2009

Lightweight mutual authentication, owner transfer, and secure search protocols for RFID systems

Lars Skaar Kulseng
Iowa State University

Recommended Citation

Kulseng, Lars Skaar, "Lightweight mutual authentication, owner transfer, and secure search protocols for RFID systems" (2009). *Graduate Theses and Dissertations*. 11375.
https://lib.dr.iastate.edu/etd/11375

Lightweight mutual authentication, owner transfer, and secure search protocols for RFID systems

by

Lars Skaar Kulseng

A thesis submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Co-majors: Information Assurance;

Computer Engineering

Program of Study Committee:
Yong Guan, Major Professor
Daji Qiao
Steffen Schmidt
Wensheng Zhang

Iowa State University

Ames, Iowa

2009

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis. First and foremost, I would like to thank Dr. Yong Guan for believing in me and being patient with me throughout my undergraduate as well as graduate career. His guidance and support has been truly invaluable. I would also like to thank my committee members: Dr. Daji Qiao, Steffen Schmidt and Wensheng Zhang for their assistance. I would additionally like to thank Dr. Zhen Yu and Yawen Wei for their positive spirit and helpful contributions to my work.

ABSTRACT

RFID technology can potentially be applied almost everywhere. A typical RFID system involves a reader and a number of tags, which may range from the battery-powered ones with Wi-Fi capabilities, to the low-cost ones that are constrained in resources with even no internal power. Keeping RFID systems secure is important, because they are vulnerable to a number of malicious attacks. As for low-cost RFID systems, security problems become much more challenging, as many traditional security mechanisms are inefficient or even impossible due to resource constraints. Some existing solutions utilize traditional cryptographic primitives such as hash or encryption functions, which are often too expensive in hardware to be implemented on low-cost RFID tags. Furthermore, some other lightweight solutions have been reported to be broken, revealing their keys and ID numbers to the attackers.

In this thesis, we propose lightweight solutions to Mutual Authentication and Ownership Transfer for RFID systems. Mutual Authentication mitigates the issues of eavesdropping and cloning of tags. Only authenticated readers and tags will successfully communicate with each other. Furthermore, we adapt our Mutual Authentication scheme to secure the Ownership Transfer of RFID tags, which is a pertinent issue in the scope of RFID. When an item passes from one owner to another, it is undesirable for the old owner to be able to access the tag or read data from it. The new user must therefore update the access-granting information without revealing this to the old owner. Tag search is another important functionality that a RFID system should provide. In this thesis, we study how to secure tag search with a focus on low-cost RFID systems for which existing solution is not efficient.

These protocols are all realized by utilizing minimalistic cryptography such as Physically Unclonable Functions (PUF) and Linear Feedback Shift Registers (LFSR). PUFs and LFSRs are very efficient in hardware, and provide the low-cost RFID tags with unique characteristics that prevent a multitude of attacks. Compared to existing solutions built on top of hash functions that require 8000 - 10000 gates, our experimental results show that the schemes we propose demand only between 650 - 1400 gates for 64 bit variables and can be easily accommodated by the cheapest RFID tags with only 2000 gates available for security functions.

CHAPTER 1. OVERVIEW

1.1 Introduction

Computing is spreading beyond the office and home environment, to becoming a truly ubiquitous presence in our daily lives. Visions of the future now include so called ”smart homes” where your TV knows what you like to watch, your refrigerator knows when you’re out of milk, and your microwave cooks your food without your instructions. This pervasiveness is made possible by the increasing wireless capabilities of our possessions. One possible technology for enabling this type of future is Radio Frequency Identification (RFID).

RFID systems are composed of three main entities, a reader, a tag, and a database backend. These three pieces come together to form a system where any physical item can be tagged and scanned wirelessly. Tags come in a variety of sizes and feature sets. Some tags are smaller than the breadth of a human hair (54), while some are several centimeters wide and have extra capabilities such as Wi-Fi connectivity (55). The tags are tuned for different frequencies depending on the read range and signal propagation requirements.

The tags we are focusing on in this work are the low-cost tags that are referred to as *passive* tags. These tags have no internal power source, and rely on the electromagnetic signals from the reader for power of both internal circuits and communications. These low-cost tags can hold a maximum of 2000 hardware gates, as postulated by the EPC standard created by Auto-ID labs of MIT (3). Due to the hardware restrictions, most standard methods of encryption cannot be used since these methods are too hardware intensive, as can be seen in Fig. 2.1.

RFID tags are already being used in areas where security should be required. Mastercard and Visa both use RFID for the contactless payment features of their credit cards, which have been found to lack the appropriate levels of security (5). In the future we may see an expansion of the use of RFID in areas where the security of our private information is a risk. Leaving personal items open for scanning makes it easier for nefarious individuals to find out what items of value you are carrying, such as cash or car keys. Using this information, a criminal could select a target for robbery based on the data he is able to gather wirelessly and anonymously from the tags the victim is carrying. For RFID technology to become a success, it is vital that security measures are in place before the technology becomes widely deployed.

1.2 Background

Several attempts have been made at securing the wireless transmissions between RFID readers and tags. The vast majority of these solutions have either been found to be too weak or too expensive. The MAP family of protocols (12), M2AP (32), (33) were broken due to their strong reliance on bitwise operations. The solutions in (21)(22)(23) by Henrici and Ohkubo et al., rely on expensive cryptographic methods such as hash functions.

In our research, we have avoided expensive traditional methods, and reliance on bitwise operations alone. Instead we take advantage of certain circuits that are both low cost in terms of hardware and provide the basis for identification based on the inherent characteristics of each device. We use two types of circuits to accomplish this, Physically Unclonable Functions (PUF), and Linear Feedback Shift Registers (LFSR). Early attempts at providing solutions utilizing Physically Unclonable Functions, contrary to our solutions, require the back-end of the RFID system to be preloaded with a very large amount of data. With a large scale enterprise, the amount of storage required will become difficult to manage.

1.3 Our contribution

In light of the resource constraints placed on the RFID platform, we propose several lightweight security solutions that are relevant in RFID systems. These include:

- Mutual Authentication
- Ownership Transfer

- Search

Mutual Authentication mitigates the issues of eavesdropping and cloning of tags. Only authenticated readers and tags will successfully communicate with each other. Our Mutual Authentication scheme also lends itself to securing Ownership Transfer, which is a pertinent issue in the scope of RFID. When an item passes from one owner to another, it is undesirable for the old owner to be able to access the tag or read data from it. The new user must therefore update the access-granting information without revealing this to the old owner. We describe how our Mutual Authentication scheme can be adapted to secure the Ownership Transfer of RFID tags.

A secure search protocol mitigates the issues of eavesdropping and cloning of tags in the context of a reader searching for a particular tag. We propose both single-response and multi-response solutions where specialized circuits are used to create a fingerprint of the tag which in turn is used to determine if the tag is authentic or not.

1.4 Mutual Authentication

Without any security measures in place, a tag or reader might give away its information too easily. In a world where many different items are tagged, which may lead to loss of privacy, issuing tags that are suspicious becomes more important. In mutual authentication, the reader and tag does not reveal its identity unless some criteria has been met. Usually this type of authentication is proven through the knowledge of secret information shared between two entities. In addition to sharing a secret, an affordable way to encrypt the data must be found as well as appropriate measures must be taken to prevent other types of attack such as message replay.

1.5 Ownership Transfer

When several of our belongings are tagged with RFID technology, it becomes relevant to ask: What happens when a tag passes from one owner to another. The sale of items is very common, and is likely to continue being popular in the future. If a tagged item changes owner, then so should the access rights to the tag placed on the item. If this security is not provided, the privacy of the new owner could be compromised by the old owner. Worse still, the old owner could gain access privileges that the tag provides, which could lead to physical access to private areas such as a vehicle or home.

1.6 Search

A third feature of a RFID enabled world would be searchability of tagged items. A warehouse might be full of tags, and the owner may want to search for the presence of the tag. A person may have several tagged items in his home and may want to locate a particular item, like a key chain or a wallet. If such a feature lacks the appropriate security, the privacy of the user will be lost. An outside person should not be able to know what the user is searching for, nor if the user was able to find it or not. Although such a goal is difficult to achieve, since an adversary can observe whether the user found the item by physically spying on the user, the technology should at least make it more difficult than to simply record wireless messages.

1.7 Organization

The rest of this thesis will be structured as follows: Chapter 2 will contain background information on the various topics that are covered in this thesis. Chapter 3 contains a review of the literature related to the research that has been conducted. Next, Chapter 4 discusses the details of our Mutual Authentication solutions. Chapter 5 contains the details of our secure search solutions. Chapter 6 shows and discusses the results of our conducted experiments. Lastly, Chapter 7 contains a summary of our work, including concluding remarks and future work.

CHAPTER 2. BACKGROUND

This chapter contains background details on RFID, as well as descriptions of the other types of hardware used in our solutions.

2.1 Introduction

The research in this thesis combines several types of hardware. Most of our focus has been on the hardware implementations, since the size of the hardware is such an important feature of our solutions.

2.2 RFID

Computing is spreading beyond the office and home environment, to becoming a truly ubiquitous presence in our daily lives. Visions of the future now include so called ”smart homes” where your TV knows what you like to watch, your refrigerator knows when you're out of milk, and your microwave cooks your food without your instructions. This pervasiveness is made possible by the increasing wireless capabilities of our possessions.

One way adding such capabilities is to simply attach an RFID tag to an object. The main idea behind RFID is that you want to implement an electronic version of the bar code system, which can scan (or in this case ”read”) and record items wirelessly. RFID, standing for Radio Frequency Identification, is not a new technology, being invented around the time of World War II, yet it has only recently gained significant interest. This is mainly due to development of modern manufacturing methods and adjacent technologies such as the integrated circuit. RFID is becoming ever more ubiquitous however, especially in global shipping and retail services. Other areas where RFID could be implemented are: Passport verification, personal identification, health care services (if you are unconscious, your medical history can be scanned), and many more.

The danger that we face if this technology becomes as widespread as it has potential to become, is that there is a great security risk involved. Measures must be taken to ensure that all transactions are handled with the appropriate amount of security. It is vital that people's private information is protected from dubious actors. The need for privacy and security is evident in the overwhelming amount of viruses, spam, and other malicious activity on the Internet. Providing a secure environment is a difficult task, but if the technology is to be successful, security and privacy must be a crux of the design from the onset.

So far there has been little effort by manufacturers to develop and adopt secure solutions for RFID, likely due to companies wanting to take advantage of the technology as soon as possible. This was seen recently when major credit card companies released RFID enabled cards that can simply be placed next to a reader to make a transaction. These cards were found to be completely without encryption, and could quite easily be sniffed, leaving the attacker with credit card number, name, and other information. This is obviously an undesirable situation, and is just one example of how such a technology could be abused if not carefully considered beforehand.

RFID could provide us with great benefits, but could also give us many headaches if we are not careful in how we deploy the technology. Manufacturers, governments, and private individuals should know the pros and cons of this technology, and how we can improve it.

2.2.1 RFID Taxonomy

The implementation of RFID technology comes in several different flavors, and can be added to a wide range of products to provide identification services. What all RFID systems have in common is that there is always an RFID tag, and an RFID reader. Depending on the needs of the application, the tag can either be ”active” or ”passive”. Active tags are usually used on large items, such as cargo containers, or other items that need to be read from a distance. These tags usually operate at 455 MHz, 2.45 GHz, or 5.8 GHz, and they typically have a read range of 60 feet to 300 feet.

Passive RFID systems work in a different manner. Unlike active RFID tags, these tags do not use an independent power source. Instead, the tags are powered by the electromagnetic signal from the RFID reader. Passive tags utilize inductive coupling or propagation coupling depending on the read range that is required. The tag does not produce a signal by itself, but uses backscatter as a way to reflect the signal back to the reader with the information manifested as a modulation of the amplitude, phase or frequency of the signal. Passive tags can operate at low frequency (124 kHz, 125 kHz or 135 kHz), high frequency (13.56 MHz) and ultra-high frequency (860 MHz to 960 MHz). Some systems also use 2.45 GHz and other areas of the electromagnetic spectrum.

2.2.2 Why RFID is vulnerable?

Indeed, all wireless communications are vulnerable at some level due to the fact that the signal travels through the air. Without security features in place, anyone with the right equipment can tune in to those signals, perhaps even without the knowledge of the benign users of the system.

In the case of RFID, the danger lies in what can be seen as a paradox: the usefulness and pliability of the technology. The number of application possibilities is not at all restricted to tracking retail items through its global shipping traversal. On the contrary, the number of ways that RFID systems can be employed seems only limited by the imagination. Other deployment examples include:

Implanting tags in people

- Medical information in case of emergency
- Tracking and locating missing people in a disaster
- Tracking prisoners in a jail

Retail

- Quick automatic checkout

- Automatic payment in parking garages

Smart home interaction

- Food cooks automatically
- Lack of tags tells fridge to inform that stock is running low
- Keyless entry

Many other applications are possible, and as can be seen from the list above the privacy concerns of the individual, as well as financial information bears the risk of being compromised by attackers.

Another reason why RFID is vulnerable is that it is hard to protect. Passive RFID tags, which do not have a power source, will not be able to perform many traditional cryptographic functions such as RSA and SHA1. Table 2.1, taken from (4), shows a table of gate and power requirements for different operations that either provide cryptographic operations or are subsets of cryptographic functions. In our research, we have followed the convention of other researchers in the field (3), and assumed that the passive tags are capable of accommodating a maximum of 2000 gates.

The largest issue with most RFID implementations is that the systems are completely devoid of any security measures. A recent effort by K. Fu et al at the University of Massachusetts showed that the card holder's name and often credit card number and expiration are leaked in plain text to unauthenticated readers (5). They were also able to clone these cards, using

Algorithm	**Gate count**
SHA-256	10,868
SHA-1	8,120
MD5	8,400
MD4	7,350
AES	3,400

Table 2.1 Gate requirements of different cryptographic algorithms. Data taken from (4).

simple tools that amounted to $150, and even make a purchase over the Internet using this information! Considering that this technology could well be completely ubiquitous within a few years, and its wide applicability, it is clear that as much effort as possible must be put into securing these devices.

2.3 PUF

A Physically Unclonable Function (PUF), is a piece of hardware that produces a signature, either based on the unique characteristics of a particular instance alone, or in concert with a user defined input. Several different types of PUFs exist (17). In general, as long as you are able to produce a unique output determined by which device is producing the output, you have a PUF. Common to all solutions is that they rely on the variation of delays in wires and gates that exist in all electronic devices. Furthermore, despite efforts to reduce this normally unwelcome feature, delays seem to increase with newer technology as IC designs are becoming smaller (19). The reason why PUFs are so attractive in the security field is not only that they are cheap to implement, both monetarily and in hardware, they are also hard for an attacker to tamper with. If the attacker tries to evaluate the PUF or IC e.g. using probes to measure wire delays, the characteristics of that particular PUF will be changed (perhaps forever), and will therefore not give the information that the attacker wants.

The design of the PUF theoretically provides a 50% chance of a '0' or a '1' in each bit position. The entropy calculation in 2.1 (based on Bernoulli trials) shows that the output has an entropy value of N bits, which would allow for the 50/50 chance.

$$\begin{aligned} H(X_i) &= -plog_2(p) - (1-p)log(1-p) \\ H &= N[-plog_2(p) - (1-p)log(1-p)] \\ H &= Nlog_2(2) \\ H &= N \end{aligned} \tag{2.1}$$

2.3.1 PUF Taxonomy

As mentioned, there are many ways to achieve what a PUF sets out to accomplish. In the context of RFID, the silicon based PUFs seem to be the most practical. Other forms of PUF include:

- Optical PUFs - A laser beam is shot at a material which scatters the light, creating a unique pattern which gives rise to a value unique to that instance of the material (6).
- Coating PUFs - An IC can be surrounded by wires, which are separated by a dielectric, creating a capacitance between the wires (7). This capacitance will change if an attacker tries to evaluate the PUF.
- SRAM PUFs - The initial state of SRAM is known to be intrinsically random, and is therefore well suited as a basis for a PUF (8).
- Silicon PUFs (SPUF) - PUFs based on the wire delays associated with signal propagation in ICs (10), (11), (17).

The PUF variant that we will be focusing on is the silicon based PUF, or SPUF. SPUFs are delay circuits that take advantage of the fact that no two circuits have exactly the same delay properties, even if they were produced on the same wafer. Within the SPUF category, there are several ways to produce a circuit that takes advantage of the variation in the wires and other components. Some examples include:

- **Multiplexor PUF**: A series of multiplexors switched by input bits are connected as shown in Fig. 2.1. The delay in each wire and multiplexor is unique to that particular device and will create a race between two signals traversing the length of the PUF at different speeds. An arbiter in the form of a flip-flop is placed at the end to determine which signal arrived first, and a ’1’ or ’0’ is produced based on this result. Each of these PUF sections are then replicated equal the amount of output bits that the application requires.

- **Ring Oscillator PUF**: This type of PUF takes advantage of inherent wire delays by creating a simple oscillator that will produce a different frequency depending on what device it is placed in. There are several ways in which an RO PUF can produce output based on input. In one example, the frequency of each oscillator is tallied by a counter, and the frequencies of two oscillators can be compared by looking at counter values. An output of '1' or '0' is then given based on which oscillator is faster. Two such comparisons are then multiplexed by the input from the user to determine the final output bit. Again, this structure is repeated for every output bit required by the application.

- **Carry-Chain PUF**: A carry-chain is created when a binary addition operation is performed between two numbers that create an overflow. For example, the addition of the binary values 1111 and 1, would create a carry-chain which eventually results in an overflow, leaving the value 10000, where the left-most bit is the overflow bit. The carry operation is something that takes the circuit some time to complete, and may differ from one device to another, again due to wire delays. Two such carry-chains of a certain length can be set to race for whoever can set the overflow bit first. The winner of the race is determined by a flip-flop, and again a '1' or '0' is produced based on which chain won the race. Two such races are then multiplexed in a similar way as above to produce an input driven output.

- **Butterfly PUF**: This kind of PUF also takes advantage of varying wire delays, but in a different way than the race based solutions previously mentioned. A Butterfly PUF (BPUF) (1) attaches two latches that are wired to create an unstable state in the circuit. The *clock* signal is always set to 'high'. The *preset* value of the first latch and the *clear* value of the second latch are always set to 'low'. The other two *preset* and *clear* pins are connected to an input called *excite*, which at first is set to 'high', and after some time drops to 'low'. The inputs of the latches are cross-coupled with each other. The result of this circuit is that when *excite* is set to 'high' the two latches are unstable and produce inconclusive output. Once the excite signal drops to 'low' however, the latches "negotiate" a stable state to either a '1' or a '0' depending on the wire delays in the circuit. Two BPUFs can also be multiplexed to let the user have control of what bits are produced in the final result.

For our solutions we will use the notation shown in (2.2).

$$P(x) = y, \tag{2.2}$$

where P is the PUF function, x is the L-bit input that switches the multiplexors, and y is the L-bit output of the PUF. For L-bits of output, L replications of the above described circuit must be in place.

There are several advantages to using a PUF circuit in an RFID setting.

- A 64-bit PUF can be implemented using only 545 gates (16).
- Two PUFs are highly unlikely to produce the exact same output given the same input (17), i.e. we can distinguish two tags by the responses they give to a certain input.
- The output is not produced by mathematical function, but by delay variation so it cannot be calculated or predicted.

In our solution, PUFs are used to verify the identity of tags. Given a certain input the tag's PUF will produce a certain output, while other tags' PUFs will produce different output. We use this challenge-response pair to verify the tag's identity.

2.4 LFSR

A Linear Feedback Shift Register (LFSR) (42) is a simple circuit that consists of shift registers and XOR gates. For an q-bit LFSR, q shift registers are used to store one bit each. By definition, an LFSR produces a 2^q-1 long sequence of values given an initial value, or "seed" that is not all zeroes, and the LFSRs "connection polynomial" is irreducible. In Fig. 2.3 we can see an example of a Galois LFSR that follows the feedback polynomial of $x^{16}+x^{14}+x^{13}+x^{11}+1$. In general, for an LFSR of length q, an irreducible polynomial over the finite field $GF(2)$ with a degree of q must be used to create an LFSR with the above mentioned properties. The "taps",

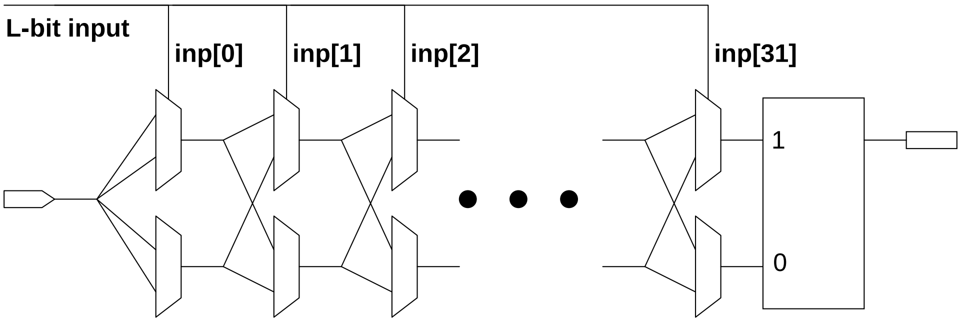

Figure 2.1 An arbiter based PUF. Each set of multiplexors are replicated to as many as the application requires, which is 32-bits in this example. An arbiter, usually a latch, is placed at the end and determines the outcome. This circuit is replicated 32 times to produce 32 output bits.

```
**for** (i from 1 to t):                                            (1)
    $l(x) = x^{(p^m-1)/r_i}$ mod $f(x)$                             (2)
    **if** $l(x) = 1$ then return (”NOT Primitive”).                (3)
return(”Primitive”).                                                (4)
```

Figure 2.2 Primality test algorithm for polynomials.

or the positions of the XORs, will have to be placed according to the exponentials expressed in these polynomials.

Tables such as (48) have been created to show where the taps should be placed. The tables are created by exhaustively testing polynomials for primality. Algorithm 4.77 in (43) shows how a polynomial can be tested for primality. The algorithm stems from Fact 4.76 in (43) which states:

Let p be a prime and the distinct prime factors of $p^m - 1$ *be r1,r2, ... ,*r_t*. An irreducible polynomial* $f(x) \in Z_p[x]$ *is primitive if and only if for each i,* $1 \leq i \leq t$*:*

$$x^{(p^m-1)/r_i} \not\equiv 1 \ mod \ f(x) \ , \tag{2.3}$$

where x is an element of order $p^m - 1$ *in the field* $Z_p[x]/(f(x))$*.*

The sequence that an LFSR produces has good statistical properties, and can be considered

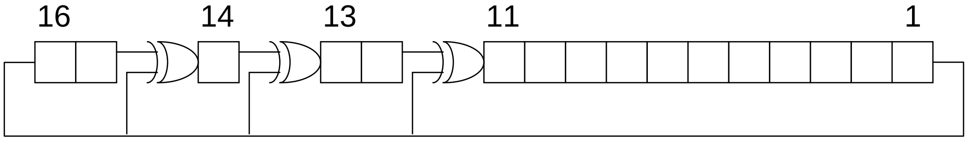

Figure 2.3 A 16-bit example of a Galois LFSR using the irreducible polynomial $x^{16} + x^{14} + x^{13} + x^{11} + 1$.

pseudo random (42). Therefore, if the seed of the LFSR is kept secret, the LFSR can be used to create q-bit pseudo random secrets. We denote the use of an LFSR as shown in (2.4).

$$LFSR(x) = y, \tag{2.4}$$

where x is the seed, and y is the output.

In our protocol, we use the secrets produced by the LFSR to obscure the transmissions between the reader and the tag. Being pseudo random, and not random, comes from the fact that the sequence does follow a mathematical formula. Additionally, the sequence does eventually repeat after $2^q - 1$ steps. The drawbacks of pseudo randomness will not adversely affect our solution however, since transmissions using LFSR outputs will be obscured by applying bitwise XOR operations with truly random values. Our protocols use far less than $2^q - 1$ steps, and will therefore not run out of values.

CHAPTER 3. REVIEW OF LITERATURE

3.1 Mutual Authentication

There have been several attempts to create authentication for RFID systems, we will first look at some of the hash based solutions. Weis et al. have proposed a Hash-lock solution and an improved Random Hash-lock solution (20). The Hash-lock solutions use hash functions as a basis for testing the authenticity of the reader and tag. A metaID is sent to the reader, who looks up the key for the tag and transmits the key to the tag, who performs a hash on the key to see if it matches with its metaID. These early protocols have several drawbacks, both from an implementation standpoint and a security standpoint. The expectation that tags can perform hash calculations is unrealistic due to hardware limitations. These protocols also suffer from replay attack vulnerability, since an adversary can replay any of the messages and get a response from the appropriate party.

Another set of hash-based solutions were proposed by Ohkubo et al. (21) and Henrici et al. (22)(23). The solution in (21) is a hash chain solution and requires the tag to perform a hash operation on its ID and send the result to the reader. The reader then makes an exhaustive search for a match. The solutions by Henrici follow the same vain, in that the tag and reader compare hash results. In (22) an improvement is made by introducing a time variable to prevent replay attacks. A further improvement follows in (23) where three types of hash functions are used to verify the messages between reader and tag. These solutions are not practical for the type of tag that we are considering here, due to the complexity of hash functions. Furthermore, none of these solutions authenticate the tag, and is therefore vulnerable to man-in-the-middle attacks, where an adversary has mined hash values from tags, and repeats them to the reader at a later time. Many other hash-based protocols exist, e.g.

(15) (24)(25)(26). Several of them do not provide mutual authentication, and common to all of them of course is that they assume that the tags in their solutions can accommodate for hash functions. Feldhofer et al. (4), and Bogdanov et al. (27) discuss their skepticism towards relying on hash functions for security ion RFID.

Another method of securing RFID systems has been the lightweight approach. These solutions base themselves on mostly bitwise operations instead of more expensive cryptographic primitives. The HB protocol by Hopper and Blum (14) is a lightweight protocol that is designed to prevent passive attackers. It works by sending a series of challenges to the tag and asking it to compute some simple bitwise operations. If the responses are correct to within a certain margin, the tag is accepted as authentic. An active attacker can easily recover the tags secret. An improved variant called HB+ was produced by Juels and Weis (3). This protocol attempts to prevent active attackers, but was broken by Gilbert et al. (29) by using a man-in-the-middle attack. A further improvement was made by Bringer et al. (28) called HB++, but this also has weaknesses discussed in (30). HB-PUF (31) is a PUF version of the HB protocol and can be compared to our solution since it also utilizes PUF circuits as a way to test a tag. The HB-PUF solution does not provide mutual authentication, but only authenticates the tag. Moreover, the HB-PUF protocol requires that the reader or database keeps a very large record of challenge-response pairs from the PUF of each tag. In contrast our solution only requires the reader or database to maintain two PUF results from each tag.

The MAP family of protocols, LMAP (12), M2AP (32) and EMAP (33), all devised by Peris-Lopez et al., are another group of lightweight protocols for RFID. The MAP protocols provide mutual authentication between tag and reader. These protocols also rely on bitwise operations to hide the ID of the tag and the shared secrets. These protocols have serious flaws that allow an attacker to learn all secrets and the ID of the tag with relative ease. LMAP was broken in (34) and (35), M2AP was broken in (34) and (36), and EMAP was broken in (37) and (38). Some attacks are active, and some are passive. For instance, M2AP can be broken by simply eavesdropping for two consecutive runs of the protocol. As mentioned above, all secrets are revealed by performing this simple passive attack.

Some protocols have tried to adhere strictly to the EPC Class 1 Gen 2 standard of tags. TRMA is one such protocol, proposed by Konidala and Kim in (13). This solution uses the pseudo random number generator on the tag, as well as bit manipulation of the 32-bit password on the tag to perform mutual authentication. TRMA was broken in (40) by Lim and Li, who were able to obtain the access password of the tag by eavesdropping. Konidala and Kim came up with an improved protocol named TRMA+ (39), which also utilizes the EPC Class 1 Gen 2 ”kill” password, which is normally used to permanently disable the tag. However TRMA+ was broken by Peris-Lopez et al. in (41), and was able to recover both access password and kill password, which would let an attacker disable the tag.

Some protocols based on PUF have been explored by (16) and (17). These solutions require that the back-end is preloaded with a very large amount of challenge response pairs for the reader to use to verify the authenticity of the tag. This is somewhat undesirable as the amount of tags could become very high, and become a strain on the resources of the system.

3.2 Ownership Transfer

There exist fewer contributions to the ownership transfer problem than to mutual authentication for RFID. Some of the solutions rely on hash functions or symmetric encryption functions (18)(44)(45)(49)(50), which we assume will not work for the cheapest tags since it pushes the required amount of hardware gates beyond the 2000 mark. A similar solution to our two-party ownership transfer protocol is mentioned in (46), using similar assumptions about the security of the backwards channel. The solution however depends on the tags ability to execute a cryptographic function.

One solution by Seo et al. (47) uses a very lightweight protocol which demands very little of the RFID tag hardware. However, the solution includes the notion of a proxy which stands between the reader and tag, and performs the authentication on its behalf. The inclusion of a proxy means that owners of tags would also need a proxy device, as well as a connection to the back-end database. A proxy may fail due to battery failure, crashes, damage and other factors. In the event that the proxy fails, so does a security suite based on its existence. It is therefore in our interest to come up with alternatives to proxy based solutions, given these drawbacks.

3.3 Secure Search

There have not been many attempts to produce a secure search protocol for RFID systems. In (15), Li et al. produced a series of search solutions that require very little storage, and is can therefore be regarded as a distributed solution without an explicit need for a back-end server. Li's solution bases itself on the RFID tag's ability to perform hash computations, which as of this writing is not feasible for low-cost RFID tags. Our solutions offer similar functions as Li's, asking very little in terms of memory usage by tags or readers, and in addition we do not rely on hash functions.

3.4 Physically Unclonable Functions

Some protocols based on PUF have been explored by (16) and (17). These solutions require that the back-end is preloaded with a very large amount of challenge response pairs for the reader to use to verify the authenticity of the tag. This is somewhat undesirable as the amount of tags could become very high, and become a strain on the resources of the system.

With regard to the variety of PUF implementations that exist, there is much work taking place. In addition to the MPUF approach, the BPUF was discussed in (1) as a method of producing identifying values based on varying delay in FPGAs. Another method of producing variation was explored in (2), where tri-state buffers were used in a delay chain. The tri-state buffer approach strongly resembles the MPUF approach, except tri-state buffers are used in place of multiplexors to create the delay chain. Lastly, the Ring-Oscillator PUF has been discussed in e.g. (17) and (9). This type of PUF measures the frequency of different oscillators based on the intrinsic delays in wires and other components of the circuit, and compare these frequencies to produce a unique output.

CHAPTER 4. Mutual Authentication

4.1 Introduction

RFID technology can provide great benefits in several areas and has many applications for both business and private individuals. A few examples include: streamlined inventory control, easy and fast checkout at retail stores, item interaction with ”smart home” devices, implantable tags for medical profile storage. To promote this great potential of RFID technology, the cost of RFID tags must be competitive with existing solutions such as bar codes, which are very low cost. There must also be contingencies for the many security and privacy issues that will arise if the technology is adopted to its full extent.

Although security usually seems to come at a higher premium, it is imperative that the security and privacy of the user is maintained. The invention of the Internet, and its many security problems, has shown us that security and privacy should ideally be in place before widespread deployment occurs. Specifically, the pressing issues in RFID security include eavesdropping, and tag cloning. Eavesdropping can lead to loss of privacy, since a tag may reveal its identity to any reader within its range. The attacker can later use the eavesdropped information to try and access either the tag or a legitimate reader by way of message replay. This kind of tag impersonation is often called ”cloning”, and can lead to fake merchandizing, or unauthorized access privileges. As RFID tags are placed in heavily targeted items such as credit cards and passports, it is clear that RFID needs security measures that are both affordable and reliable.

To keep the cost of RFID tags low, the hardware must be minimalistic. Passive RFID tags with no battery have between 200 - 2000 (3) hardware gates available for security measures. Less hardware means lower cost. Adding traditional security features to the RFID system will require a large amount of gates, and hence a higher price for each tag. A low-cost version of AES has been shown to require 3,400 gates, while hash functions such as MD5 and SHA-256 have been implemented using between 8,000 - 10,000 gates (4). Therefore, solutions relying on encryption functions and hash functions become prohibitively expensive in the case of low-cost RFID systems.

Previous work in this area has come short of providing solutions that are both affordable in terms of hardware, while at the same time preventing the threats against RFID systems. Solutions such as the authentication protocol of Li et al. (15) fall short of providing mutual authentication, and also use methods that are far too expensive in terms of hardware. Some previous solutions include the LMAP (12), TRMA (13) and HB (14) family of protocols, that have all been either broken or weakened.

In this chapter, we propose a lightweight solution to Mutual Authentication for RFID systems. Mutual Authentication mitigates the issues of eavesdropping and cloning of tags. Only authenticated readers and tags will successfully communicate with each other. Our solution only needs 850 gates for 64 bit variables, in contrast to solutions using hash functions that will need at least 8000 gates.

4.2 Problem Statement

4.2.1 System Model

Traditionally, an RFID system consists of three main entities: a reader, a tag, and a back-end database. The reader communicates wirelessly with the tag, who presents its identification number or other stored information to the reader upon request. The reader will then communicate with the database, either though a wired or wireless connection, as shown in Fig. 6.1. We will here assume that the communication between the reader and the database is secure in the sense that some kind of standard encryption technique is used. We further assume that an adversary can hear all transactions between reader and tag. In this paper we will denote the database as S, the reader as R and the tag as T.

There are three main classes of RFID tags: active, semi-active and passive tag. An active

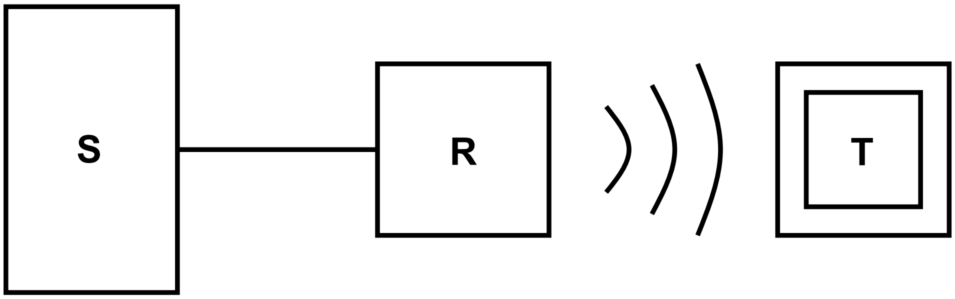

Figure 4.1 The RFID system model.

tag is the largest and most powerful variety and is usually powered by a battery. The battery is used to power the internal functions of the tag as well as transmission. Semi-active tags are also equipped with a power source, but these tags only use the batteries for powering the chip, not for communication. Lastly, passive tags have no battery, and use the electromagnetic signal from the reader as a source of power. The type of tag used in a RFID system depends on the needs of the application. For example, credit cards and passports are too small to contain batteries or large circuits and therefore only the passive tags are suitable for them. The traditional cryptography methods are too expensive for the passive tags, for example, one of the most popular passive tag is the EPC Class 1 Gen 2, which contains at most 2000 hardware gates available for security features (3). Our focus in this paper is providing lightweight security solutions for the low cost passive tags.

4.2.2 Problem Definition

4.2.2.1 Mutual Authentication

The problems that we want to solve in this work arise from the lack of secure mutual authentication protocols available for RFID systems. Without any added security, a naive approach to tag identification does not prove the identity of the tag to the reader, nor does it tell the tag that the reader can be trusted. In the credit card example, users of RFID-enabled credit cards that allow touchless checkout at the grocery store, will want assurance that nefarious readers are unable to harvest their sensitive information. On the other hand, the reader should be able to recognize and refuse fake credit cards being used. Generally, in any application where trust is a desired feature, identification protocols without mutual authentication will not suffice.

4.2.3 Threat Model

The mutual authentication protocol and ownership transfer protocol will face many threats launched by the attackers. The attackers can either active or passive. While the passive ones just eavesdrop on the messages transmitted between the reader and the tag, the active ones will further send out bogus messages, seeking to either impersonate the tag or the reader.

Active attackers can launch Physical attacks or Replay attacks. A *Physical attack* consists of probing the wires of the tag to learn the tag secrets and model the tag's behavior. Copying all the information from one tag to another is also a physical attack. A successful physical attack can give the adversary the ability to create fake tags, or impersonate a legitimate tag using some other device. In *Replay attack*, the adversary eavesdrops the communications between the reader and the tag, and replay the messages at a later time. The goal of a replay attack is either to attempt to impersonate someone, or to track an entity (usually a tag) for profiling purposes.

Passive attackers simply eavesdrops the transmissions between reader and tag. They intend to learn some secret or identifying information about the communicating parties. This information can then be used for the purpose of tracking or profiling, or finding secrets in other messages by utilizing bit manipulation or other offline methods.

4.2.4 Goals

4.2.4.1 Mutual Authentication

Our goals for the mutual authentication protocol include the following. Basically, the tag can authenticate the reader and the reader can authenticate the tag utilizing the protocol. Second, no private information is leaked to an eavesdropper or an active attacker. Third, the adversary cannot authenticate himself as a legitimate tag or reader, by manipulating the bits of the eavesdropped communications. Finally, the protocol should demand gate count below

2000 to be accommodated by the the EPC standard passive tags.

4.3 Mutual Authentication

To achieve mutual authentication between a reader and a RFID tag, we mean that the following two properties should be satisfied.

- **Property 1:** The tag can only accept and respond to the messages from the reader that the tag intends to, or equivalently, the tag can verify that the received messages are generated by the reader that the tag intend to authenticate.

- **Property 2:** The reader can only accept and respond to the messages from the tag that the reader intends to, or equivalently, the reader can verify that the received messages are generated by the tag that the reader intend to authenticate.

In this section, we first describe a naive mutual authentication protocol, then propose our solution by analyzing why the naive one does not work and how we could improve it.

4.3.1 A Naive Mutual Authentication Protocol

The protocol consists of two phases, an offline *setup phase* and an online *authentication phase*. Fig. 4.2 illustrates the detailed procedure of this protocol that is explained as follows:

Setup phase: In this phase, both the reader and the tag are preloaded with a tuple of three secrets $\{ID,\ G_n,\ G_{n+1}\}$ that is known only by themselves, where ID is the tag's identification, while G_n and G_{n+1} are two *greeting* numbers. Since the reader needs to authenticate multiple tags, it maintains a table that stores the tuples of all the tags that it can authenticate.

Authentication phase: In this phase, the reader and the tag exchange their secrets for mutual authentication. The phase may be executed multiple rounds, because the reader and the tag may have to authenticate each other multiple times. Each round of the phase consists of four steps.

- *Step 1*: The reader continuously broadcasts *Req* to request for the tag.

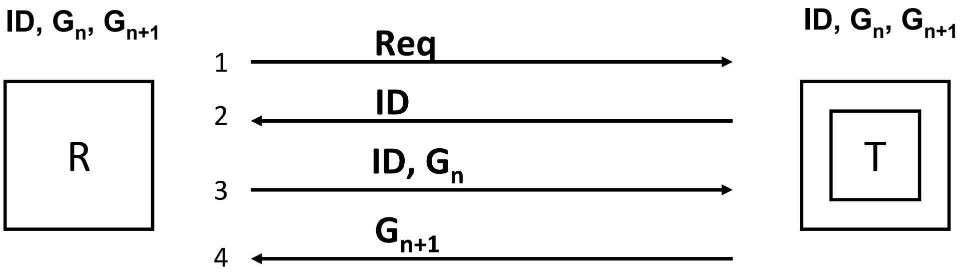

Figure 4.2 A naive mutual authentication protocol

- *Step 2*: Receiving *Req* from the reader, the tag responds to the reader its ID to claim its existence.

- *Step 3*: The reader looks up the tuples of all tags by the received ID. Once it finds the G_n corresponding to the ID, it sends back the ID along with the G_n to the tag. Since the tag is aware that only the reader and itself know the G_n, it is able to authenticate this reader, which satisfies the first property.

- *Step 4*: Receiving the correct G_n, the tag responds to the reader with its G_{n+1}. Similarly, the reader can authenticate the tag using this G_{n+1}, since only the particular tag knows that G_{n+1} besides the reader itself. Thus, the second property is satisfied.

There are several problems with this protocol. First, anyone overhearing the ID can track the tag later on. Second, anyone can learn the three secrets after overhearing all of the messages and impersonate either the reader or the tag later on. Third, to prevent from impersonating attacks, the protocol can be used only once. Essentially, all of the problems arise for the same reason, that is, the messages are not protected from eavesdropping attacks. To overcome these problems, we propose our solution.

4.3.2 Our Mutual Authentication Protocol

When designing our protocol, we have the following considerations:

- The tag ID should be protected. To allow the reader to identify a certain tag, we utilize a pseudonym called IDS, which will be updated every round of authentication.

- The greeting numbers should be protected. We use random numbers and XOR operations to construct an efficient encryption.

- The authentication process should be carried out many rounds. We choose to update the greeting numbers every round in order to allow the same reader and tag mutually authenticate each other as many times as possible.

Fig. 4.3 illustrates the procedure of our protocol that is explained here.

Setup phase: In this phase, both the reader and the tag share the following items:

- ID: the tag's identification number.

- IDS: a pseudonym that is updated every round and serves as the index of the tag's tuple stored in the reader's table.

- G_n: the *greeting* from the reader to the tag in current round, where n is the number of round of each mutual authentication process and it is initially set to 1.

- $F : [1, q] \rightarrow [1, q]$, a random permutation function mapping within range $[1, q]$, where $\log q$ is the length of tag's ID in bits. F serves as a random number generator and can be public. The key is that F must be efficient for low-cost tags to implement in hardware.

Besides these items, the reader also stores

- G_{n+1}: the *greeting* from the reader to the tag in the next round. It also serves as the expected greeting from the tag to the reader in current round.

In addition, the tag also implements a function P in itself.

- $P : [1, q] \rightarrow [1, q]$, a random permutation function mapping within range $[1, q]$. P is different from F in that each instance of P is only known to a certain tag, that is, every tag has a different P function. Although P can serve as a random number generator the same as F, in our protocol, it is used by the tag to authenticate itself to the reader. Another property of Pp is that even a tag is compromised, the adversary cannot construct an instance of P which is exactly the same as that stored in the compromised tag. So, P serves as a fingerprint of the tag and is unclonable.

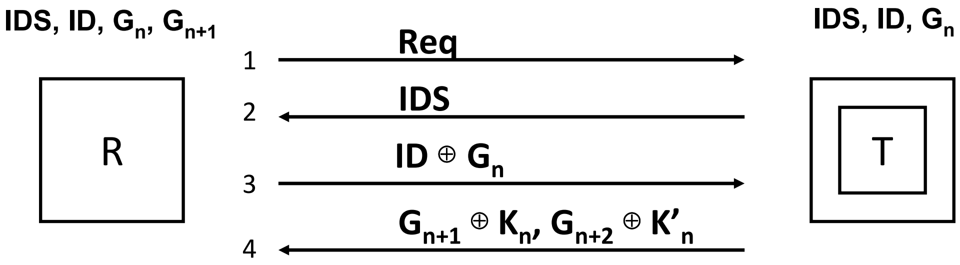
IDS, ID, G_n, G_{n+1}
R
1
Req
2
IDS
ID ⊕ G_n
3
G_{n+1} ⊕ K_n, G_{n+2} ⊕ K'_n
4
IDS, ID, G_n
T

Figure 4.3 Our mutual authentication protocol

Authentication phase:

- *Step 1*: The reader continuously broadcasts *Req* to request for the tag.
- *Step 2*: Receiving *Req* from the reader, the tag responds to the reader its IDS, which does not reveal the tag's ID.
- *Step 3*: The reader uses IDS to look up the tuple associated to the tag, then, it sends $ID \oplus G_n$ as response. In this response, ID is protected by G_n, as long as that G_n is never revealed. Hence, tracking the tag by its ID is impossible. Receiving $ID \oplus G_n$, the tag verifies its correctness using its own ID and G_n. If it is correct, the tag can authenticate the reader because the ID and G_n are shared between only the reader and the tag. So, our protocol satisfies the first property.
- *Step 4*: After the reader is authenticated, the tag calculates two greetings G_{n+1} and G_{n+2} from G_n using function P. That is,

  $$G_{n+1} = P(G_n) \ , \tag{4.1}$$

  and

  $$G_{n+2} = P(G_{n+1}) = P^2(G_n) \ . \tag{4.2}$$

  Then, the tag calculates K_n and K'_n from G_n using function F, so that

  $$K_n = F(G_n) \ , \tag{4.3}$$

  and

  $$K'_n = F(K_n) = F^2(G(n)) \ . \tag{4.4}$$

Now, the tag sends $G_{n+1} \oplus K_n$, $G_{n+2} \oplus K'_n$ back to the reader. Then, the reader can calculate K_n from G_n and recover G_{n+1} from $G_{n+1} \oplus K_n$. If it recover G_{n+1} correctly (compared with the copy of G_{n+1} it stores), it can authenticate the tag, because only the particular tag has the correct function P to produce G_{n+1}. Thus, our protocol satisfies the second property. Once the tag is authenticated, the reader extracts G_{n+2} from $G_{n+2} \oplus K'_n$ using K'_n, where this G_{n+2} will be used to authenticate the tag in the next round.

Finally, both the reader and the tag will update IDS on the fly. Let IDS_{old} denote the current IDS that has been used in step 2 and IDS_{new} denote the new copy. Then, they update IDS as follows:

$$IDS\_new = LFSR(IDS\_old \oplus G_n) \tag{4.5}$$

4.4 Security Analysis

In our analysis of the security of our schemes, we look at some of the common attacks that can be launched on RFID systems. We will describe the nature of the different attacks, as well as discuss how our solutions protect the RFID system from these attacks.

Physical attacks: A physical attack includes attempting to model the behavior of the PUF by probing the wires of the tag, or trying to clone the tag in other mechanical ways. The inclusion of a PUF circuit in our solution means that we can protect the tag from these attacks, since a PUF will alter its behavior if the hardware itself is altered. A probe on a wire will change the resistance in the link that is being probed, and therefore render the modeling attempts useless. If the content of the tag is somehow copied to another tag, the new tag will not be able to mimic the behavior of the original tag, because no two PUF circuits behave in exactly the same way. Although we do not directly address the issue of tag theft, a stolen tag will not compromise the security of other tags, since each tag authenticates differently from other tags. It is also possible to create the tag in such a way that if it is removed from the item it is attached to, it will become useless since the positions and composition of the wires will be altered.

Replay attack: As discussed earlier, the other form of active attack is a replay attack. An adversary may try to impersonate an authenticated tag or reader in order to gain some sort of access. Our protocol uses messages that become stale immediately after use, so a replay of a message will not be useful to an attacker. An active adversary may succeed in tracking the tag, as he can send a request to the tag, and the tag will reply with its IDS. The IDS itself does not offer any meaningful information about what item the tag is placed on or who owns the tag, but a persistent attacker could infer the physical identity of the tag that corresponds to a particular IDS, and track it until it changes upon mutual authentication.

Eavesdropping: An eavesdropping attack is when an adversary is listening in on the transmission between a tag and a reader in hopes of learning something useful, like an ID or a secret in order to exploit the system. A passive listener will fail in any tracking attempts since the IDS changes every time the mutual authentication protocol succeeds. The passive attacker will try to learn any of the shared secrets from listening in on the transmissions between tag and reader. All of the information that is sent wirelessly is encrypted. The randomness of G_n, G_{n+1}, and G_{n+2} ensures that all the messages become random. The inclusion of K_n, K'_n, and K''_n prevents simple cancelation of G_n values in successive rounds. Simple XOR encryption guarantees perfect security if one of the elements is random (51). Since all our messages contain random variables, the messages themselves will have to be brute forced to be found. With 64-bit variables, a brute force attempt faces $1/(2^{64})$ odds of succeeding.

After eavesdropping and collecting the messages, an attacker may try to manipulate the collected bits, as to attempt to exploit the commutative and associative nature of the XOR operation to find any secret information. The stalest part of our solution is the G_{n+2} variable, whose value is repeated 3 times in successive authentication rounds:

- MA 1, message 4: $G_{n+1} \oplus K'_n$, $G_{n+2} \oplus K''_n$
- MA 2, message 4: $G_{n+2} \oplus K'_n$, $G_{n+3} \oplus K''_n$
- MA 3, message 3: $ID \oplus G_{n+2}$

Can an adversary use this fact to perhaps solve for one of the secrets? The three equations that would be used are:

$$G_{n+2} \oplus K''_n = d1 \tag{4.6}$$

$$G_{n+2} \oplus K'_n = d2 \tag{4.7}$$

$$ID \oplus G_{n+2} = d3 \tag{4.8}$$

In these equations exist 4 unknowns, G_{n+2}, K''_n, K'_n, and ID. It is not possible to uniquely solve for any of the variables since we have more unknowns than equations.

Message blocking: Another form of attack is blocking of messages, which may not be an attack at all, but simply a result of lost messages. In our protocol, nothing adverse will happen if message 1, 2 , or 3 is blocked or dropped. However, if message 4 is blocked we have a synchronization problem. To mitigate this issue, we instruct the reader to store IDS_new as well as IDS. If IDS does not match the expected value, IDS_new is used to look up the greeting values and ID.

On the tag side, each tag stores G_{n-1} (along with G_n), to be ready to repeat the previous response. The system will not move on until the reader has received the correct data.

4.5 Wormhole attack

One possible attack on the mutual authentication protocol is a so called *Wormhole attack*. This attack is performed by a sophisticated attacker who relays messages between an authentic reader and an authentic tag. This attack will only work if the adversary has simultaneous communication access to both reader and tag. The attacker can impersonate the tag and reader because he simply acts as a relay station, without manipulating the message at all, and will therefore get the correct responses from the tag and reader and therefore authenticate himself as legitimate.

Although this attack may be too sophisticated for most adversaries, it can be done wit the correct equipment. However, several solutions exist. For instance, if the tag and attacker are both in the range of the reader, the reader will receive two replies, one from the attacker, and one from the tag. In such an event, it is likely that the tag is physically closer to the reader than the attacker, and the reader will only trust the first reply.

In a situation where the attacker relays the messages beyond the communication range of both reader and tag, we need different solutions. If the attacker wants to relay the message from reader to tag and then from tag to reader across a longer distance, the total transmission time will be longer than the average response time from the tag under normal circumstances. This can give the reader a reason for suspicion, and raise an alarm. Other solutions could include analysis of the physical signal received from the tag, to see if it matches the expected characteristics of a tag response, or some other type of test of response time.

CHAPTER 5. Ownership Transfer

5.1 Introduction

RFID has many applications for both business and private individuals. Several of these applications will include items that change owners at least once in its lifetime. The swapping and resale of items is a practice that is likely to be popular in the future, and so any item that depends on RFID for function or convenience should be equipped to deal with change of ownership. Ownership transfer presents its own set of threats, and therefore demands the attention of security researchers.

The pressing issues in RFID security include eavesdropping, and tag cloning. Eavesdropping can lead to loss of privacy, since a tag may reveal its identity to any reader within its range. The attacker can later use the eavesdropped information to try and access either the tag or a legitimate reader by way of message replay. This kind of tag impersonation is often called ”cloning”, and can lead to fake merchandizing, or unauthorized access privileges.

To keep the cost of RFID tags low, the hardware must be minimalistic. Passive RFID tags with no battery have between 200 - 2000 (3) hardware gates available for security measures. Less hardware means lower cost. Adding traditional security features to the RFID system will require a large amount of gates, and hence a higher price for each tag. A low-cost version of AES has been shown to require 3,400 gates, while hash functions such as MD5 and SHA-256 have been implemented using between 8,000 - 10,000 gates (4). Therefore, solutions relying on encryption functions and hash functions become prohibitively expensive in the case of low-cost RFID systems.

Previous work in this area has come short of providing solutions that are both affordable in terms of hardware, while at the same time preventing the threats against RFID systems. Such solution include those in (18)(44)(45)(49)(50), which we assume will not work for the cheapest tags since it pushes the required amount of hardware gates beyond the 2000 mark.

In this chapter, we propose a variation of our Mutual Authentication scheme which lends itself to securing Ownership Transfer, which is a pertinent issue in the scope of RFID. When an item passes from one owner to another, it is undesirable for the old owner to be able to access the tag or read data from it. The new user must therefore update the access-granting information without revealing this to the old owner. We describe how our Mutual Authentication scheme can be adapted to secure the Ownership Transfer of RFID tags. Our solution only needs 850 gates for 64 bit variables, in contrast to solutions using hash functions that will need at least 8000 gates.

5.2 Problem Statement

5.2.1 System Model

Traditionally, an RFID system consists of three main entities: a reader, a tag, and a back-end database. The reader communicates wirelessly with the tag, who presents its identification number or other stored information to the reader upon request. The reader will then communicate with the database, either though a wired or wireless connection, as shown in Fig. 6.1. We will here assume that the communication between the reader and the database is secure in the sense that some kind of standard encryption technique is used. We further assume that an adversary can hear all transactions between reader and tag. In this paper we will denote the database as S, the reader as R and the tag as T.

There are three main classes of RFID tags: active, semi-active and passive tag. An active tag is the largest and most powerful variety and is usually powered by a battery. The battery is used to power the internal functions of the tag as well as transmission. Semi-active tags are also equipped with a power source, but these tags only use the batteries for powering the chip, not for communication. Lastly, passive tags have no battery, and use the electromagnetic signal from the reader as a source of power. The type of tag used in a RFID system depends on the needs of the application. For example, credit cards and passports are too small to contain

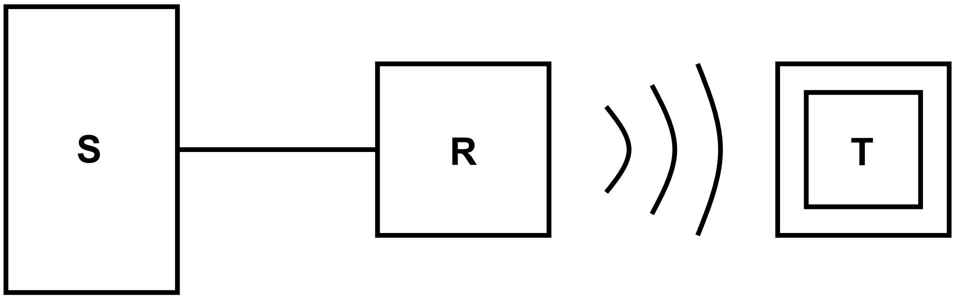

Figure 5.1 The RFID system model.

batteries or large circuits and therefore only the passive tags are suitable for them. The traditional cryptography methods are too expensive for the passive tags, for example, one of the most popular passive tag is the EPC Class 1 Gen 2, which contains at most 2000 hardware gates available for security features (3). Our focus in this paper is providing lightweight security solutions for the low cost passive tags.

5.2.2 Problem Definition

5.2.2.1 Ownership Transfer

In the case of ownership transfer, we want to prevent the old owner of a tag from accessing that tag or the services that the tag provides to the current owner. For instance, an old owner should not be able to use any keys or identifying values to impersonate the tag to make a purchase, open a door, or even change the ownership back to him. Observing that tags may become highly ubiquitous in the future, with tagged object changing hands often, secure owner transfer would be essential to the RFID systems.

5.2.3 Threat Model

The mutual authentication protocol and ownership transfer protocol will face many threats launched by the attackers. The attackers can either active or passive. While the passive ones just eavesdrop on the messages transmitted between the reader and the tag, the active ones will further send out bogus messages, seeking to either impersonate the tag or the reader.

Active attackers can launch Physical attacks or Replay attacks. A *Physical attack* consists of probing the wires of the tag to learn the tag secrets and model the tag's behavior. Copying all the information from one tag to another is also a physical attack. A successful physical attack can give the adversary the ability to create fake tags, or impersonate a legitimate tag using some other device. In *Replay attack*, the adversary eavesdrops the communications between the reader and the tag, and replay the messages at a later time. The goal of a replay attack is either to attempt to impersonate someone, or to track an entity (usually a tag) for profiling purposes.

Passive attackers simply eavesdrops the transmissions between reader and tag. They intend to learn some secret or identifying information about the communicating parties. This information can then be used for the purpose of tracking or profiling, or finding secrets in other messages by utilizing bit manipulation or other offline methods.

5.2.4 Goals

5.2.4.1 Ownership Transfer

Our ownership transfer protocol should ensure the new reader (owner) and the tag can setup some initial secretes to start the mutual authentication process. In addition, it should guarantee the previous owner of the tag cannot derive any secrets or get accessed to the tag or impersonate the tag to deceive the reader.

5.3 Ownership Transfer Protocol

In this section, we discuss the Ownership Transfer Protocols which are natural extensions of the Mutual Authentication protocols. First, an ownership transfer protocol should satisfy the following two properties:

- *Property 1*: The old owner should not be able to access the tag after the ownership transfer has taken place.

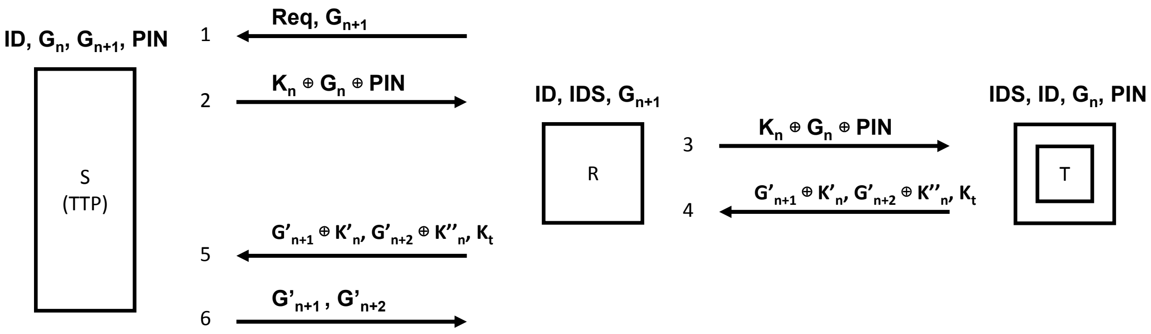

Figure 5.2 The ownership transfer protocol. The transmissions between S and R are considered secure.

- *Property 2*: The new owner should be able to perform mutual authentication with the tag after the ownership transfer has taken place.

We propose two protocols both of which satisfy the above properties. They can be used in different scenarios. The first one assumes the existence of an authority trusted by both the RFID reader and the tags, named the Trusted Third Party (TTP). The main task of the TTP is to assist the reader and the tag to construct a new verification pair of (G'_n, G'_{n+1}), such that the reader and the tag can start the mutual authentication process described in section III. The second ownership transfer protocol involves no third party, instead, the asymmetric communication between the reader and the tag is assumed: the tag-to-reader range is much smaller than the reader-to-tag range, and is assumed not to be intercepted/eavesdropped by the adversaries. This assumption is reasonable in that it is consistent to the hardware implementation, and has also been used by previous works [][].

5.3.1 Ownership Transfer using TTP

The communications between the TTP and the readers are through wired links and can be safely protected from adversaries. However, the communications between the TTP and the tag are not necessarily direct because the TTP may locate somewhere out of the communication range of the tag, Hence, the TTP will send messages first to the reader who will then relay the message to the tag. The protocol steps are shown in Fig. 5.2.

- Setup: To initiate the transfer of the tag, the old owner gives it's stored tuple (IDS, ID, G_{n+1}) to the new owner, meanwhile, it informs the TTP about the verification pair (G_n, G_{n+1}). A secret PIN is securely shared between the TTP and the tag. The PIN is preloaded in the tag hardware during production and is not accessible to anyone, e.g., neither the previous and the current owner of the tag knows about the value of PIN.
- Step 1: The new reader sends a secure request to the TTP, using G_{n+1} as proof that it has access rights to the tag.
- Step 2: The TTP can verify if the received G_{n+1} from the new reader equals the one received from the previous owner, if yes, then the new reader gets authenticated. Then the TTP sends $K_n \oplus G_n \oplus PIN$ to the reader, where K_n is the LFSR result using PIN as the seed:

$$K_n = LFSR(PIN) \tag{5.1}$$

- Step 3: The reader relays this message to the tag.
- Step 4: The tag first verifies if this message is originated from the TTP. It computes K_n using Eq. (5.1) and XORs both PIN and K_n with the message to obtain G_n. If the computed G_n equals the one it stores, then the message is verified to be sent by the TTP.

  The tag now generates a new pair of (G'_n, G'_{n+1}) to replace the old pair, such that the new reader and the tag can starts their mutual authentication process. The new pairs are generated using the following equations:

$$G'_n = P(G_{n+1}) \tag{5.2}$$

$$G'_{n+1} = P(G'_n) \tag{5.3}$$

  Since the old owner of the tag has seen the values of G_n, G_{n+1} and G_{n+2} in previous exchanged messages, he has no knowledge of the new G'_n or G'_{n+1} which are the outputs of the PUF function. In other words, without the correct values of G'_n and G'_{n+1}, the old owner cannot access the tag any longer. Property 1 is satisfied.

To protect (G'_n, G'_{n+1}) from being eavesdropped from the old owner and other malicious eavesdroppers, the tag generates another two random numbers using PIN:

$$\begin{aligned} K'_n &= LFSR(K_n) && (5.4) \\ K''_n &= LFSR(K'_n) && (5.5) \end{aligned}$$

The random numbers are XORed with the G'_n and G'_{n+1} respectively, namely, $K'_n \oplus G'_{n+1}$ and $K''_n \oplus G'_{n+2}$ are computed. Additionally, the tag computes $LFSR(G_n \oplus G_{n+1})$, denoted by K_t, which will help the TTP to verify that this message is originated from the tag. The three parts all together are sent back to the reader.

- Step 5: The reader relays this message to the TTP.
- Step 6: Upon receiving the message, the TTP verifies the correctness of the value K_t, computes the random numbers K'_n and K''_n using Eq. (5.4) and (5.5), obtains the values of the pair value of G'_n and G'_{n+1}, and sends them back through the secure channel to the new reader. Now the reader can start performing a normal mutual authentication with the tag. Property 2 is satisfied.
- Step 7: Both the TTP and the tag can do the updating of PIN internally (this step is not depicted in the figure), using the LFSR and the old G_n, as shown in the following equation:

$$PIN\_new = LFSR(PIN\_old \oplus G_n) \qquad (5.6)$$

5.3.2 Two-Party Ownership Transfer

A two-party ownership transfer solution is a protocol without a TTP, i.e., only the new owner and the tag exchange messages to update to a new pair (G_n, G_{n+1}). Such a protocol can be constructed using our original mutual authentication protocol directly. An added assumption is that the backwards channel, i.e. the tag-to-reader communications cannot be received by anyone but the authenticated reader []. Fig. 5.3 depicts the two-party ownership transfer protocol. The setup step is similar to that in the TTP protocol, i.e., the old owner gives the tuple stored for the particular tag (IDS, ID, G_n, G_{n+1}) to the new owner.

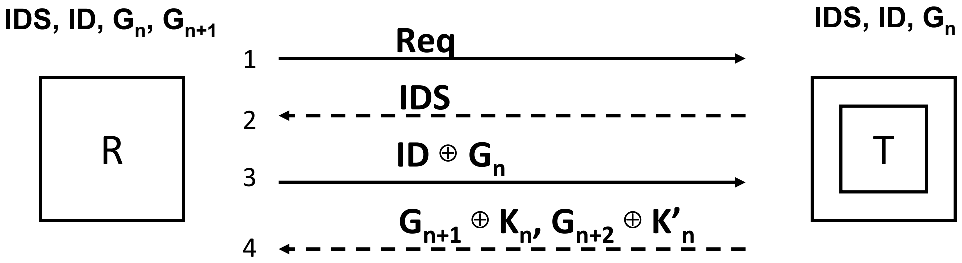
IDS, ID, G_n, G_{n+1}
IDS, ID, G_n
R
T
1
Req
2
IDS
3
ID ⊕ G_n
4
G_{n+1} ⊕ K_n, G_{n+2} ⊕ K'_n

Figure 5.3 Two-party ownership transfer protocol. Dashed lines indicate communications that can only be heard by the new owner.

After two mutual authentication rounds, the old owner no longer has control over the tag since the last message is only heard by the new owner, therefore ownership transfer Property 1 is fulfilled. Property 2 of ownership transfer is satisfied as soon as the new owner receives the tag data from the old owner. Therefore, the two-party ownership can successfully achieve ownership transfer, and can serve as a complement to the TTP protocol when the third party is unavailable.

5.4 Security Analysis

Since the ownership transfer solutions are modeled after our mutual authentication protocols, the security analysis will resemble the analysis of our mutual authentication solutions. In this section, we will describe the nature of the different attacks, as well as discuss how our solutions protect the RFID system from these attacks.

Physical attacks: A physical attack includes attempting to model the behavior of the PUF by probing the wires of the tag, or trying to clone the tag in other mechanical ways. The inclusion of a PUF circuit in our solution means that we can protect the tag from these attacks, since a PUF will alter its behavior if the hardware itself is altered. A probe on a wire will change the resistance in the link that is being probed, and therefore render the modeling attempts useless. If the content of the tag is somehow copied to another tag, the new tag will not be able to mimic the behavior of the original tag, because no two PUF circuits behave in exactly the same way. Although we do not directly address the issue of tag theft, a stolen tag will not compromise the security of other tags, since each tag authenticates differently from other tags. It is also possible to create the tag in such a way that if it is removed from the item it is attached to, it will become useless since the positions and composition of the wires will be altered.

Replay attack: The other form of active attack is a replay attack. An adversary may try to impersonate an authenticated tag or reader in order to gain some sort of access. Our protocol uses messages that become stale immediately after use, so a replay of a message will not be useful to an attacker. In our two-party solution, the adversary cannot hear the transmission of IDS, and so this value cannot be used for tracking as is possible from a dedicated attack on the mutual authentication protocol.

Eavesdropping: An eavesdropping attack is when an adversary is listening in on the transmission between a tag and a reader in hopes of learning something useful, like an ID or a secret in order to exploit the system. A passive listener will fail in any tracking attempts since the PIN changes every time the ownership transfer protocol succeeds. The passive attacker will try to learn any of the shared secrets from listening in on the transmissions between tag and reader. All of the information that is sent wirelessly is encrypted. The randomness of G_n, G'_{n+1}, and G'_{n+2} ensures that all the messages become random. The inclusion of K_n, K'_n, and K''_n prevents simple cancelation of G_n values in successive rounds. Simple XOR encryption guarantees perfect security if one of the elements is random (51). Since all our messages contain random variables, the messages themselves will have to be brute forced to be found. With 64-bit variables, a brute force attempt faces $1/(2^{64})$ odds of succeeding.

5.5 Wormhole attack

One possible attack on the mutual authentication protocol is a so called *Wormhole attack*. This attack is performed by a sophisticated attacker who relays messages between an authentic reader and an authentic tag. This attack will only work if the adversary has simultaneous communication access to both reader and tag. The attacker can impersonate the tag and reader because he simply acts as a relay station, without manipulating the message at all, and will therefore get the correct responses from the tag and reader and therefore authenticate himself as legitimate.

Although this attack may be too sophisticated for most adversaries, it can be done wit the correct equipment. However, several solutions exist. For instance, if the tag and attacker are both in the range of the reader, the reader will receive two replies, one from the attacker, and one from the tag. In such an event, it is likely that the tag is physically closer to the reader than the attacker, and the reader will only trust the first reply.

In a situation where the attacker relays the messages beyond the communication range of both reader and tag, we need different solutions. If the attacker wants to relay the message from reader to tag and then from tag to reader across a longer distance, the total transmission time will be longer than the average response time from the tag under normal circumstances. This can give the reader a reason for suspicion, and raise an alarm. Other solutions could include analysis of the physical signal received from the tag, to see if it matches the expected characteristics of a tag response, or some other type of test of response time.

CHAPTER 6. Secure Search

6.1 Introduction

RFID technology can potentially be applied almost everywhere. In the foreseeable future, we may become as dependent on RFID technology as we are on e-mail or cellular phones today. A typical RFID system involves a reader and a number of tags, which may range from the battery-powered tags with Wi-Fi capabilities, to the low-cost ones that are constrained in resources with even no internal power.

Keeping RFID systems secure is important, because they are vulnerable to a number of malicious attacks such as eavesdropping, impersonating or physically compromising. As for low-cost RFID systems, security problems become much more challenging, because many traditional security mechanisms such as symmetric key encryption/decryption and hash function are inefficient or even impossible due to resource constraints. For example, a passive RFID tag can hold only 200 to 2000 hardware gates for security purposes (3). However, one implementation of AES algorithm has been reported to consume 3400 gates, while an implementation of MD5 and SHA-256 hashing algorithms require 8000 to 10000 gates (4). Therefore, security solutions for low-cost RFID tags cannot rely on these methods.

One important functionality that a RFID system should provide is tag search, where a reader can detect if a particular tag is present or not. Tag search poses challenge to security and privacy. For example, an adversary may eavesdrop on the wireless communication between the reader and the tag to learn the identity of the tag and then track the movement or location of the tag. Or he may impersonate the tag and cheat the reader to believe the tag's presence. So, secure search protocols are demanding. In this paper, we study secure search problem with a focus on low-cost RFID systems, while existing solution (15) is based on hash function, which is too expensive for low-cost RFID systems.

In this chapter, we propose several lightweight secure search protocols. Our solutions can prevent adversaries from learning tag identity and from replaying query and response messages to impersonate RFID reader or tag. Moreover, they are built on top of Linear Feedback Shift Registers (LFSR) and Physically Unclonable Functions (PUF), which are very efficient for implementation in low-cost tags. We use LFSR to generate random numbers for encrypting communication and reply on PUF to authenticate the identity of tags. Experimental results show that our solutions have negligible processing time and require no more than 1400 hardware gates. So, they are very suitable for low-cost RFID systems with at most 2000 gates available for security purposes.

6.2 Problem Statement

6.2.1 System Model

A RFID system usually consists of three main components: a reader, a tag, and a back-end database. The communication channel between the reader and the tag is wireless, while that between the reader and the database can be either wired or wireless. Fig. 6.1 shows a simple RFID system in which the interface between the reader and the database is wired. The tag presents its identification number or other stored information to the reader upon request. The reader will then communicate with the database. We assume that the communication between the reader and the database is secure due to the use of some kind of standard encryption technique. We further assume that an adversary can hear all transactions between a reader and a tag. In this paper we denote the database as S, the reader as R and the tag as T, although our solutions do not explicitly require a database.

There are three main types of RFID tags, active tags, semi-active tags and passive tags. An active tag has its own battery to power the internal circuitry of the tag as well its transmission component. Semi-active tags are also features its own power source, but they only use the batteries for powering the internal chip, not the communication. The passive tags have no internal battery, but use the electromagnetic signal from the reader as a source of power. In

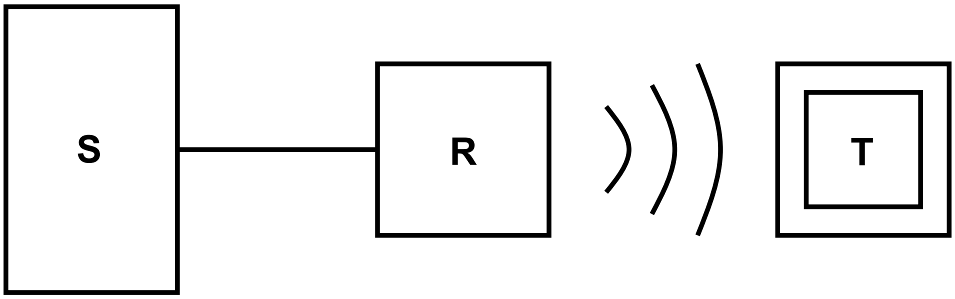

Figure 6.1 The RFID system model.

this paper, we focus on passive tags, which are low-cost and resource-constrained. For example, the most popular passive tag, EPC Class 1 Gen 2, has at most 2000 hardware gates available for security features (3).

6.2.2 Threat Model

RFID systems face with many threats launched by the attackers. The attackers can either active or passive. The passive attackers mainly launch *eavesdropping attacks* to capture the messages transmitted between the reader and the tag. They intend to learn some secret or private information about the communicating parties. This information can then be used for the purpose of tracking or profiling, or finding secrets in other messages by utilizing bit manipulation or other offline methods.

The active attackers can jam wireless communication, send out bogus messages, or compromise some tag. In this paper, we focus on *physical attacks* or *replaying attacks* launched by the active attackers. A *physical attack* consists of probing the wires of the tag to learn the tag secrets and model the tag's behavior. Copying all the information from one tag to another is also a physical attack. A successful physical attack can give adversaries the ability to create fake tags, or impersonate a legitimate tag using some other device. In *replaying attacks*, the adversaries eavesdrop on the communication between the reader and the tag, and replay the messages at a later time. The goal of a replay attack is either to attempt to impersonate someone, or to track an entity (usually a tag) for profiling purposes.

6.2.3 Problem Definition

One important functionality of RFID systems is tag search. Imagine a warehouse full of tagged items and a manager of the warehouse wants to know if a particular item is present in the warehouse or not. The manager can use a reader to query the tag attached to that item and listen for a correct response from the tag to detect the presence of the item. Based on this example application, we define tag search problem as how a reader could determine among a number of tags whether a particular one is present. Here, we call the tag being search for the *target tag*. We assume that the reader knows the identity (ID) of the target tag and therefore initiates a search with this ID.

A naive search protocol is that the reader broadcasts the ID and the target tag sends back a response. However, this protocol involves severe privacy and security problems. For example, an adversary can easily track the location of the tag using its ID he overheard, or he can forge the presence of the tag by replaying the overheard response. To solve these problems, we demand a *secure search protocol*. By a secure search protocol between a reader and a tag, we mean that the following two properties should be satisfied.

- **Property 1:** *Only the reader is aware of the identity of the target tag, but an eavesdropper cannot infer the tag's identity from the communication between the reader and the tag.*
- **Property 2:** *The reader can determine the presence of the tag, but an adversary is not able to forge the tag's presence if it is not present.*

Except for these two properties, a secure search protocol should be efficient, since our focus is the low-cost passive tags that provide at most 2000 gates for the implementation of security protocols.

6.3 Our Secure Search Protocols

Intuitively, to satisfy the two properties of secure search protocol, we need to encrypt both query and response in order to prevent an eavesdropper from learning the identity of the target tag. Meanwhile, the messages should be changed each time of search in order to prevent an adversary from replaying them. Based on these ideas, we design several secure search protocols. Each of our protocols consists of two phases, an offline *setup phase* and an online *search phase*. In the setup phase, the reader and all the tags are preloaded with some secrets. Then, in the search phase, they exchange their secrets for the reader to detect the presence of the target tag.

6.3.1 A Basic Protocol

Fig. 6.2 illustrates the detailed procedure of our first secure search protocol, called the basic protocol. For simplicity, we only show the reader and the target tag in the figure. Our protocol consists of a setup phase and a search phase, which are described as follows.

Setup phase: In this phase, the reader and all the tags are preloaded with some secrets. First, the reader and the target tag share three items:

- ID_T, the identity of the target tag, where ID_T is a q-bit integer such that $1 \leq ID_T \leq 2^q$. Note: each tag has its own ID and the reader may contain multiple IDs if more than one tag should be searched for.

- K, the shared secret key between the reader and the target tag, where K is also q-bit. Note: each tag may share a different key with the reader.

- $L : [1, 2^q] \rightarrow [1, 2^q]$, a random permutation function whose input and output are both q-bit integers. L function serves as a random number generator and is constructed using Linear Feedback Shift Register (LFSR) (42). (As we will discuss later, LFSR is very efficient for low-cost tags to implement in hardware.) Note: the construction of L function is public, so all the tags including even adversaries know how to build it.

Besides these three items, each tag contains:

- $P : [1, 2^q] \rightarrow [1, 2^q]$, another random permutation function operating within the range of $[1, 2^q]$. P function is constructed using Physically Unclonable Function (PUF) (17) (which will be discussed later). Each tag has its own instance of P function, while all

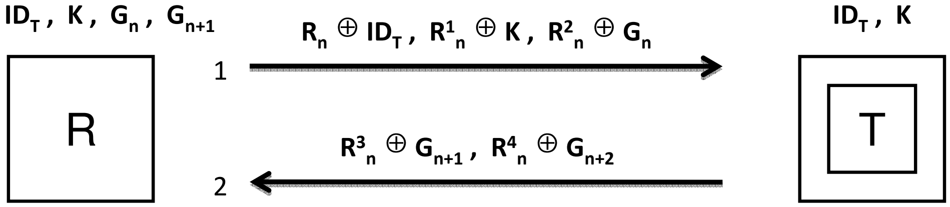
ID_T , K , G_n , G_n+1
R
1
R_n ⊕ ID_T , R^1_n ⊕ K , R^2_n ⊕ G_n
2
R^3_n ⊕ G_n+1 , R^4_n ⊕ G_n+2
ID_T , K
T

Figure 6.2 The procedure of the basic protocol, where ID_T is the identity of target tag and K is a shared key. G_n, G_{n+1} and G_{n+2} are greeting numbers, while R_n and R_n^i for $i = 1, \cdots, 4$ are random numbers.

these instances share the same physical construction. A nice property of PUF is its *physical unclonability*. That is, given the same input, each instance of P will generate a different output. Thus, an instance of P serves as a fingerprint of the corresponding tag. It means that even after an adversary compromises a tag, he has no way to produce a new tag behaving exactly the same as the compromised one by cloning its P function.

In addition, the reader stores two *greeting* numbers such as:

- G_n, the *greeting* from the reader to the tag used in the current round of search, where n denotes the index of round and $n = 1$ initially. Here, we define a round as a successful search in which the reader receives a correct response for its query. The bit length of G_n is q.

- G_{n+1}, the *greeting* used in the next round, where G_{n+1} is also q-bit. In our protocol, G_{n+1} is computed from G_n by the target tag using its P function such that

$$G_{n+1} = P(G_n) \ . \tag{6.1}$$

Search phase: In this phase, the reader first broadcasts a search query to all the tags in its communication range. Then, every tag performs some operations on the received query to determine if the query is meant for it. Once a tag identifies itself as the target one, it sends back a response announcing its presence. More precisely, this phase consists of the following two steps.

- *Step 1*: To search for a particular tag ID_T, the reader broadcasts its query

$$\{R_n \oplus ID_T, R_n^1 \oplus K, R_n^2 \oplus G_n\} \ , \tag{6.2}$$

where R_n is a random number generated by the reader to mask ID_T. R_n^1 and R_n^2 are two sequential random numbers generated from R_n using L function, that is,

$$R_n^1 = L(R_n) \ \text{ and } \ R_n^2 = L(R_n^1) \ . \tag{6.3}$$

Receiving this query, each tag tries to derive R_n by XORing $R_n \oplus ID_T$ with its ID. Then, it calculates R_n^1 from the derived R_n and tries to verify K from $R_n^1 \oplus K$. Since only the target tag has the correct ID, only it can successfully derive R_n and verify K. Then, it derives G_n from $R_n^2 \oplus G_n$. Other tags will discard the query, because they cannot verify K.

- *Step 2*: Obtaining G_n, the target tag calculates G_{n+1} and G_{n+2} sequentially using its P function, that is,

$$G_{n+1} = P(G_n) \ \text{ and } \ G_{n+2} = P(G_{n+1}) \ . \tag{6.4}$$

Then, it calculates two sequential random numbers R_n^3 and R_n^4 from R_n^2 using L function such as

$$R_n^3 = L(R_n^2) \ \text{ and } \ R_n^4 = L(R_n^3) \ . \tag{6.5}$$

Finally, the target tag masks G_{n+1} and G_{n+2} with R_n^3 and R_n^4, respectively, and sends back its response

$$\{R_n^3 \oplus G_{n+1}, R_n^4 \oplus G_{n+2}\} \ . \tag{6.6}$$

Receiving this response, the reader first derives G_{n+1} and then compares the derived copy of G_{n+1} with another copy of G_{n+1} it stores. If two copies match, it determines the presence of the tag. As we mentioned before, P function is unique for each tag and serves as a fingerprint of each tag. So, the reader can guarantee the tag's presence when receiving the correct G_{n+1}, because it can be generated only by the target tag. Receiving the correct G_{n+1}, the reader derives G_{n+2} from $R_n^4 \oplus G_{n+2}$ and then updates the stored G_n and G_{n+1} into G_{n+1} and G_{n+2} for the search of next round.

6.3.2 Security Analysis for the Basic Protocol

Built on top of LFSR and PUF, our search protocol is not only efficient for low-cost RFID tags, but also robust against a number of malicious attacks.

Eavesdropping attacks: In these attacks, adversaries eavesdrop on the communication between a reader and a tag hoping to learn some secrets such as the tag's ID, the shared key or the greeting numbers, in order to exploit the systems, e.g., tracking the tag.

In our protocol, all the secrets are XORed with some random numbers changed every time of search. Shannon's theory (51) proves that a simple XOR encryption can guarantee perfect security if at least one of the elements involved in XOR operations is random. So, the adversaries have no way but brute force the secrets. Given q-bit for each secret, a brute force attempt faces $\frac{1}{2^q}$ odds of succeeding. Typically, we can set q as 64 for providing an appropriate level of security to low-cost RFID systems. Meanwhile, knowing the construction of L function could not be more helpful for the adversaries to learn any secrets, because they do not know R_n.

One observation of our protocol is that a greeting number may be repeated in three consecutive rounds. Would it be more helpful for the adversaries if they can capture the messages of multiple rounds? The answer is negative. Assuming an adversary targeting G_n buffered the messages from the $(n-2)$-th, the $(n-1)$-th and the n-th round. Then, he can obtain the following three equations:

$$\begin{aligned} V_{n-2} &= R^4_{n-2} \oplus G_n \ , \\ V_{n-1} &= R^3_{n-1} \oplus G_n \ , \\ V_n &= R^2_n \oplus G_n \ , \end{aligned} \tag{6.7}$$

where V_{n-2}, V_{n-1}, and V_n are the values he derived from the corresponding rounds. Obviously, the adversary has three equations with four unknowns, so he cannot derive G_n.

Physical attacks: In physical attacks, the adversaries attempt to model the behavior of the tag's P function by probing the wires of the tag, or trying to clone the tag in other mechanical ways.

In our protocol, the implementation of P function based on PUF circuit can protect the tag from physical attacks. A probe on a wire of PUF will change the resistance in the link that is being probed, and therefore render the PUF to alter its behavior. Meanwhile, as we already mentioned, P function is unclonable. So, if the content of the tag is somehow copied to another tag, the new tag will not be able to mimic the behavior of the original tag, because no two PUF circuits behave exactly the same.

Note: although we do not directly address the issue of tag theft, one possible solution is available, that is, the tag can be created in such a way that once it is removed from the item it is attached to, it will become useless since the positions or composition of the wires will be altered.

Replaying attacks: In these attacks, an adversary can either replay a tag's response to forge a presence of the tag, or replay a previous query in order to impersonate the reader to track the tag.

In our protocol, replaying a tag's response is not useful. Each time when the reader receives a response, it will update the greeting number it stores, however, a previous response contains only an old greeting. Therefore, replaying such a response could not make the reader to believe a presence of the tag.

Meanwhile, replaying a query is also no a problem. In our protocol, receiving a replayed query, the target tag will send back its response. Since the response is encrypted, so no secrets will be revealed. The adversary, even observing the existence of a response, still does which tag it is. So, he cannot track a tag as he wants. However, strictly speaking, this replying attack does leak some information about the tag, because the adversary observing a response is aware that at least some tag is present. From this point of view, the basic protocol can still be improved.

6.3.3 A Synchronization-based Protocol

In the basic protocol, the target tag always respond a query, because it maintains no records about the state of the reader's query. If it can somehow memorize what queries have been used, then it can recognize the replayed query and keep silence. Following this idea, we propose our second secure search protocol, called a synchronization-based protocol, because it make the tag synchronized with the reader. The detail procedure is shown in Fig. 6.3. Since this protocol is similar to the basic one, we briefly describe it and focus on the differences between these two protocols.

Setup phase: In this phase, the reader and the target tag share no secret key K. Instead, the tag stores two greeting numbers: G_{n-1}, the last greeting number, and G_n, the current greeting number. From the perspective of the tag, G_n can also be regarded as the expected greeting from the reader. By recording G_{n-1} and G_n, the tag is now able to memorize the state of query. (Note: in the first round, only G_1 is needed, so we can ignore G_0 or directly set it as zero.)

Search phase: In this phase, the reader broadcasts its search query to all tags in the communication range. The target tag checks for the greeting in the query to see if the query is a replayed one. It does not respond a replayed query (except for the last query). Then, the tag decides if the stored greeting numbers should be updated. By updating G_{n-1} and G_n, the tag is synchronized with the reader.

Precisely speaking, receiving a query, the target tag first verifies if the query contains G_n as shown in Fig. 6.3. If it can verify G_n, it generates G_{n+1} and G_{n+2}, while sending them back in its response. Meanwhile, it updates the stored G_{n-1} and G_n into G_n and G_{n+1}.

If the query does not contain G_n, the tag does not discard the query, instead, it keeps on checking if the query contains G_{n-1}. If G_{n-1} is found, the tag sends back its response as

$$\{R_n^2 \oplus G_n, R_n^3 \oplus G_{n+1}\} \ , \tag{6.8}$$

but it will not update the stored greeting numbers. (We will explain why the tag should check for G_{n-1} later.)

If the query contains neither G_n nor G_{n-1}, the target tag determines that is a replayed query, so discards the query. Obviously, a tag other than the target tag cannot verify its own G_n nor G_{n-1} from the query, so it will not respond.

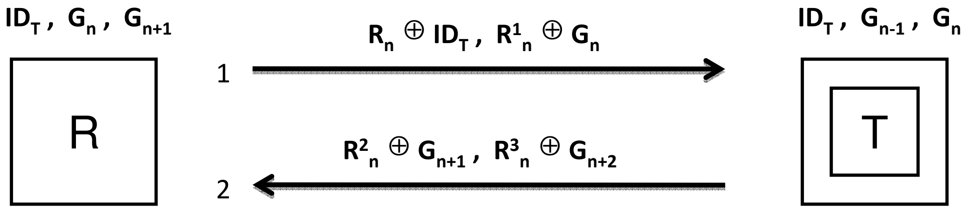

Figure 6.3 The procedure of the multi-response protocol.

Receiving a response as shown in Fig. 6.3, the reader derives G_{n+1} and G_{n+2} from the response and update the greeting number it stores. If the response is like 6.8, the reader does not update, but queries the tag again by using its current greeting G_n. Only when the reader receives an expected response, can it determine that the tag is present.

Now, we answer why we need store G_{n-1} in the tag. Actually, without G_{n-1}, the tag may be desynchronized with the reader. For example, an adversary can block a response from the target tag to the reader, after the tag receives a valid query containing G_n. In this case, the tag will update its G_n into G_{n+1}, but the reader still hold G_n. So, the greeting stored in the tag is one step ahead of that of the reader. When the tag also stores G_{n-1}, then when it updates G_n into G_{n+1}, it also update G_{n-1} into G_n. Thus, even under attacks, the tag can still use G_n to keep synchronized with the reader.

One may ask if storing G_{n-1} is enough. Do we need to store more greeting numbers? The answer is no. By carefully analyzing the desynchronization attacks, we found that the adversary can cheat the target tag to update its greeting number only once. After the tag updates G_n into G_{n+1} due to attacks, the adversary have no way to obtain a search query containing G_{n+1} to make the tag updating again, because they cannot generate G_{n+1} themselves, and the reader still holds G_n and hence will not generate a query containing G_{n+1}.

Compared to the basic protocol, the synchronization-based one mitigates the impact of replaying attacks by reducing the number of replayed query that the tag will respond. In the basic one, the adversary can replay any previous query. But in the synchronization-based one, he can replay only the last query, because the tag will discard other replayed queries.

6.3.4 A Multi-response Protocol

Our basic protocol and the synchronized one bear the same feature: *there is only one response for each query if the target tag is present.* Or, these two protocols are single-response one, because there exists a one-to-one mapping between query and response. So, an adversary can learn the existence of a target tag by observing this one-to-one mapping, even he does not know which tag the target is. If we can disrupt this one-to-one mapping, the adversary could not learn anything. This inspires us to design a multi-response protocol, which adopts multiple responses for each query.

We can easily convert the basic protocol into a multi-response one. That is, for all the non-target tags, even they cannot verify the shared key K from the query, they still generate a fake response. Now, the reader should receive a lot of responses for its query. However, among these response, only one is correct and can be detect by the reader. Other fake responses will be directly discarded by the reader. Meanwhile, the adversary cannot decrypt any response, so he does not if a target tag is present or not.

The drawback is there might be too many responses, which incurs high computation overhead and increases search delay at the reader. We can improve this protocol by reducing the number of responses. For example, each tag could maintain a predefined probability and decide to generate a fake response based on this probability. If we set the probability as 50%, we can reduce the number of responses by half.

In summary, we propose three lightweight secure search protocols, all of which can prevent the adversary from learning the identity of tags or impersonating tags. In the basic protocol, the target tag responds to any query, so an adversary may replay any previous query and know the presence of a target tag. The synchronization-based one mitigates the impact of replaying attacks by reducing the number of queries that a target tag should respond. The best one is the multi-response protocol from which the adversary learns nothing about the target tag.

CHAPTER 7. Experimental Results

As part of the research work, we have conducted a series of experiments to test the feasibility of our solutions. This chapter will focus on these results.

7.0.5 Setup

We evaluate the performance of our protocols in experiments. In particular, we implement 64-bit LFSR and PUF using Cyclone II FPGAs on Altera DE2 boards. The FPGA is programmed using Verilog through Altera's Quartus II 8.0 IDE. To retrieve the outputs from the chip, both the onboard 7-segment hexadecimal LED display and the serial port of the DE2 board are used. To send input data across the serial interface, we take advantage of the open source package *pyserial* in Python 2.5 programming environment.

Our experiments consist of two parts. In the first part, we verify the unclonability and randomness of PUF in our implementation. We construct the same Multiplexor PUF (MPUF) circuit in two devices and study the hamming distance between the outputs of the same device and those from two devices. In the second part, we study the efficiency of our protocols in terms of protocol processing time and the number of gates. We test both Butterfly PUF (BPUF) and Multiplexor PUF (MPUF) for our protocols. The BPUF uses the variety of stable values of latches as the basis, while the MPUF is an arbiter based design.

7.0.6 The Unclonability of PUF

The unclonability of PUF is extremely important, because our protocols extensively utilize PUF to verify the identity of tags and to defeat physical attacks. In this experiment, we implement the same MPUF circuit on two Altera DE2 boards, which we call device 1 and device 2. (Note: the reason why we choose MPUF is for comparison with the results in (17), which also implements MPUF.) Then, we give the same input to both devices and compare the hamming distance between their outputs.

The results that we call *between-class hamming distances*, are shown in TABLE 7.1. (Note: we only show 32-bit results in the table, because there are only 8 7-segment LED displays on our DE2 board, but our actual implementation of PUF is still 64-bit.) The average of the between-class hamming distances is 9.2 bits, which translates to $\frac{9.2}{32} = 28.75\%$. This means the outputs of these two PUFs are 28.75% different from each other, which corresponds well with the result in (17), i.e., a between-class variation of 23%. We conclude that our implementation of PUF is unclonable and suitable for our protocols.

7.0.7 The Randomness of PUF

In our protocols, each target tag generates a sequence of greeting numbers $G_2, \cdots, G_n$ from the first one G_1. Since these greeting numbers are used to prove the identify of tag, we hope they are random and unpredictable so that the adversaries cannot easily guess them and impersonate tags. In this experiment, we give each device just one input and let them generate a sequence of output. Then, we calculate the hamming distance between each pair among the outputs of the same device. Our purpose is to see how random that our implemented PUF could be.

TABLE 7.2 shows the outputs of each device given the same input 91:D4:4B:29 and we also plot the differences among the outputs of each device in Fig. 7.1, in which the y-axis is the number of pairs of each device's outputs that have a specific hamming distance. We call these results shown in Fig. 7.1 *within-class hamming distances*, where we can see that most of pairs have a distance between 10 and 17, and the average distance for device 1 and 2 is 13.8 and 15.7, respectively. These averages are close to 16, which is half of 32 bits. This demonstrates that our implemented PUF produces values that have a difference of about 16 bits on average, which is sufficiently random and suitable for the purpose of tag identification.

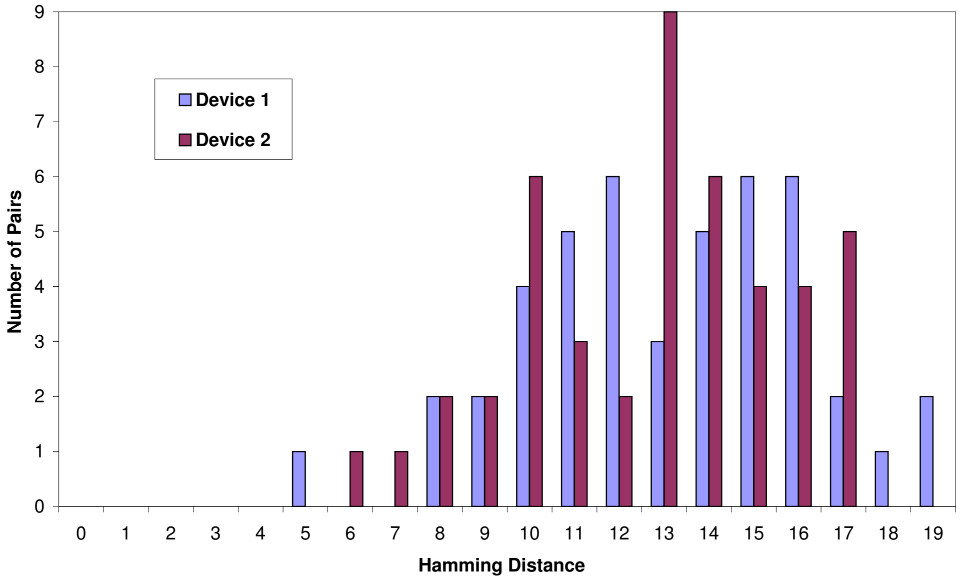

Figure 7.1 The hamming distance within the outputs of each device.

7.0.8 Efficiency of our protocols

Our protocols are designed for low-cost RFID tags, whose computation resources are extremely constrained. For example, they have no internal power and can accommodate at most 2000 hardware gates for security purposes (3). Thus, our protocols must be very efficient for these resource-constrained tags. In these experiments, we study the efficiency of our protocols in terms of the protocol processing time and the number of gates. We implement both BPUF and MPUF for our protocols and our experimental results show that both of them are efficient and suitable for low-cost RFID systems.

TABLE 7.3 shows the processing time of our protocols using BPUF and MPUF. In our implementation, each LFSR operation takes 20 ns, while each BPUF operation needs 11 ns and MPUF 57 ns. To process a search query, our protocols require two LFSR operations. (Actually the synchronization-based one needs only one LFSR, but we ignore it here for simplicity.) So, the processing time is 40 ns. To generate one response, a tag needs two LFSR and two PUF operations, which consume 62 with BPUF implementation and 154 ns with MPUF implementation. So, the processing time for our protocols is 102 ns with BPUF and 194 ns with MPUF. Although BPUF is faster than MPUF, both of them are sufficient for our protocols.

TABLE 7.4 shows the number of gates that our protocols require using BPUF and MPUF. In our implementation, each LFSR circuit consumes 300 gates, while the implementation of BPUF and MPUF needs 1088 and 320 gates. Totally, our protocols require 1388 gates with BPUF and 620 gates with MPUF. Considering the processing time results, we can see that although BPUF is faster than MPUF, it is more complicated in implementation and consumes much more gates. However, no matter whether we choose BPUF and MPUF, our protocols can be accommodated in low-cost tags with at most 2000 gates. Recall that a MD5 or SHA-256 hash function requires 8000 to 10000 gates for implementation (4), our protocols are much more efficient for low-cost RFID systems than Li's solutions (15).

Table 7.1 The hamming distance between the outputs of two devices

Input	Device 1	Device 2	Distance
91:D4:4B:29	20:80:56:70	20:80:54:70	1
92:00:11:EB	7F:E3:73:FF	00:00:32:20	21
AC:4D:BB:D7	3F:5F:7F:3E	36:0B:5E:3C	8
E8:53:01:98	B4:6D:17:DC	84:45:17:C1	8
0A:23:72:D1	64:F5:B2:B2	44:74:80:12	8
7F:A2:99:DA	79:FE:F5:73	60:EA:C1:51	10
19:F3:20:E7	B8:AF:BB:3A	A0:A8:8B:18	9
3D:83:AA:DC	F5:EF:FF:D7	F7:E3:5D:F5	8
29:EC:25:73	DE:EB:ED:BD	CC:C9:C9:5C	10
65:FD:85:81	FE:F7:DE:F5	3A:E6:CE:C1	9

Table 7.2 The outputs of two devices given the same input 91:D4:4B:29

Round	Device 1	Device 2
1	74:EC:54:FF	20:80:54:70
2	75:96:7E:3B	CC:70:05:08
3	55:55:7F:59	E7:08:31:14
4	FE:31:DD:5B	97:A5:F7:DF
5	E7:BF:79:7B	00:7B:AF:94
6	B6:B2:DE:2A	3E:21:59:DC
7	E0:57:F6:3C	89:14:07:38
8	DC:30:31:68	76:83:12:20
9	F4:F7:F7:F7	9F:F3:57:EF
10	FE:EF:D9:C9	10:00:56:20

Table 7.3 The processing time of our protocols

	Processing time (ns)	
LFSR	20	
PUF	11 (BPUF)	57 (MPUF)
Query	40	
Response	62	154
Protocol	102	194

Table 7.4 The number of gates consumed in our protocols

	Number of gates	
LFSR	300	
PUF	1088 (BPUF)	320 (MPUF)
Protocol	1388	620

CHAPTER 8. Summary

8.1 Conclusion

In this thesis, we have presented several solutions to some of the pressing security issues facing RFID systems.

We have shown that it is possible to provide mutual authentication for low-cost RFID tags using minimal hardware such as PUFs and LFSRs. Additionally, we have produced protocols that allow for secure ownership transfer for three-party and two-party systems that utilize the same low cost hardware as our mutual authentication protocol.

We have also studied the secure search problem focusing on low-cost RFID systems. We proposed several lightweight protocols based on LFSRs and PUFs. Our protocols prevent adversaries from learning tag identity and from replaying query and response messages to impersonate RFID reader or tag.

In our own experiments, we were able to confirm the feasibility of our design. Although our tests were conducted in an FPGA environment, an ASIC implementation would perform similarly, as has been confirmed in related work. Our experimental results show that our protocols have negligible processing time and require between 620 and 1400 hardware gates (for 64 bit key length) depending on the choice of PUF design. This is well beneath the 2000 gate threshold that is set by the EPC standard. This shows that they are very suitable for low-cost RFID systems with at most 2000 gates, while existing solutions based on hash functions must consume between 8000 and 10000 gates.

8.2 Future work

Future improvements to our protocols include efforts to remove the possibility of tracking completely, as is possible by a persistent active attacker. A further improvement to our two-party ownership transfer protocol would be to remove the assumption of a secure backward channel.

It would also be desirable to mitigate the worm-hole attacks on our mutual authentication protocols, although we have presented possible solutions to this problem. Finally, we would like to examine our protocol using real ASIC hardware, including PUF and LFSR circuits, to provide more data on the stability and functionality of our protocols.

BIBLIOGRAPHY

[1] S. Kumar, J Guajardo et al., ”Extended Abstract: The Butterfly PUF Protecting IP on every FPGA” , in *Hardware-Oriented Security and Trust (HOST 2008)* , pp.67-70, June 2008.

[2] E. Ozturk, G. Hammouri, B. Sunar, ”Physical unclonable function with tristate buffers” , in *Circuits and Systems, 2008 (ISCAS 2008)*, pp.3194-3197, May 2008.

[3] A. Juels, S. A. Weis, “Authenticating Pervasive Devices with Human Protocols” , in *Advances in Cryptology (CRYPTO 2005)*, pp. 293-308, 2005.

[4] Feldhofer, Martin, and Recberger, “A Case Against Currently Used Hash Functions in RFID Protocols” , in *Printed handout of Workshop on RFID Security (RFIDSec 06)*, 2006.

[5] T. Heydt-Benjamin et al., Vulnerabilities in First-Generation RFID-enabled Credit Cards , in *Financial Cryptography and Data Security, 2008*, pp. 14, 2008.

[6] R. Pappu et al., Physical One-Way Functions, in *Science*, vol. 297, Sep. 2002, pp. 2026-2030.

[7] B. Skoric et al., Information-theoretic analysis of capacitive physical unclonable functions , in *Journal of Applied Physics*, vol. 100, Jul. 2006, pp. 024902-11.

[8] Jorge Guajardo, Sandeep S. Kumar, Klaus Kursawe, Geert-Jan Schrijen, Pim Tuyls, Intrinsic Physical Unclonable Functions in Field Programmable Gate Arrays , in *ISSE 2007*, Sep 25-27, Warsaw, Poland.

[9] R. Maes, P. Tuyls, and I. Verbauwhede, ”Statistical Analysis of Silicon PUF responses for Device Identification” , in *SECSI Workshop 2008*, 2008.

[10] B. Gassend et al., Silicon physical random functions , in *Proceedings of the 9th ACM conference on Computer and communications security*, ACM, 2002, pp. 148-160.

[11] Daihyun Lim et al., Extracting secret keys from integrated circuits , in *Very Large Scale Integration (VLSI) Systems*, IEEE Transactions on, vol. 13, 2005, pp. 1200-1205.

[12] Peris-Lopez, Hernandez-Castro,Estevez Tapiador, and Ribagorda, “LMAP: A Real Lightweight Mutual Authentication Protocol for Low-cost RFID tags” , in *Printed handout of Workshop on RFID Security (RFIDSec 06)*, 2006.

[13] Konidala and Kim, “RFID Tag-Reader Mutual Authentication Scheme Utilizing Tags Access Password” , in *Auto-ID Labs White Paper WP-HARDWARE-033*, 2007.

[14] N. J. Hopper and M. Blum,, “Secure Human Identification Protocols” , in *Advances in Cryptology - ASIACRYPT 2001*, LNCS, volume 2248, pages 52-66, 2001.

[15] Chiu C. Tan, Bo Sheng, and Qun Li, “Serverless Search and Authentication Protocols for RFID” , in *IEEE Percom*, pages 3-12, 2007.

[16] L. Bolotnyy and G. Robins, “Physically Unclonable Function -Based Security and Privacy in RFID Systems” , in *Proc. IEEE International Conference on Pervasive Computing and Communications (PerCom 2007)*, pp. 211-218, March, 2007.

[17] G. E. Suh, S. Devadas, ”Physical Unclonable Functions for Device Authentication and Secret Key Generation” , in *Design Automation Conference - DAC 2007*, 2007.

[18] K. Osaka, T. Takagi, K. Yamazaki and O. Takahashi, ”An Efficient and Secure RFID Security Method with Ownership Transfer” , in *Proceedings of the International Conference on Computational Intelligence and Security (CIS)*, LNAI 4456, pp. 778787, 2007.

[19] D. Chinnery, K. Keutzer, ”Closing the Gap Between ASIC and Custom: An ASIC Perspective” , in *Design Automation Conference*, 637-642, June, 2000.

[20] S. A. Weis, S. E. Sarma, R. L. Rivest, D. W. Engels, ”Security and Privacy Aspects of Low-Cost Radio Frequency Identi?cation Systems” , in *Security in Pervasive Computing*, LNCS, vol. 2802, pp. 201-212, 2003.

[21] M. Ohkubo, K. Suzuki, and S. Kinoshita, ”Cryptographic Approach to Privacy-Friendly Tags” , in *RFID Privacy Workshop 2003*, 2003.

[22] D. Henrici, and P. Müller, ”Hash-based Enhancement of Location Privacy for Radio-Frequency Identi?cation Devices using Varying Identifiers” , in *Proceedings of PerSec04*,IEEE PerCom, pp.149-153, 2004.

[23] D. Henrici, and P. Müller, ”Providing Security and Privacy in RFID Systems Using Triggered Hash Chains” , in PerCom’08, 50-59, 2008.

[24] G. Tsudik. “A family of dunces: Trivial RFID identification and authentication protocols” , in *PET 2007*, volume 4776 of LNCS, pp 45-61, 2007.

[25] M. Burmester, T. v. Le, and B. d. Medeiros, ”Provably secure ubiquitous systems: Universally composable RFID authentication protocols” , in *Conference on Security and Privacy for Emerging Areas in Communication Networks (SecureComm06)*, 2006.

[26] M. Conti, R. Di Pietro, L. Vincenzo Mancini, ”RIPP-FS: an RFID Identi?cation, Privacy Preserving protocol with Forward Secrecy” , in PerCom’07, 2007.

[27] Bogdanov, Andrey and Leander, Gregor and Paar, Christof and Poschmann, Axel and Robshaw, Matt J.B. and Seurin, Yannick, ”Hash Functions and RFID Tags : Mind The Gap” , in *Proceedings of the 10th International Workshop Cryptographic Hardware and Embedded Systems (CHES 2008)*, LNCS, vol. 5154, 2008.

[28] J. Bringer and H. Chabanne and E. Dottax, ”HB++: a Lightweight Authentication Protocol Secure against Some Attacks” , in *Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU)*, 2006.

[29] Henri Gilbert and Matt Robshaw and Herve Sibert, ”An Active Attack Against HB+ - A Provably Secure Lightweight Authentication Protocol” , in *IEEE Electronic Letters* 41, 21, pp. 1169-1170, 2005.

[30] S. Piramuthu, ”HB and Related Lightweight Authentication Protocols for Secure RFID Tag/Reader Authentication Title” , in *CollECTeR Europe Conference*, 2006.

[31] G. Hammouri and B. Sunar, ”PUF-HB: A Tamper-Resilient HB Based Authentication Protocol” , in ACNS 2008.

[32] P. Peris-Lopez, J. C. Hernandez-Castro, J. Estevez-Tapiador, and A. Ribagorda, ”M2AP: A minimalist mutual-authentication protocol for low-cost RFID tags” , in *International Conference on Ubiquitous Intelligence and Computing (UIC06)*, vol. 4159 of LNCS, pp. 912923. 2006.

[33] P. Peris-Lopez, J. C. Hernandez-Castro, J. Estevez-Tapiador, and A. Ribagorda, ”EMAP: An Ef?cient Mutual-Authentication Protocol for Low-cost RFID tags” , in *OTM Federated Conferences and Workshop: IS Workshop*, 2006.

[34] T. Li and G. Wang, ”Security analysis of two ultra-lightweight RFID authentication protocols” , in IFIP SEC, 2007.

[35] M. Bárász, B. Boros, P. Ligeti, K. Lója, and D. A. Nagy, ”Breaking LMAP” , in *Printed handout of Workshop on RFID Security (RFIDSec 07)*, 2007.

[36] M. Bárász, B. Boros, P. Ligeti, K. Lója, and D. A. Nagy, ”Passive Attack Against the M2AP Mutual Authentication Protocol for RFID Tags” , in RFID 2007.

[37] T. Li, R. Deng, ”Vulnerability Analysis of EMAP - An Efficient RFID Mutual Authentication Protocol” , in *The International Conference on Availability, Reliability, and Security Bridging Theory and Practice*, 2007.

[38] M. Bárász, B. Boros, P. Ligeti, K. Lója, and D. A. Nagy, ”Breaking EMAP” , in SecureComm 2007.

[39] D.M. Konidala, Z. Kim, and K. Kim, ”A Simple and Cost-e?ective RFID Tag-Reader Mutual Authentication Scheme” , in *Proceedings of Intl Conference on RFID Security 2007 (RFIDSec 07)*, pp. 141-152, 2007.

[40] T.L. Lim, and T. Li, ”Addressing theWeakness in a Lightweight RFID Tag-Reader Mutual Authentication Scheme” , in *Proceedings of the IEEE Intl Global Telecommunications Conference (GLOBECOM 07)*, pp. 59-63, 2007.

[41] P. Peris-Lopez, J. C. Hernandez-Castro, J. Estevez-Tapiador, T. Li, and L. Tong Lee, ”Vulnerability Analysis of a Mutual Authentication Scheme under the EPC Class-1 Generation-2 Standard” , in RFIDSec08, 2008.

[42] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, ”Handbook of Applied Cryptography” , Chapter 6.2.1.

[43] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, ”Handbook of Applied Cryptography” , Chapter 4.5.3.

[44] K. Koralalage, S. Reza, J. Miura, Y. Goto and J. Cheng, ”POP Method: An Approach to Enhance the Security and privacy of RFID Systems Used in Product Lifecycle with an Anonymous Ownership Transferring Mechanism” , in *Proceedings of the 2007 ACM Symposium on Applied Computing*, pp. 270275, 2007.

[45] S. Fouladgar and H. Afifi, ”A Simple Delegation Scheme for RFID Systems (SiDeS)” , in *Proceedings of the IEEE International Conference on RFID*, 2007.

[46] J. Saito, K. Imamoto, and K. Sakurai, ”Reassignment Scheme of an RFID Tags Key for Owner Transfer” , in *Proceedings of the IFIP International Conference on Embedded and Ubiquitous Computing (EUC) Workshop*, LNCS vol. 3823, pp. 13031312, 2005.

[47] Y. Seo, T. Asano, H. Lee and K. Kim, ”A Lightweight Protocol Enabling Ownership Transfer and Granular Data Access of RFID Tags” , in *Proceedings of the 2007 Symposium on Cryptography and Information Security (SCIS)*, 2007.

[48] R. Ward, T. Molteno, ”Table of Linear Feedback Shift Registers” , Online: *http://www.physics.otago.ac.nz/px/research/electronics/papers/technical-reports/lfsr_table.pdf*, Technical Reports, University of Otago, New Zealand.

[49] B. Song, ”RFID Tag Ownership Transfer” , in RFIDSec08, 2008.

[50] D. Molnar, A. Soppera, and D. Wagner. ”A scalable, delegatable pseudonym protocol enabling ownership transfer of RFID tags.” . in *Bart Preneel and Stafford Tavares, editors, Selected Areas in Cryptography (SAC 2005)*, LNCS, vol. 3897, pp. 276290, 2005.

[51] C. Shannon, ”Communication Theory of Secrecy Systems” , in *Bell System Technical Journal*, vol. 28(4), pp. 656-715, 1949.

[52] S. Lemieux and A. Tang, ”Clone Resistant Mutual Authentication for Low-Cost RFID Technology” , in *Cryptology ePrint Archive*, Report 2007/170, 2007.

[53] H. Lee and Choi, Young and Lee, Su-Mi and Lee, Dong Hoon ”Trapdoor-Based Mutual Authentication Scheme without Cryptographic Primitives in RFID Tags” , in *Third International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU)*, 2007.

[54] Hitachi Product Specification sheet, URL: http://www.hitachi.co.jp/Prod/mu-chip/index.html, 2007.

[55] Aeroscout Product Description, URL: http://www.aeroscout.com/content/tags, 2007.