n, h)$, private key $d$ and a challenge $\delta_R$
Output: Response $(r_T, s_T)$, an ordered pair based on $\delta_R$
1. Select a random integer $k \in [1, n-1]$.
2. Compute $P = kG$ and convert the field element $x_P$ to integer $\bar{x}_P$.
3. Compute $r_T = \bar{x}_P \bmod n$. If $r_T = 0$, goto step 1.
4. Compute $s_T = k^{-1}(\delta_R + dr_T) \bmod n$. If $s_T = 0$, goto step 1.
5. Return $(r_T, s_T)$.
```

Figure 5.3: Algorithm 1 (ProveTag)

```
**Algorithm 2.** VerifyTag
Input: Domain Parameters $D = (q, FR, S, a, b, G, n, h)$, public key $Q$ and received response $(r_T, s_T)$ based on $\delta_R$
Output: Acceptance or rejection of the response.
1. Verify that $r_T \in [1, n-1]$ and $s_T \in [1, n-1]$.
   If *any verification fails*, then
       Return (*Reject the response*).
2. Compute $w = s_T^{-1} \bmod n$.
3. Compute $u_1 = \delta_R w \bmod n$ and $u_2 = r_T w \bmod n$.
4. Compute $P' = u_1 G + u_2 Q$.
   If $P' = \infty$, then
       Return (*Reject the response*).
5. Convert $x_{P'}$ to integer $\bar{x}_{P'}$ and compute $v = \bar{x}_{P'} \bmod n$.
6. If $v = r_T$, then Return (*Accept the response*);
   Else Return (*Reject the response*).
```

Figure 5.4: Algorithm 2 (VerifyTag)

5.3.2 Step2: The reader authenticates itself to the tag

During step 1, in addition to the response (r_T, s_T), the tag T sends a new challenge δ_T for the reader R. Now it is the reader's turn to prove itself to the tag. If the reader finds the validity of the purported response of T, the reader produces a response (r_R, s_R) to be sent to the tag. Otherwise R generates two different random numbers as the response. To generate a valid response the reader executes $\mathrm{ProveReader}(.)$ (see Figure 5.5) using the certificate of the tag. The seed parameter S (see section 5.2.1) plays a significant role to generate the response. Its length makes the response hard to be forged.

Upon receiving the response from the reader, the tag verifies (r_R, s_R) by using $\mathrm{VerifyReader}(.)$ (see Figure 5.6) to ascertain that the other entity is the legitimate reader. With this step, ERAP terminates. In the step 2, the tag T does not have to search exhaustively. One execution of $\mathrm{VerifyReader}(.)$ is enough for T to figure out the validity of the reader.

5.3.3 Proof of the verification algorithms

Proof that $\mathbf{VerifyTag}(.)$ works. If the response (r_T, s_T) against the challenge δ_R is indeed generated by the legitimate tag, then we can get $s_T = k^{-1}(\delta_R + dr_T) \bmod n$. From the equation we obtain,

$$
\begin{aligned}
k \equiv s_T^{-1}(\delta_R + dr_T) &\equiv s_T^{-1}\delta_R + s_T^{-1}r_T d \\
&\equiv w\delta_R + wr_T d \\
&\equiv u_1 + u_2 d \qquad \pmod n
\end{aligned}
$$

Now in VerifyTag$(.)$, a point $P' = u_1 G + u_2 Q$ is generated. If the reader is authorized to access to the tag T, then the reader's contact list contains the valid G and the valid Q to generate P'. So

$$P' = u_1 G + u_2 Q = (u_1 + u_2 d)G = kG = P$$

and therefore $v = r_T$ as required.

```
**Algorithm 3.** ProveReader
Input: Domain Parameters $D = (q, FR, S, a, b, G, n, h)$, public key $Q$
and a challenge $\delta_T$
Output: Response $(r_R, s_R)$, an ordered pair based on $\delta_T$
1. Select a random integer $k' \in [1, n-1]$.
2. Compute $P = k'Q$ and convert the field element $x_P$ to integer $\bar{x}_P$.
3. Compute $r_R = \bar{x}_P \bmod n$. If $r_R = 0$, goto step 1.
4. Compute $s_R = {k'}^{-1}(\delta_T + Sr_R) \bmod n$. If $s_R = 0$, goto step 1.
5. Return $(r_R, s_R)$.
```

Figure 5.5: Algorithm 3 (ProveReader)

```
**Algorithm 4.** VerifyReader
Input: Domain Parameters $D = (q, FR, S, a, b, G, n, h)$, private key $d$
and received response $(r_R, s_R)$ based on $\delta_T$
Output: Acceptance or rejection of the response.
1. Verify that $r_R \in [1, n-1]$ and $s_R \in [1, n-1]$.
   If *any verification fails*, then
       Return (*Reject the response*).
2. Compute $w = s_R^{-1} \bmod n$.
3. Compute $u_1 = \delta_T w \bmod n$ and $u_2 = r_R w \bmod n$.
4. Compute $P' = (u_1 + u_2 S)dG$.
   If $P' = \infty$, then
       Return (*Reject the response*).
5. Convert $x_{P'}$ to integer $\bar{x}_{P'}$ and compute $v = \bar{x}_{P'} \bmod n$.
6. If $v = r_R$, then Return (*Accept the response*);
   Else Return (*Reject the response*).
```

Figure 5.6: Algorithm 4 (VerifyReader)

Proof that $\mathbf{VerifyReader}(.)$ works. If the response (r_R, s_R) against the challenge δ_T is certainly produced by the authorized reader, then $s_R = {k'}^{-1}(\delta_T + Sr_R) \bmod n$. From this equation we obtain,

$$\begin{aligned} k' \equiv s_R^{-1}(\delta_T + Sr_R) &\equiv s_R^{-1}\delta_T + s_R^{-1}r_R S \\ &\equiv w\delta_T + wr_R S \\ &\equiv u_1 + u_2 S \qquad (\bmod n) \end{aligned}$$

The tag T generates a point $P' = (u_1 + u_2 S)dG$ according to VerifyReader$(.)$. Since the tag has its own domain parameters and private key, we can say the point P' is

$$P' = (u_1 + u_2 S)dG = k'dG = k'Q = P$$

and as a result $v = r_R$, as required.

5.4 Security and privacy analysis of the protocol

In this section, we describe a number of attacks that an RFID system faces. This is followed by the counter measures an RFID system should take. Then, we explain how ERAP defends the system against these attacks. A legitimate reader and a legitimate tag are denoted by R and T, respectively.

Privacy protection. According to [Juels06], RFID gives rise to two major privacy concerns: covert tracking and inventorying.

A. Tracking. As RFID tags respond to any reader's challenges, tags can provide a ready vehicle for tracking by responding with a constant reply, for example, a fixed serial number. This privacy problem exacerbates when any personal information is combined with the tag's serial number. To prevent clandestine physical tracking, a tag's response must be scrambled. Moreover, the response must not carry any personal data.

The adversary $\hat{A}$ can launch an attack to track the tag T by controlling a rogue reader $\hat{R}$. This is an active attack initiated by $\hat{A}$ to track T. However, a passive attacker can even harvest enough information by eavesdropping so that the attacker is able to track T. If repeatedly challenging T with a same value yields a consistent reply, then $\hat{A}$ can distinguish the tag from other RFID tags since the consistent reply becomes a signature of T. Therefore $\hat{R}$ starts querying T with a fixed δ_R learned from any previous session of our protocol. But a random integer k used in $\text{ProveTag}(.)$ makes the responses of T random to $\hat{R}$, even though $\hat{R}$ reuses the same δ_R. Thus, $\text{ProveTag}(.)$ thwarts the physical tracking of a tag.

B. Inventorying. In some protocols, a tag's response contains unique serial number as well as the information of the product that the tag is attached to. People carrying such tags are subject to secret inventorying. A rogue reader comes to know about the consumer's personal information. To protect the consumer's interests, a protocol must ensure that the tag's response contains no product information. Moreover tags should reply in disguise so that only authorized readers can identify them.

ERAP prevents secret inventorying attack since ERAP does not exploit any intrinsic information, like the tag's ID. Rather the protocol uses domain parameters, private and public keys, and the challenges and responses between the tag and the reader. Thus, ERAP protects user privacy.

Cloning. Now we consider the cloning attack launch by an active attacker. To perform this attack, $\hat{A}$ first queries the tag T and obtains a response from the tag. By writing this response in a fake tag $\hat{T}$, the adversary attempts to counterfeit the real tag. $\hat{A}$ becomes successful in her attempts if she can deceive a legitimate reader R, i.e. R fails to distinguish $\hat{T}$ from T.

According to our protocol, whenever $\hat{A}$ challenges the tag T, she gets a different response each time due to the random integer k in $\text{ProveTag}(.)$. Now suppose $\hat{A}$ writes this response in $\hat{T}$. But $\hat{T}$ frustrates the adversary as it fails to fool the valid R because the adversary cannot predict the challenge δ_R that R will use to query the tag $\hat{T}$. Therefore $\hat{A}$ cannot obtain the valid response from the valid tag at the time of counterfeiting T. Hence we can say that ERAP is secured against the cloning attack.

Eavesdropping. So far we consider the active attacks that $\hat{A}$ can launch in our RFID system. Having the capability of a passive attacker, the adversary can eavesdrop on both the channels between the reader and the tag. Therefore, $\hat{A}$ learns the challenges and the responses between R and T, and later uses these learned data to launch any of the above mentioned attacks.

According to our protocol, $\hat{A}$ cannot track T because T replies with a different response each time it is queried, even if the same δ_R is reused. Even $\hat{A}$ fails to get any inventory information as ERAP does not reveal any personal and/or product related information. Moreover, ERAP prevents $\hat{A}$ from launching a cloning attack by eavesdropping. Suppose the adversary tries to impersonate the tag T with a fake tag $\hat{T}$ and she wants to fool an honest reader R with which T has communicated recently. To deceive R, $\hat{T}$ has to generate a valid response. But since each time R generates a new challenge δ_R' (not used before), so $\hat{T}$ fails to generate a valid response (r_T', s_T'). Therefore eavesdropping cannot help the adversary to attack the system.

Physical attack. Physical attack means $\hat{A}$ compromises either the reader R or the tag T. We consider each case. Our assumptions include that once $\hat{A}$ compromises R or T, she learns everything about the reader or the tag. However, here we do not address hardware based physical attack in this chapter.

A. $\boldsymbol{\hat{A}}$ compromises *R*. When $\hat{A}$ compromises the reader, she gains access to the contact list L of the reader. Therefore, the adversarial reader $\hat{R}$ can successfully impersonate R and communicate with the tags that the reader R has access to. Suppose L has the identifying information of the tag T and $\hat{A}$ wants to counterfeit T denoted as $\hat{T}$. The adversary succeeds in her attempt if $\hat{T}$ is able to fool another legitimate reader R_x that is also authorized to access T. But under our scheme, counterfeiting a tag and thereby deceiving a valid reader are possible if $\hat{A}$ succeeds to recover the private key d of the tag by knowing the tag's public key Q. This problem of recovering the private key with the knowledge of the public key is known as *Elliptic Curve*

Discrete Logarithm Problem (ECDLP) [ANSIX9.62]. Our adversary cannot be resolved the problem because of the polynomially bounded resources. Therefore $\hat{A}$ fails to counterfeit T.

B. $\boldsymbol{\hat{A}}$ compromises $\boldsymbol{T}$. Whenever $\hat{A}$ compromises T, she learns about the domain parameters D and the private key d of the tag. As a result, the adversarial tag $\hat{T}$ can effectively impersonate the tag T. Now with this information $\hat{A}$ wants to clone another valid tag T_x and with this cloned tag $\hat{A}$ wants to cheat an honest reader R that is authorized to access T_x. Under our protocol, knowing the D and d of the tag T does not help $\hat{A}$ to clone T_x. Since the tag T_x receives uniquely different domain parameters D and private key d from CA, $\hat{A}$ cannot recover the information of this tag T_x by compromising only T. As a result, $\hat{A}$ fails to create a fake tag to fool the reader R.

5.5 Summary

In this chapter, we have presented an RFID authentication protocol which is entirely based on Elliptic curve cryptography (ECC). It is a better choice than RSA because ECC can provide security similar to RSA, but with shorter key lengths. Our proposed ECC based authentication scheme can be used to solve product counterfeiting problem which has experienced a steep increase in recent years. This protocol is a mutual authentication protocol since it provides reader-to-tag and tag-to-reader authentication. According to security and privacy analysis of this protocol, we can conclude that it ensures security and protects privacy of the RFID system against common major attacks.

Chapter 6: An Efficient Anonymous Private Authentication Protocol

6.1 Introduction

RFID systems have been studied actively and frequently in pervasive computing environments for last few years. The inherent capability of precise and reliable identification attracts RFID systems in the area of tracking applications. This potentiality, however, can put individual privacy at a risk. A threat to consumer privacy is one of the major obstacles in the widespread deployment of RFID systems. A field trial of RFID embedded loyalty cards in Europe was cancelled due to consumer protest over privacy concerns [Caspian04]. Strong authentication can be a solution to the privacy problem. One party (*prover*) has to prove its own identity to another party (*verifier*) in such way that an adversary can neither identify nor track the party (prover). In this chapter we will consider only one way authentication where the tag has to authenticate itself to the backend server via the reader. Here the tag is the prover and the reader is the verifier. To address the privacy problem, the tag has to obfuscate its identity from eavesdroppers in such a way that only the valid reader can understand and identify the tag. Encrypting the tag’s message can protect its privacy. However, this technique cannot provide any hint to the reader about the key that the tag is using to encrypt its message. Therefore the reader has to search among a set of candidate keys until it finds the right key that correctly decrypts the tag’s message. As a result, the reader becomes inefficient in terms of identifying a single tag since it has to search a number of keys. This problem is exacerbated when the number of tags in the system increases.

Several private authentication schemes proposed in [Juels07, Ohkubo03, and Weis03] provide strong privacy at the cost of the search complexity on the reader’s side. Under these protocols, the workload of the reader increases linearly with the number of tags in the system. In other words, the search complexity is $O(N)$, where N is the total number of tags in the system.

These approaches become infeasible in some applications, such as tracking each product at every stage of supply chain management or automated display of flight information on smart tickets, where there is a huge of number of tags in the system. Molnar and Wagner [Molnar04] first proposed a tree based hash protocol for RFID systems to reduce the search complexity of the reader from $O(N)$ to $O(\log_\alpha N)$, where α is the branching factor at each level of the tree. The tag has to always perform $\log_\alpha N$ encryptions for every authentication. However, for authenticating a single tag, the worst case complexity of the reader is reduced to $\alpha \log_\alpha N$. But this approach achieves better scalability at the cost of some privacy loss of the tags [Nohl06]. Despite the privacy loss, this protocol has been held in great consideration by the RFID community because this is the first private authentication protocol that reduces the complexity of the reader. In fact, this is the only protocol so far that can be practically deployed in large scale applications. Therefore, improving the tradeoff between scalability and privacy of RFID systems has a great significance in reality. In [Buttyan06], the authors proposed a modified version of the tree based scheme where the branching factors are different at the different levels of the tree. This approach improves privacy protection. The authors also propose an algorithm to determine the optimal key tree for a given number of tags. Later Avoine et al. [Avoine07] proposed a group based private authentication scheme that improves the tradeoff between scalability and privacy by dividing the tags into a number of groups. The reader's complexity is cut down to γ, where γ is the number of groups in the system. In other words, the reader has to search through γ keys to find out the correct key. A benefit of this approach is that the tag has to perform only two encryptions for every authentication. In addition, this approach provides significant improvement in privacy protection. A serious limitation of this protocol is that whenever any tag is compromised (the group key and the tag's key become known to the adversary), all other tags of the same group lose their complete privacy. The level of privacy provided by the scheme decreases as more and more tags are compromised.

In this chapter, we propose a group based anonymous private authentication protocol (AnonPri) that provides higher level of privacy than the above mention group based scheme and achieves better efficiency than the approaches that prompt the reader to perform an exhaustive search.

6.2 Privacy in RFID Systems

So far several protocols have been proposed through which a tag obfuscates its identity while authenticating itself to an authorized reader, so that the tag can protect its privacy (and thereby consumer's privacy) from a rogue reader. To provide strong privacy, these protocols increase the complexity of the reader (some protocols increase the complexity of the tag, too). Thus a tradeoff arises between privacy and scalability of RFID systems.

6.2.1 Privacy vs. Scalability

Ensuring strong privacy imposes a higher complexity on the reader. Conversely, improving efficiency may hamper some privacy. In this chapter we focus on this major problem of between privacy and scalability of RFID systems.

Public key cryptography would be a better candidate to solve the problem between privacy and scalability. In this approach, the tag would encrypt its message using the public key of the reader so that only the real reader would be able to decrypt the message and identify the tag. But public key encryption is too expensive for low cost tags. In this chapter we consider the low cost tags which are capable of doing symmetric key encryption, in which keys are shared between the tag and the legitimate readers.

First, we outline how the tree based hash protocol provides scalability but sacrifice some privacy. Then, we describe how the group based protocol provides improved scalability as well as a higher level of privacy. Finally, we point out the privacy problem of this group based protocol.

Tree based hash protocol. The tree based hash protocol proposed by Molnar and Wagner [Molnar04] reduces the reader's complexity from $O(N)$ to $O(\log_\alpha N)$. Tags are organized in a secret key tree where each tag is assigned to a leaf of the tree. Secret keys are associated with each branch of the tree. Each tag (each leaf) receives all the secret keys along the path from the root to itself. If the tree has L levels, each tag stores L keys. The authors [Molnar04] proposed the key tree as a balanced tree. So if the branching factor is α, the $\log_\alpha N$ will be equal to L. Each tag has only one key that is not shared with any other tag of the system. Figure 6.1 shows a balanced key tree with $N = 8$ and $\alpha = 2$.

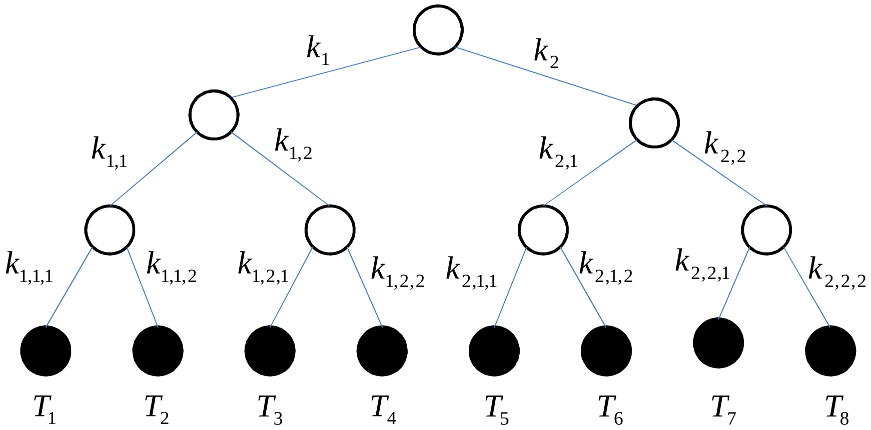


Figure 6.1: A secret key tree for the tree based hash protocol with $N = 8$ and $\alpha = 2$

According to this protocol, the reader queries a tag with a nonce n_r. Upon the reception of the nonce from the reader, the tag generates another nonce n_t and replies to the reader with

$$n_t, h\left(k_{l_1} \parallel n_r \parallel n_t\right), h\left(k_{l_1,l_2} \parallel n_r \parallel n_t\right), \dots, h\left(k_{l_1,l_2,\dots,l_L} \parallel n_r \parallel n_t\right),$$

where each $l_i \in \{1, \dots, \alpha\}$, $1 \leq i \leq L$, $h(.)$ is a hash function and $\parallel$ represents concatenation. The nonce produced by the tag provides unlinkability between two consecutive responses from the same tag. One the other side, the nonce from the reader prevents replay attacks. After receiving the response, the reader first finds a match with the first hash value of the response by hashing with all the keys of level 1. Whenever the reader obtains a match, the reader starts to search for the second hash value of the response by hashing with all the keys at the next level of the sub-tree rooted at the node where the reader has found the match. The reader repeats this step until it reaches a leaf. Thus, the reader's complexity is reduced to $O(\log_\alpha N)$. In the worst case, the reader has to search with all the α keys at each level of the tree and therefore, the complexity becomes $\alpha \log_\alpha N$.

The major drawback of this approach is the loss of privacy if any tag is compromised by the adversary. Since the tags share keys with some of the tags in the system, whenever a single tag becomes compromised all the tags that share at least one key with the compromised tag have to sacrifice their privacy. Suppose the tag T_3 in Figure 6.1 becomes compromised. All the tags of the system are partitioned into three disjoint sets. The adversary can now uniquely distinguish the tag T_4 and identify the tags T_1 and T_2 as a unique partition. All the remaining tags (T_5, T_6, T_7, T_8) form a single partition because the tag T_3 shares no key with them. Therefore each tag of this partition (T_5, T_6, T_7, T_8) is anonymous among these four tags. The privacy provided by this scheme diminishes as more and more tags are compromised by the adversary.

Group based protocol. Avoine et al. [Avoine07] proposed a group based authentication protocol to address the privacy problem of the tree based hash protocol. According to this protocol, tags are divided into γ disjoint groups of equal size. Each group is associated with a unique key that we refer to as a *group key*. Every tag shares this group key with other members of the given group. Additionally, each tag is assigned a unique key that is known only to the tag and the reader. Figure 6.2 shows the group organization of the tags where $N = 8$ and $\gamma = 4$. The k_i's are the group keys, where $1 \leq i \leq \gamma$. The identifier of the jth tag is represented by ID_j (not shown in Figure 6.2) and the unique secret key of the same tag is denoted as k_{T_j}, where $1 \leq j \leq 8$.

According to this protocol, the reader queries the tag with a nonce n_r. The tag, then, replies the following encrypted message (we assume that each tag has the knowledge of the encryption algorithm) with the nonce n_t produced by the tag.

$$E_{k_i}\left(n_r \parallel n_t \parallel ID_j\right) \parallel E_{k_{T_j}}(n_r \parallel n_t).$$

Now the reader tries all the group keys to decrypt the first portion of the message. If the reader finds the right key that correctly decrypts the message, then the reader can learn ID_j and decrypt the following portion of the response with the secret key of the tag T_j. Thus, the reader verifies the tag's legitimacy. This protocol reduces the complexity of both the reader and the tag. The tag always has to perform two encryptions. In the worst case, the reader has to perform $\gamma + 1$ encryptions. In addition, each tag needs to store only two keys for the authentication.

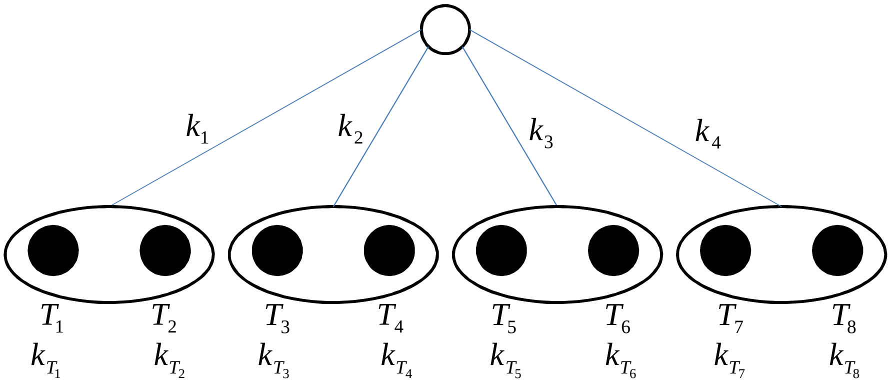


Figure 6.2: The group organization of the tags for the group based authentication protocol, with $N = 8$ and $\gamma = 4$

The group organization of this protocol improves the level of privacy. If any tag is compromised by the adversary, then this compromised tag affects only the other members of its group. After compromising the tag, the adversary learns the group key and the tag's secret key. Now the adversary can uniquely identify every single tag from the same group since the adversary can discover each tag's identifier by decrypting the first portion of the response from each tag with the learned group key. All the remaining tags that belong to different groups form a single partition so that the adversary cannot distinguish the tags that belong to this partition. For instance, if the tag T_3 is compromised, the adversary can uniquely identify only the tag T_4 (see Figure 6.2). The adversary cannot uniquely distinguish the other tags $T_1, T_2, T_5, T_6, T_7, T_8$. Each of these tags remains anonymous among these six tags. This is definitely a significant improvement in privacy protection of RFID systems in comparison with the other protocols, including the tree based hash protocol.

Like other protocols, this protocol also has some limitations. There is a tradeoff between the number of groups and the group size. To reduce the complexity of the reader, the number of groups has to be minimized. In this case, with larger group size, more tags will face the loss of privacy if any tag becomes compromised. On the other hand, to keep the loss of privacy to a minimum, the group size needs to be reduced, which eventually increases the reader's complexity. The loss of privacy increases for a system with large number of tags because to keep the reader's complexity moderate, tags have to be divided into groups of large size.

To address this problem, we propose an efficient anonymous private authentication (AnonPri) scheme that improves the privacy protection by keeping the reader's complexity moderate. In our approach, each tag is assigned a couple of identifiers. A single tag shares some of its identifiers with some members of its group. Thus this protocol prevents tracking by increasing the uncertainty of the adversary.

6.2.2 Privacy characterization

In literature several different notions of privacy have been proposed so far. Some authors mention *information privacy* as the privacy of RFID systems. This privacy notion is the act of preventing a tag from disclosing its product information [Weis03, Ohkubo03]. But protecting information privacy keeps tags traceable. Therefore it is a weak notion of RFID privacy. Some define *unlinkability* as the strong notion of RFID privacy [Nohl06, Chatmon06]. Unlinkability means the inability to distinguish between the responses from the same tag and the responses from different tags of the system. Providing unlinkability ensures strong privacy when the adversary cannot distinguish between two tags with a probability better than random guessing [Juels07]. In our protocol, we protect privacy of the tags by providing unlinkability between two tags of the system.

The level of privacy obtained by any protocol can be measured using the *anonymity set*. *Anonymity* has been proposed in the context of mix-nets in [Diaz02]. Mix-nets are used to make the sender (and the recipient) of a message anonymous. The anonymity set is defined as the set of all potential senders (recipients) of the message. Anonymity is defined as being not identifiable among a group of entities, i.e., the members of the anonymity set. A higher degree of anonymity is achieved with an anonymity set of larger size. Perfect anonymity is achieved if the anonymity set contains all the members capable of sending (receiving) messages in the system.

6.3 System model

Our protocol is based on the group based scheme. Therefore, tags are divided into groups of equal size. Suppose, N is the total number of tags in the system and τ is the number of groups. So, the group size is $n = \frac{N}{\tau}$. In this section, we define the components and parameters of our system.

Issuer. The issuer initializes each tag during the deployment by writing the tag's information into its memory. The issuer also authorizes the reader access to the tags. Even each group receives its unique group key and a pool of identifiers from the issuer.

Group. Each group has a n number of tags. The issuer assigns a unique group key k_{G_i} to the ith group G_i of the system. This key is shared between the members (tags) of this group. Each group also receives the following pool of identifiers from the issuer

$$\xi_i = \{ID_{i,1}, ID_{i,2}, \dots, ID_{i,M}\},$$

where, $1 \leq i \leq \tau$ and M is a system parameter. The pools of any two groups do not share any identifier, i.e., $\xi_i \cap \xi_j = \emptyset, \forall i \neq j$. Each tag of the group G_i is assigned a couple of identifiers from ξ_i by the issuer.

Tag. All the tags of the system are divided into τ groups. Each tag receives the shared group key of the group that the tag belongs to, a unique secret key that is known only to the reader and the tag itself, and a set of identifiers from the pool of identifiers of the group. Suppose, the tag T_j belongs to the group G_i. This tag possesses the group key k_{G_i}, the unique secret key k_{T_j}, and a set of identifiers Ω_{ij}. Each key is of θ bits, where θ is the security parameter of symmetric key encryption. We define the Ω_{ij} as follows

$$\Omega_{ij} = \{ID_{i,j_1}, ID_{i,j_2}, \dots, ID_{i,j_m}\},$$

where

- each ID_{i,j_x} is chosen randomly following uniform distribution from the pool ξ_i and $j_x \in \{1,2,\dots,M\}$, where $1 \leq x \leq m$
- $ID_{i,j_x} \neq ID_{i,j_y}$, for all $x \neq y$
- m is also a system parameter and $M > m$.

The identifiers are assigned to the tags in such a way that at least one identifier of a tag is shared with at least two other members of the same group. Therefore, we can say for the tag T_j,

$$\exists p, q\left[ID_{i,j_x} \in (\Omega_{ip} \cap \Omega_{iq})\right],$$

where p,q are any two members of G_i and $p \neq q$.

Reader. The reader is connected to the backend server. In this chapter, we assume the communication channel between the reader and the backend server is secured. From now on, we denote the backend server as the reader. In our system, the tag is the prover and the reader is the verifier. The reader receives all the secret information by the issuer during the deployment. The issuer issues the reader a set of secret information for each group in the system $\psi = \{\langle k_{G_i}, \sigma_i\rangle | 1 \leq i \leq \tau\}$, where k_{G_i} is the secret group key and σ_i is the mapping of the identifiers of the pool ξ_i with the secret keys of tags. Formally,

$$\sigma_i = \{\langle ID_{i,x}, \pi_x\rangle | 1 \leq x \leq M \text{ and } ID_{i,x} \in \xi_i\},$$

where π_x is the set of secret keys of tags associated with the $ID_{i,x}$. π_x can be defined as an empty set if no tag is associated with the $ID_{i,x}$ or it can be a set of size at least one. Formally,

$$\pi_x = \begin{cases} \{k_{\omega_1}, k_{\omega_2}, \dots\}, & \text{where } \omega_* \in \{T_1, T_2, \dots, T_N\} \\ \emptyset, & \text{otherwise} \end{cases}.$$

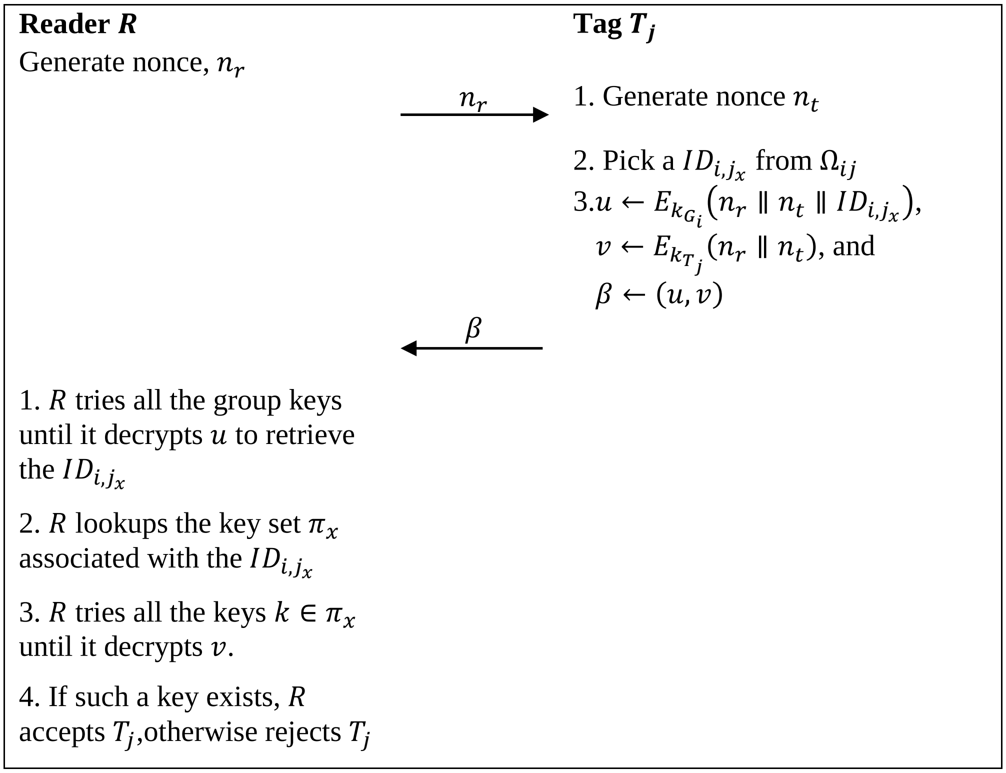
System parameters. Since each tag receives m identifiers randomly chosen from the pool of M identifiers, according to the ID distribution strategy, we can say that each tag has at least one identifier common with at least two group members. The probability that each tag shares at least one identifier with at least two group members is

$$P_{share} = 1 - \left(\frac{\binom{M-m}{m}}{\binom{M}{m}} \times \frac{\binom{M-2m}{m}}{\binom{M}{m}}\right) = 1 - \frac{\left((M-m)!\right)^3}{(M!)^2(M-3m)!}, \text{ where } M \geq nm.$$

For example, we consider an RFID system of 1000 tags divided in 10 groups. 100 tags are in each group. For simplicity, we assume M = 100 and m = 10. Then the probability that each tag shares at least one identifier with at least two group members is P_{share} = 96.87%.

6.4 Our protocol: AnonPri

In this section, we describe our protocol. The reader starts to query the tag with a nonce n_r. Upon the reception of the query, the tag generates another nonce n_t. Suppose the reader interrogates the tag T_j. In the second step, the tag picks an identifier, say ID_{i,j_x}, from Ω_{ij}. Then the tag computes β as shown in Figure 6.3. Here, $E_k(.)$ denotes symmetric key encryption with key k. The tag replies with the β. Now the reader searches all the group keys until it finds the correct one that properly decrypts the first part (u) of the response. If the reader retrieves the identifier ID_{i,j_x} that the tag used in its response, then the reader tries to decrypt the second part (v) of β with the potential set of secret keys (π_x) associated with ID_{i,j_x}. After finding the right secret key, the reader can uniquely identify the tag T_j. Sharing some identifiers of a tag with other members of the group provide unlinkability even if any tag is compromised by the adversary. We will discuss this in section 6.7.



Reader R

Generate nonce, n_r

$\xrightarrow{n_r}$

Tag T_j

1. Generate nonce n_t

2. Pick a ID_{i,j_x} from Ω_{ij}

3.$u \leftarrow E_{k_{G_i}}(n_r \parallel n_t \parallel ID_{i,j_x})$,
$v \leftarrow E_{k_{T_j}}(n_r \parallel n_t)$, and
$\beta \leftarrow (u, v)$

$\xleftarrow{\beta}$

1. R tries all the group keys until it decrypts u to retrieve the ID_{i,j_x}

2. R lookups the key set π_x associated with the ID_{i,j_x}

3. R tries all the keys $k \in \pi_x$ until it decrypts v.

4. If such a key exists, R accepts T_j,otherwise rejects T_j

Figure 6.3: The efficient anonymous private authentication protocol

6.5 Attack model

One of the major goals of an adversary in any RFID system is to infringe the tags' privacy by means of tracking. In this chapter, an adversary is denoted as $\hat{A}$. We assume $\hat{A}$ as an active adversary who has full control over all the communications between the tag and the reader. She can not only eavesdrop, but also intercept, modify and even initiate authentication session. The adversary can, for example, impersonate a tag and communicate with the valid reader. Even the adversary can query a valid tag and learn the tag's response. Our assumptions also include that the adversary can control a number of readers and tags. Each reader and tag controlled by the adversary are denoted as $\hat{R}$ and $\hat{T}$, respectively. $\hat{R}$ is unauthorized to have access to any real tags since $\hat{R}$ has no secret information like the real reader R. Similarly, $\hat{T}$ is not valid as it does not have the secret and identifying information of a valid tag. However, the adversarial reader $\hat{R}$ can communicate with a valid tag. Even the fake tag $\hat{T}$ can communicate with a legitimate reader. In both cases, the ultimate goal of the adversary is to track any tag of the RFID system. We assume that the adversary, the adversarial reader, and the adversarial tag have polynomially bounded resources. In addition, the adversary can launch physical attacks. However, the hardware based defenses against physical attacks are beyond the scope of this chapter. Our assumptions for this chapter also include that the reader cannot be compromised.

6.6 Privacy model

At the end of the protocol description, we mention that this protocol provide unlinkability and thereby preserves privacy. The adversary cannot link the responses with the tags, even if she can decrypt the first portion of the response and learn the identifier that the tags are using to produce the response. Like Juels and Weis [Juels07], we use an experiment based definitions to formalize RFID privacy. We conclude that the adversary cannot break unlinkability or invade privacy with probability better than random guessing. The following oracle-like construction exists:

$\mathcal{O}_{pick}$ is an oracle that randomly chooses some tags from all the N tags of the system.

$\mathcal{O}_{encrypt}$ takes a tag T as an input. Given the nonce n_r, the group key k_G, the secret key k_T and the set of identifiers Ω, the oracle randomly selects an $ID \in$ Ω, generates another nonce and finally produces the response $\beta = (u, v)$. It outputs the cipher text β.

$\mathcal{O}_{query}$ is an oracle that, provided with a tag T, queries the tag and outputs the received response β.

$\mathcal{O}_{flip}$ is an oracle that, provided with two tags T_0, T_1, randomly chooses $b \in \{0,1\}$ and queries the tag T_b using $\mathcal{O}_{query}$. Then it outputs the response β_b.

6.6.1 Information privacy against $\hat{A}$

Given a tag T, the set of identifiers Ω stored on T, and an identifier ID, an adversary can break the information privacy of our protocol if she can guess whether the tag T is using the ID.

Moreover, θ is the security parameter and $t \in \mathbb{N}$ is the maximum number of time the adversary can query the tag T. In addition, since the oracles of our privacy model are random, the inputs are computationally intractable from the outputs of the oracles.

Experiment $\mathrm{Exp}_{\hat{A}}^{priv}[\theta, t]$

1. **Setup:** The issuer initializes the N tags of the system with their corresponding unique secret keys, the group keys, and the sets of identifiers after dividing the tags into τ groups. It shares all the secret information with only the reader.
2. **Learning:** $\mathcal{O}_{pick}$ provides the adversary with a challenged tag T that the adversary queries t times and appends each response β to the list L (initially L is an empty list).
3. **Guess:** Now the adversary transmits the tag T to the oracle $\mathcal{O}_{encrypt}$ with a nonce and receives a response β from the oracle. The adversary selects an identifier ID. Given the list of t responses in L, $\hat{A}$ outputs 1 if she guesses that β is produced using ID, and 0 otherwise. $\hat{A}$ is successful if her guess is right.

Definition 1. *AnonPri is said to preserve information privacy with security parameter* θ *and* $poly(\theta)$ *representing any polynomial function of* θ*, if*

$$\forall \hat{A}, \Pr\left[\mathrm{Exp}_{\hat{A}}^{priv}[\theta, t] \text{ succeeds}\right] \leq \frac{1}{2} + \frac{1}{poly(\theta)}.$$

6.6.2 Unlinkability against $\hat{A}$

The adversary should not be able to distinguish between the two responses from the same tag.

Experiment $\mathrm{Exp}_{\hat{A}}^{unlink}[\theta, t]$

1. **Setup**: The issuer initializes the N tags of the system with their corresponding unique secret keys, the group keys, and the sets of identifiers after dividing the tags into τ groups. It shares all the secret information with only the reader.

2. **Learning:** $\mathcal{O}_{pick}$ provides the adversary with two challenged tags T_0, T_1 from the same group. The adversary queries each tag t times and appends each response β_0, β_1 to the list L (initially L is an empty list).

3. **Guess:** The adversary transmits T_0, T_1 to the oracle $\mathcal{O}_{flip}$. $\hat{A}$ receives the response β_b from $\mathcal{O}_{flip}$. Given the list of responses L and the response β_b, the adversary guesses the value of b. $\hat{A}$ succeeds if her guess is right.

Definition 2. *AnonPri is said to provide unlinkability with security parameter* θ *and* $poly(\theta)$ *representing any polynomial function of* θ, *if*

$$\forall \hat{A}, \Pr\left[\mathrm{Exp}_{\hat{A}}^{unlink}[\theta, t] \text{ succeeds}\right] \leq \frac{1}{2} + \frac{1}{poly(\theta)}.$$

6.7 Security and privacy analysis

In this section, we formally prove that our protocol preserves data privacy and provides unlinkability. In addition, we analyze the preservation of privacy in some attack scenarios where some of the tags of the system are compromised by the adversary $\hat{A}$.

6.7.1 Information privacy

Theorem 1. *AnonPri preserves information privacy with respect to the adversary* $\hat{A}$.

Proof. Let us assume $\mathcal{O}_{pick}$ provides the adversary $\hat{A}$ with a tag T. $\hat{A}$ transmits this tag to the oracle $\mathcal{O}_{encrypt}$ with a nonce n_1. Then $\mathcal{O}_{encrypt}$ provides $\hat{A}$ with the response β.

Now, $\hat{A}$ selects a ID. To break data privacy, $\hat{A}$ should tell if $\beta$is produced using the ID. This implies that $\hat{A}$ has to identify the input of the encryption by just learning the cipher text. $\hat{A}$ can succeed in two cases. First, if she can retrieve the inputs from the output of the random oracle. But this contradicts with our assumption that the inputs of a random oracle are computationally intractable from the output of the oracle. Second, if $\hat{A}$ knows the secret keys of the tag T. Without tampering the tag T, if $\hat{A}$ can determine the keys by learning the cipher texts, this again breaks the semantic security of the symmetric key cryptography. Therefore $\hat{A}$ can break data privacy with probability no better than random guessing. Thus it proves data privacy property of Definition 1. ■

6.7.2 Unlinkability

Theorem 2. *AnonPri provides unlinkability with respect to the adversary $\hat{A}$.*

Proof. Let us assume $\mathcal{O}_{pick}$ provides the adversary $\hat{A}$ with two tags T_0, T_1 from the same group. These two tags go into the learning phase. $\hat{A}$ transmits T_0, T_1 to $\mathcal{O}_{flip}$ which outputs the response β_b.

Now, to break unlinkability, the adversary $\hat{A}$ has to tell the value of b. We assume that the adversary's guess is right. In other words, the adversary can determine whether the response β_b is produced by T_0 or T_1, given the learned responses from both the tags. The responses of a tag cannot be a signature of the tag because according to our protocol, a nonce on the tag side makes each response different from all the previous responses originated from the same tag. Therefore, we can say that the guess is right because the adversary knows the keys (the group key and the secret key) stored on these two tags. Without tampering the tags T_0, T_1, the adversary has to determine the keys stored on these tags by just observing the cipher texts. But this contradicts with the semantic security of symmetric key cryptography. Therefore the adversary can break unlinkability with no better approach than random guessing. Thus it proves the unlinkability property of Definition 2. ■

6.7.3 Physical attack

Under this attack, we consider that the adversary $\hat{A}$ can compromise any tag with a probability of $\frac{1}{N}$. Whenever a tag T_j becomes compromised, the adversary learns all private information stored on the tag T_j. Therefore, the adversary can now decrypt u of each response β originated from the other members of the group G_i. Thus, $\hat{A}$ can learn the identifier that a tag is using to produce its response by decrypting the u. We discuss the aftereffect of this attack with an example and demonstrate how AnonPri provides unlinkability even if the adversary realizes the identifiers used in the responses.

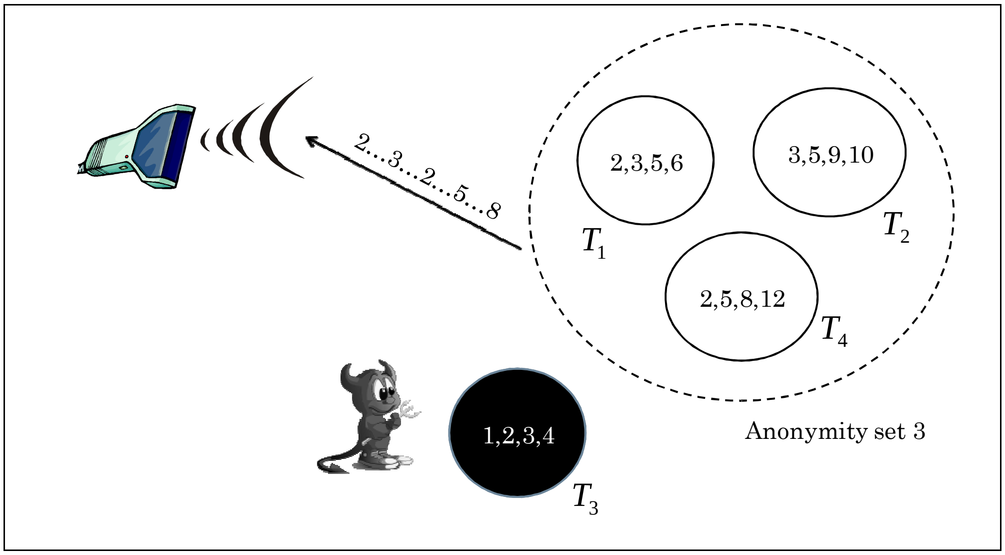


Figure 6.4: Aftereffect of a physical attack on AnonPri, where T_3 is compromised by the adversary

We consider a group G_i of four tags T_1, T_2, T_3, and T_4. Suppose the adversary compromised the tag T_3 as shown in Figure 6.4. Now the adversary learns the group key k_{G_i}, the tag secret key k_{T_3} and a set of identifiers $\Omega_3 = \{1,2,3,4\}$. From now on, the adversary can decrypt u part of all the responses originated from T_1, T_2, and T_4 with the group key k_{G_i}. But, the adversary still cannot decrypt v part of these responses since she does not possess the secret keys of these tags. With this learned information (k_{G_i} and Ω_3), the adversary tries to track the other tags of this group. Since the adversary can decrypt u of each responses, she can learn the identifier underlying the cipher text u. In other words, she can discover which identifier has been used to produce a response. The arrow in the Figure 6.4 represents that the responses of the authentication sessions (after T_3 is compromised) are transmitted from the tags (T_1, T_2, T_4) to the reader. The identifiers used in these responses are shown on above the arrow. Each identifier is shown in plaintext since the adversary can retrieve the identifier by decrypting u of β using k_{G_i}.

According to our protocol, even if the adversary comes to know about the identifier used in a response, she cannot conclude which of the potential tags is the sender of this response. In our example, the adversary discovers the identifier 2 is used two times, but she cannot be certain which of these tags (T_1, T_2, T_4) is the originator(s) of these responses. Though T_3 shares the identifier 2 with only T_1 and T_4, however, the adversary has no knowledge about the parties with whom T_3 is sharing which of its identifiers. Even the adversary does not know how many of the identifiers of Ω_3 are being shared. So, under this scenario, the anonymity set of the potential senders of a given response seems to be 3 to the adversary. Therefore, when the adversary compromises one tag from the group of n uncorrupted tags, AnonPri forms an anonymity set of size 1 and another anonymity set of size $(n-1)$ from the group instead of n anonymity sets of size 1 like the group based authentication [Avoine07]. This is the noticeable partition that improves the level of privacy provided by AnonPri. Because, the remaining $(N-n)$ tags of the system forms the other anonymity set which is same under both the protocols. Thus AnonPri prevents the adversary to gain any benefit for tracking by compromising a tag.

We now consider the case of compromising multiple tags of the same group. In the above scenario, even if $\hat{A}$ compromises either T_1 or T_4 after compromising T_3, the adversary cannot be certain whether T_2 has identifier 2 in Ω_2 or not. Therefore, the size of anonymity set is still 2, i.e., $n-c$, where c is the number of compromised tags of the group. If $\hat{A}$ compromises T_2 instead of T_1 or T_4, the size of anonymity set is still 2 (i.e., $n-c$). Therefore, we conclude that the anonymity set, formed from a group that is under physical attack, is of size $(n-c)$, where n is the group size and c is the number of compromised tags of the given group.

AnonPri provides protocol-level privacy only. In real world, there are many possible side channels. If tags emit distinct “radio-fingerprint”, then no protocol-level privacy countermeasures can prevent privacy infringement [Avoine05].

6.8 Measurement of privacy

In this section, we measure the level of privacy achieved by AnonPri as a function of the total number of compromised tags. We consider two privacy metrics for the measurement of privacy. First, our privacy measurement technique is based on anonymity set like the privacy metric used by Avoine et al. [Avoine07]. Second, we identify the amount of information disclosed by a scheme as another metric presented in [Nohl06]. This metric is based on Shannon's information theorem [Shannon48].

6.8.1 Measurement of privacy based on anonymity set

The level of privacy of an RFID system, achieved by a scheme, at a given time, is a function of the total number of compromised tags at that time. When some tags are compromised, the set of all tags are partitioned such that the adversary cannot distinguish the tags belong to the same partition, but she can distinguish the tags that belong to different partitions. Therefore, these partitions become the anonymity sets of their members. The level of privacy based on anonymity set, $\wp$, can be measured as the average anonymity set size [Avoine07].

$$\wp = \frac{1}{N}\sum_{i} |P_i| \frac{|P_i|}{N} = \frac{1}{N^2}\sum_{i} |P_i|^2$$

where $|P_i|$ denotes the size of partition P_i and $\frac{|P_i|}{|N|}$ is the probability that a randomly chosen tag belongs to partition P_i.

According to AnonPri, a similar kind of partitions is formed when tags become compromised. If c_i is the number of compromised tags within group G_i, then the set of the tags within this group is partitioned into c_i anonymity sets of size 1 and another anonymity set of size $(n - c_i)$. If $\mathbb{C} = \{c_i | c_i \text{ is the total compromised tags within } G_i\}$ is the set of compromised groups, $|\mathbb{C}|$ is the total number of compromised groups, and $C = \sum_{each\ c_i \in \mathbb{C}} c_i$ is the total number of compromised tags, the level of privacy $\wp$ achieved by AnonPri can be expressed as

$$\wp = \frac{1}{N^2}\left(\left(n(\tau - |\mathbb{C}|)\right)^2 + \sum_{\text{each } c_i \in \mathbb{C}} (c_i + (n - c_i)^2)\right)$$

where

- N = total number of tags in the system
- n = total number of tags within a group
- τ = total number of groups in the system.

6.8.2 Measurement of privacy based on information leakage

We measure the information leakage in bits based on Shannon's information theorem [Shannon48]. If we have a group of tags of size S and the adversary divides this group into two disjoint subgroups of size $S/2$, then 1 bit of information is disclosed out of $\log_2 S$ bits. Extending this concept from two subgroups of equal size to two subgroups of different sizes, where $\frac{S}{a}$ tags are in one subgroup and the remaining tags $\left(1 - \frac{1}{a}\right)S$ are in another subgroup, we can measure the average amount of information disclosed in bits as follows

$$I = \frac{1}{a}\log_2(a) + \frac{a-1}{a}\log_2\left(\frac{a}{a-1}\right).$$

In general, if the adversary splits N tags of the system into k disjoint partitions, then

$$I = \sum_{i=1}^{k} \frac{|P_i|}{N} \cdot \log_2\left(\frac{N}{|P_i|}\right)$$

where $|P_i|$ denotes the size of partition P_i.

According to our protocol, if $\mathbb{C} = \{c_i | c_i$ is the total compromised tags within $G_i\}$ is the set of compromised groups, $|\mathbb{C}|$ is the total number of compromised groups, and $C = \sum_{each\ c_i \in \mathbb{C}} c_i$ is the total number of compromised tags, the amount of information leakage in bits I can be expressed as

$$I = \left(\frac{n(\tau - |\mathbb{C}|)}{N}\log_2\left(\frac{N}{n(\tau - |\mathbb{C}|)}\right)\right) + \sum_{\text{each } c_i \in \mathbb{C}} \left(c_i\left(\frac{1}{N}\log_2 N\right) + \frac{(n - c_i)}{N}\log_2\left(\frac{N}{(n - c_i)}\right)\right)$$

where

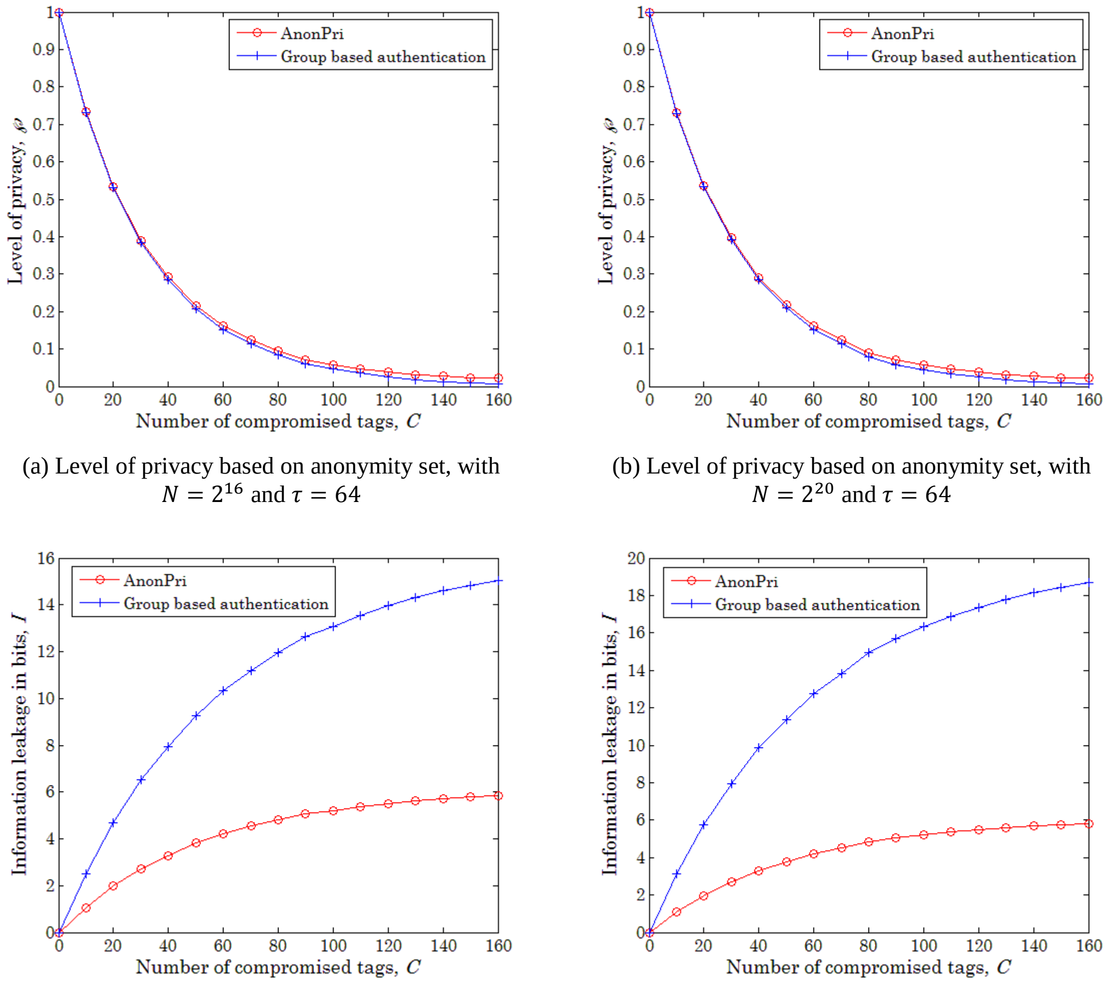
- N = total number of tags in the system
- n = total number of tags within a group
- τ = total number of groups in the system.

6.8.3 Experimental results

We have compared both the protocols, AnonPri and the group based authentication, using a Matlab simulation. The experiment results establish that the level of privacy provided by AnonPri is higher than that of the group based authentication. Our comparison is based on the two metrics presented above, the level of privacy (based on anonymity set) and information leakage. We have come up with a conclusion same as [Nohl06] that the information leakage describes the privacy threats better than the anonymity set.

In our simulation, we have considered two systems with $N = 2^{16}, \tau = 64$ and $N = 2^{20}, \tau = 64$. Tags are selected to be compromised with a uniform random distribution. The number of compromised tags ranges from 0 to 160. We have run the simulation for 100 times and computed the average $\wp$ achieved by AnonPri and the group based authentication as a function of the total number of compromised tags C (see Figure 6.5 (a)-(b)). The small increase in the level of privacy achieved by AnonPri is visible when the total number of compromised tags becomes more than 30.

During the simulation, we have also computed the average amount of information leakage I, for both the protocols, as a function of the total number of compromised tags C (see Figure 6.5 (c)-(d)). The plots depict that a significant amount of improvement in privacy protection is achieved by AnonPri. With the increase in the total number of compromised tags C, the average amount of information disclosed by the group based authentication is quite higher than the information disclosed by AnonPri. In Figure 6.5(c) ($N = 2^{16}$), when C becomes 160, the



(a) Level of privacy based on anonymity set, with $N = 2^{16}$ and $\tau = 64$

(b) Level of privacy based on anonymity set, with $N = 2^{20}$ and $\tau = 64$

(c) The amount of information leakage, with $N = 2^{16}$ and $\tau = 64$

(d) The amount of information leakage, with $N = 2^{20}$ and $\tau = 64$

Figure 6.5: Experimental results of AnonPri against the group based authentication

group based authentication discloses about 15 bits out of 16 bits of information, while AnonPri discloses about 6 bits of information. The group based authentication discloses 56.25% more information than AnonPri in a similar setup. Figure 6.5(d) ($N = 2^{20}$) shows that the group based authentication reveals almost 19 bits out of 20 bits of information and AnonPri reveals around 6 bits of information. This time the group based authentication discloses 65% more information than AnonPri. Based on the simulation results, we can conclude that the information disclosed by the group based authentication increases with the size of the system; however, AnonPri shows consistency in the information leakage in both the cases.

Information leakage is a better metric to demonstrate the privacy threats in RFID systems than anonymity set. Though the improvement in $\wp$ provided by AnonPri against the group based authentication is not significant, however, we can say that AnonPri provides better privacy protection than the group based authentication, based on the results of the amount of information disclosed by these two protocols.

6.9 Discussion

In this section, we discuss the limitations of AnonPri.

Search complexity. According to AnonPri, the reader's complexity is slightly increased than the group based scheme [Avoin07]. After receiving the response $\beta = (u, v)$ from a tag T_j, the reader searches for the correct group key to decrypt u. In the worst case, the reader has to perform this operation τ times. If such a group key exists, the reader can retrieve the identifier ID_{i,j_x} from u. Now, the reader has to search for the tag's secret key to identify T_j by decrypting v properly. The reader searches a key space of size $|\pi_x|$. Therefore, in the worst case, the reader's total complexity is $\tau + |\pi_x|$. In the best case, the size of π_x is 3 and in the worst case, it can be n, size of the group. But in the group based scheme, the reader's complexity in worst case is $\tau + 1$. Nevertheless, AnonPri is much better than the other schemes where the worst case reader's complexity is N, the number of total tags in the system. To provide improvement in privacy protection, we have to sacrifice this small increase in the complexity of the reader. Since readers are more powerful than tags, they can handle this increase in search complexity.

Memory complexity. According to AnonPri, tags need to store m number of identifiers along with the group key and the unique secret key. Though tags have limited resources, however, the increase in memory requirement is acceptable than the increase in computation and communication complexity. A smart RFID tags have memory capacity of 32kBytes or more [Laurie07]. Even RFID tags with extended memory capacity are available at the market [Fujitsu08]. All these tags can store the information required for AnonPri.

6.10 Summary

RFID systems will be welcomed for many applications if the system can guarantee consumer privacy as well as improve scalability. To address the tradeoff between privacy and scalability, we have proposed an efficient anonymous private authentication protocol (AnonPri) in this chapter. We have presented a brief comparison between the tree based hash protocol and the group based authentication for RFID systems. Then we have presented a privacy definition that an RFID system should consider. A detail security and privacy analysis of AnonPri establishes that AnonPri preserves information privacy as well as unlinkability. In addition, AnonPri provides higher level of privacy than the group based scheme when some of the tags are compromised by the adversary. However, according to AnonPri, the reader faces a slight increase in the search complexity, which is much better than performing linear search in the database to identify a single tag. Finally, we can say that AnonPri is suitable for many applications where privacy violation is a major point-of-failure.

Chapter 7: Conclusions and Future Work

In this chapter, we summarize the contribution of this thesis and present some future research directions.

7.1 Conclusions

In this thesis, we focus on private authentication protocols to protect privacy and ensure security of RFID systems. The research achievements of this thesis are as follows:

In chapter 2, we have identified six security and privacy requirements of an RFID system. We have also presented how these requirements form the safety ring. These requirements are the real need that an RFID system must achieved before deployment. If any of the six requirements is not fulfilled by an RFID system, then the system will be at a risk of privacy violation and/or security attacks. Later in this chapter, we have assessed some significant RFID authentication protocols based on symmetric key cryptography against the safety ring. Since some recent publications provide evidence about the feasibility of public key cryptography (e.g., Elliptic Curve Cryptography (ECC)) implementation on RFID tags, we have briefly discussed and made reference to those articles for interested readers.

In chapter 3, we have proposed a lightweight serverless authentication protocol for typical RFID tags that can perform simple symmetric key operations. This protocol makes use of pseudo random number generator (PRNG) and one way hash function to provide the security and privacy requirements of RFID systems. Since this protocol is serverless, the system is no longer vulnerable to the single point-of-failure. We have presented the security and privacy analysis of the protocol with respect to our proposed attack model. We have assessed the cost of this protocol in terms of storage, computation and communication. Two additional features of this protocol, ownership transfer and scalability, are presented in this chapter. We have explained how the current owner (a reader) of a tag can transfer the ownership information to a new owner (another reader). Finally, we have compared this protocol with some significant RFID authentication protocols based on the identified security and privacy requirements.

In chapter 4, we have defined the desynchronizing attack on RFID systems. We have proposed a robust private authentication protocol for RFID systems that not only supports recovery of the disabled tags but also protects privacy and prevents security attacks. We have, then, exemplified how this protocol recovers an RFID system that is under the desynchronizing attack. Later we have analyzed this protocol against our identified security and privacy requirements.

In chapter 5, we have presented that anti-counterfeiting is one of major needs for the widespread development of RFID systems. To address the counterfeiting problem, we have proposed authentication protocol which is entirely based on Elliptic curve cryptography (ECC). Though ECC is a class of public key cryptography, however, this protocol does not publicly share the public key of a tag. This protocol is a mutual authentication protocol as it provides reader-to-tag and tag-to-reader authentication. We have analyzed how this protocol is secure against common major attacks with respect to our proposed attack model.

In chapter 6, we have presented the tradeoff between scalability and privacy of RFID systems. We have explained two significant solutions to this problem – tree based hash protocols and group based authentication. We have, then, analyzed the limitations of these protocols. We have characterized the RFID privacy and proposed an efficient anonymous authentication protocol for RFID systems. A privacy model for RFID systems based on random oracles has been proposed in this chapter. Later we have analyzed and formally proved that our protocol protects the privacy of RFID tags. Finally we have described the reader's complexity and the tag's memory complexity of this protocol.

7.2 Future work

In this section, we present some future research directions in RFID security and privacy.

- Ensuring perfect privacy protection results in an RFID system with poor scalability. On the other hand, to achieve better scalability, an RFID system has to sacrifice some privacy. This tradeoff should be studied further for the development of better solutions that can ensure perfect privacy protection as well as better scalability.
- In this thesis, we consider the communication channel between a reader and the server is secure. However, this channel is also subject to security attacks. Ensuring secure communication over this channel can be a new research direction.
- In chapter 3, we have presented a technique for ownership transfer where reader to reader communication is required. We have considered that readers can communicate with other readers through a secure channel. Authentication over this channel can be fruitful topic.
- Some authentication protocols have recently been proposed based on public key cryptography. A further study on this area can be productive for some good solutions to RFID security and privacy problems.

BIBLIOGRAPHY

[Ahamed08A] Ahamed, S. I., Rahman, F., and Hoque, M., Kawsar, E., and Nakajima, T. (2008). *YA-SRAP: Yet another serverless RFID authentication protocol*. In Proceedings of the 4th IET International Conference on Intelligent Environment (IE08). Seattle, USA. pp. 1-8.

[Ahamed08B] Ahamed, S. I., Rahman, F., and Hoque, M. (2008). *ERAP: ECC based RFID authentication protocol*. In Proceedings of the 12th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS 2008), China. pp. 219-225.

[ANSIX9.62]ANSI X9.62. The Elliptic Curve Digital Signature Algorithm (ECDSA). http://www.ansi.org

[Avoine05] Avoine, G., and Oechslin, P. (2005). *A scalable and provably secure hash based RFID protocol*. In Proceedings of the IEEE International Workshop on Pervasive Computing and Communication Security (PerSec ’05), Hawaii, USA. pp. 110–114.

[Avoine07] Avoine, G., Buttyan, L., Holczer, T., and Vajda, I. (2007). *Group-based private authentication*. In Proceedings of the IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2007), Finland. pp. 1-6.

[Batina06A] Batina, L., Guajardo, J., Kerins, T., Mentens, N., Tuyls, P., and Verbauwhede, I. (2006). *Public key cryptography for RFID-tags*. In Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07), New York, USA. pp. 217-222.

[Batina06B] Batina, L., Guajardo, J., Kerins, T., Mentens, N., Tuyls, P., and Verbauwhede, I. (2006). *An elliptic curve processor suitable for RFID tags*. In Cryptology ePrint Archive, 2006/227. http://eprint.iacr.org/2006/227

[Bringer08] Bringer, J., Chabannel, H., and Icart, T. (2008). *Cryptanalysis of EC-RAC, a RFID Identification Protocol*. In Proceedings of the International Conference on Cryptology and Network Security - CANS’08, Hong Kong, China, LNCS, Springer-Verlag. pp. 149-161.

[Burmester06] Burmester, M., Le, T. v., and Medeiros, B. d. (2006). *Provably secure ubiquitous systems: Universally composable RFID authentication protocols*. In Proceedings of the IEEE Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm), Baltimore, Maryland, USA. pp. 1-9.

[Buttyan06] Buttyan, L., Holczer, T., and Vajda, I. (2006). *Optimal key-trees for tree-based private authentication*. In Privacy Enhancing Technologies Workshop – PET, Springer. pp. 332-350.

[Caspian04] CASPIAN Press Release. Metro's decision to drop the loyalty card, 2004. Last accessed June 2010 - http://www.spychips.com/metro/press-release-feb-27.html

[Chatmon06] Chatmon, C., Le, T. v., and Burmester, M. (2006). *Secure anonymous RFID authentication protocols*. Technical report, Florida State University, Department of Computer Science, Tallahassee, Florida, USA. http://www.cs.fsu.edu/~burmeste/TR-060112.pdf

[Conti07] Conti, M., Pietro, R., Mancini, L., and Spognardi, A. (2007). *RIPP-FS: an RFID identification, privacy preserving protocol with forward secrecy*. In Proceedings of the IEEE International Workshop on Pervasive Computing and Communication Security (PerSec '07), New York, USA. pp. 229-234.

[Cui07] Cui, Y., Kobara, K., Matsuura, K., and Imai, H. (2007). *Lightweight asymmetric privacy-preserving authentication protocols secure against active attack.* In Proceedings of the IEEE International Workshop on Pervasive Computing and Communication Security (PerSec '07), New York, USA. pp. 223-228.

[Deursen09] Deursen, T., and Radomirovi´c, S. (2009). *Untraceable RFID protocols are not trivially composable: Attacks on the revision of EC-RAC*. In Cryptology ePrint Archive: Report 2009/332. http://eprint.iacr.org/2009/332.pdf

[Diaz02] Diaz, C., Seys, S., Claessens, J., and Preneel, B. (2002). *Towards measuring anonymity*. In Privacy Enhancing Technologies Workshop – PET, CA, USA. pp. 54-68.

[Dimitriou05] Dimitriou, T. (2005). *A lightweight RFID protocol to protect against traceability and cloning attacks*. In Proceeding of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05). pp.59-66.

[Feldhofer04] Feldhofer, M., Dominikus, S., and Wolkerstorfer, J. (2004). *Strong authentication for RFID systems using the AES algorithm*. In M. Joye and J. -J. Quisquater, editors, Cryptographic Hardware and Embedded Systems – CHES'04, vol. 3156, LNCS, Springer. pp. 357-370.

[Fujitsu08] Fujitsu develops world's first 64KByte high-capacity FRAM RFID tag for aviation applications (2008). Last accessed June 2010. http://www.fujitsu.com/global/news/pr/archives/month/2008/20080109-01.html

[Garfinkel05] Garfinkel, S., Juels, A., and Pappu, R. (2005). *RFID privacy: An overview of problems and proposed solutions.* In IEEE Security and Privacy, vol. 3, no. 3. pp. 34-43. http://dx.doi.org/10.1109/MSP.2005.78

[Hankerson04] Hankerson, D., Menezes, A. and Vanstone, S. (2004). *Guide to Elliptic Curve Cryptography*. Springer-Verlag.

[Hein09] Hein, D., Wolkerstorfer, J., and Felber, N. (2009). *ECC is ready for RFID --- a proof in silicon*. In 15th International Workshop in Selected Areas in Cryptography, SAC 2008, New Brunswick, Canada. Revised Selected Papers. Springer-Verlag. pp. 401 – 413. http://dx.doi.org/10.1007/978-3-642-04159-4_26

[Hoque09A] Hoque, M., Rahman, F., Ahamed, S., and Park, J. (2009). *Enhancing privacy and security of RFID system with serverless authentication and search protocols in pervasive environments*. Springer Wireless Personal Communication, http://dx.doi.org/10.1007/s11277-009-9786-0

[Hoque09B] Hoque, M., Rahman, F., and Ahamed, S. (2009). *Supporting recovery, privacy and security in RFID systems using a robust authentication protocol*. In Proceedings of the 24th ACM Symposium on Applied Computing (ACMSAC '09), Honolulu, Hawaii. pp. 1062-1066. http://doi.acm.org/10.1145/1529282.1529514

[Johnson01] Johnson, D., Menezes, A., and Vanstone, S. (2001). *The elliptic curve digital signature algorithm (ECDSA)*. In International Journal of Information Security, vol 1. No 1, Springer. pp. 36-63. http://dx.doi.org/10.1007/s102070100002

[Juels03] Juels, A., Rivest, R., and Szydlo, M. (2003). *The blocker tag: Selective blocking of RFID tags for consumer privacy*. In Proceedings of the10th ACM Conference on Computer and Communications Security (CCS‘03), Washington DC, USA. pp. 103-111.

[Juels05] Juels, A., and Weis, S. (2005). *Authenticating pervasive devices with human protocols*. In Victor Shoup, editor, Advances in Cryptology – CRYPTO’05, volume 3126 of LNCS, California, USA. pp. 293-308.

[Juels06] Juels, A. (2006). *RFID security and privacy: a research survey*. In the Journal of Selected Areas in Communication (J-SAC), Vol 24 No 2. pp. 381-395.

[Juels07] Juels, A., and Weis, S. (2007). *Defining strong privacy for RFID*. In Proceedings of Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW ’07), New York, USA. pp. 342-347.

[Kerschbaum09] Kerschbaum, F., and Sorniotti, A. (2009). *RFID-based supply chain partner authentication and key agreement*. In Proceedings of the Second ACM Conference on Wireless Network Security (WiSec09), Zurich, Switzerland. pp. 41-50.

[Kortuem07] Kortuem, G., Davies, N., Efstratiou, C., Kinder, K., White, M., Hooper, R., Finney, J., Ball, L., Busby, J., and Alford, D. (2007). *Sensor networks or smart artifacts? An* *exploration of organizational issues of an industrial health and safety monitoring system*. In Proceedings of the 9th International Conference on Ubiquitous Computing (UbiComp 2007), Austria. pp. 465-482.

[Laurie07] Laurie, A. (2007). *Practical attacks against RFID*. In Network Security, 2007(9). pp. 4-7.

[Lee08A] Lee, Y., Batina, L., and Verbauwhede, I. (2008). *EC-RAC (ECDLP based randomized access control): provably secure RFID authentication protocol*. In Proceedings of IEEE International Conference on RFID, Nevada, USA. pp. 97–104.

[Lee08B] Lee, Y., Sakiyama, K., Batina, L., and Verbauwhede, I. (2008). *Elliptic curve based security processor for RFID*. In IEEE Transactions on Computer, vol. 57, No. 11. pp.1514-1527.

[Lee09] Lee, Y., Batina, L., and Verbauwhede, I. (2009). *Untraceable RFID authentication protocols: Revision of EC-RAC*. In Proceedings of IEEE International Conference on RFID, Florida, USA. pp. 178-185.

[Molnar04] Molnar, D., and Wagner, D. (2004). *Privacy and security in library RFID: Issues, practices, and architectures*. In Proceedings of the 11th ACM conference on Computer and Communications Security. Washington DC, USA, pp. 210-219.

[Molnar05] Molnar, D., Soppera, A., and Wagner, D. (2005). *A scalable, delegatable pseudonym protocol enabling owner-ship transfer of RFID tags*. In Proceedings of Selected Areas in Cryptography (SAC 2005), 3897, Springer-Verlag. Kingston, Canada. pp. 276-290.

[Nohl06] Nohl, K., and Evans, D. (2006). *Quantifying information leakage in tree-based hash protocols*. In Proceedings of the 8th International Conference on Information and Communications Security (ICICS), USA. pp. 228-237.

[Ohkubo03] Ohkubo, M., Suzuki, K., and Kinoshita, S. (2003). *Cryptographic approach to "Privacy-Friendly" tags*. In RFID Privacy Workshop, MIT, MA, USA.

[Okamoto92] Okamoto, T. (1992). *Probably secure and practical identification schemes and corresponding signature schemes*. In E.F. Brickell, editor, Advances in Cryptology-CRYPTO'92, volume 740 of LNCS, Springer-Verlag . pp. 31-53.

[Rieback05] Rieback, M., Crispo, B., and Tanenbaum, A. (2005). *RFID guardian: A battery-powered mobile device for RFID privacy management*. In Proceedings of the 10th Australasian Conference on Information Security and Privacy (ACISP05), LNCS, Vol. 3574, Springer-Verlag. pp. 184-194.

[Royal40] Royal air force. history: 1940. Last accessed June 2010 - http://www.raf.mod.uk/history_old/line1940.html

[Seo06A] Seo, Y., and Kim, K. (2006). *Scalable and untraceable authentication protocol for RFID.* In International Workshop on Security in Ubiquitous Computing Systems (Secubiq'06), LNCS, Seoul, Korea. pp. 252-261.

[Seo06B] Seo, Y., Lee, H., and Kim, K. (2006). *A Lightweight Authentication Protocol Based on Universal Re-encryption of RFID Tags*. (http://caislab.kaist.ac.kr/publication/paper_files/2006/CISC_1115_Youngjoon.pdf - Last accessed June 2010).

[Shannon48] Shannon, C.E. (1948). *A mathematical theory of communication*. In Bell System Technical Journal, vol. 27, pp. 379-423 and 623-656.

[Sharma03] Sharma, S., Weis, S., and Engels, D. (2003). RFID systems and security and privacy implications. In Proceedings of the Workshop on Cryptographic Hardware and Embedded Systems (CHES 2002), LNCS, Springer 2003. pp. 454-470.

[Tan07] Tan, C., Sheng, B., and Li, Q. (2007). *Severless search and authentication protocols for RFID.* In Proceedings of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '07), New York, USA. pp. 3-12 .

[Tsudik06] Tsudik, G., (2006). *YA-TRAP: Yet another trivial RFID authentication protocol*. In Proceedings of the 4th annual IEEE International Conference on Pervasive Computing and Communications Workshops, Italy. pp. 640-643.

[Tuyls06] Tuyls, P., and Batina, L. (2006). *RFID-tags for anti-counterfeiting*. In D. Poincheval, editor, Topics in Cryptology - CT-RSA 2006, vol. 3860, LNCS, Springer Verlag. pp. 115-131.

[Vajda03] Vajda, I. and Butty´an, L. (2003). *Lightweight authentication protocols for low-cost RFID tags*. In Second Workshop on Security in Ubiquitous Computing (Ubicomp '03). Seattle, WA, USA.

[Weis03] Weis, S., Sarma, S., Rivest, R., and Engels, D. (2003). *Security and privacy aspects of low-cost radio frequency identification systems*. In International Conference on Security in Pervasive Computing - SPC 2003, Springer-Verlag, Boppard, Germany. Vol. 2802. pp. 454-469.

[Wolkerstorfer05] Wolkerstorfer, J. (2005). *Scaling ECC hardware to a minimum*. In ECRYPT workshop – Cryptographic Advances in Secure Hardware – CRASH 2005. Invited Talk.

Appendix A

Glossary of terms

Term	Definition
RFID	Radio Frequency IDentification is the use of a wireless transponder embedded into objects for automatic identification and tracking of assets, animals and/or persons.
RFID tag	An RFID tag is an identification device which usually has an identifier and which transmits data wirelessly using radio frequency (RF) in response to interrogation by an RFID reader
RFID reader	An RFID reader is a transceiver that interrogates and read data from tags by broadcasting an RF signal using its antenna.
Backend server/ central server/ backend database	A backend server can retrieve the detailed information of a tag from its database by using the tag’s response as a key.
Security	Techniques that control who may use or modify the computer system or the information contained in it
Privacy	The notion of controlling where, when, to whom and what amount of information is provided to the external entities
Identification	In RFID systems, identification means the act of retrieving the identity of tags.
Authentication	Authentication means the act of confirming someone (or something) as authentic.
Mutual authentication	Mutual authentication is the act of proving one party’s identity to the other communicating party and vice versa.
Private authentication	Private authentication is the act of proving one party’s identity to the other communicating party using shared secrets (e.g., cryptographic keys, seeds of PRNG)
Symmetric key cryptography	A class of algorithms for cryptography that use shared secret cryptographic keys
Public key cryptography	A class of algorithms for cryptography that use a pair cryptographic key: public key (known to public) and private key (known only to the owner).