A Thesis for the Degree of Master of Science

Mutual Authentication of RFID System using Synchronized Secret Information

Sangshin Lee

School of Engineering

Information and Communications University

2006

Mutual Authentication of RFID System using Synchronized Secret Information

Mutual Authentication of RFID System using Synchronized Secret Information

Advisor : Professor Kwangjo Kim

by

Sangshin Lee
School of Engineering
Information and Communications University

A thesis submitted to the faculty of Information and Communications University in partial fulfillment of the requirements for the degree of Master of Science in the School of Engineering

Daejeon, Korea
Dec. 28. 2005
Approved by

(signed)

Professor Kwangjo Kim
Major Advisor

Mutual Authentication of RFID System using Synchronized Secret Information

Sangshin Lee

We certify that this work has passed the scholastic standards required by Information and Communications University as a thesis for the degree of Master of Science

Dec. 28. 2005

Approved:

Chairman of the Committee
Kwangjo Kim, Professor
School of Engineering

Committee Member
Jae Choon Cha, Assistant Professor
School of Engineering

Committee Member
HoWon Kim, Ph.D
Electronics Telecommunications Research Institute (ETRI)

M.S. Sangshin Lee
20042038

Mutual Authentication of RFID System using Synchronized Secret Information

School of Engineering, 2006, 43p.
Major Advisor : Prof. Kwangjo Kim.
Text in English

Abstract

Radio Frequency IDentification (RFID) is an automatic identification system, relying on storing and remotely retrieving data about objects we want to manage using devices called "RFID tag". The RFID system is more useful for various purposes than optical barcode technology since the RFID system can identify lots of tags quickly through RF with neither physical nor visual contact. The RFID system can be used in lots of industries such as supply chain management, inventory, storage, *etc.* and give facilities for individuals with a ubiquitous computing environment.

However, RFID system can have security problems inherently if the tag offers no access-control and tamper-resistance mechanisms. RFID system can induce an information leakage problem of companies and privacy problems of individuals since the RFID tag emits its data to everyone including adversaries. For example, a dishonest company may try to collect information of competing company about physical distribution. By utilizing responses from a tag, an adversary may try to get knowledge of products which an individual user carries or traces a user. In addition, we must consider an attack that an adversary earns unfair profits by responding a reader's query with forged information. These vulnerabilities make people reluctant to use RFID

technology [2, 23].

Even though there are many cryptographic primitives against similar vulnerabilities, they can not be applied to the RFID system due to the limited computation power of a low-cost tag. Consequently, new security protocols with less calculation in the tag are required. To protect users from tracing, we propose an RFID mutual authentication scheme which utilizes a hash function and synchronized secret information like others [11, 9, 17, 16]. To the best of our knowledge, our scheme offers the most enhanced security feature in RFID mutual authentication scheme with respect to user privacy allowing one more hash operation in comparison with [17] which requires identical computational complexity at a back-end server.

Contents

List of Tables

List of Figures

List of Abbreviations

CRHF Collision Resistant Hash Function

DB Database

DoS Denial-of-Service

EEPROM Electrical Erasable Programmable Read Only Memory

IC Integrated Circuit

OWHF One-Way Hash Function

PRNG PseudoRandom Number Generator

RFID Radio Frequency Identification

RNG Random Number Generator

TTP Trusted Third Party

XOR Exclusive-or

List of Notations

$\mathcal{T}$ RF tag, or transponder.

$\mathcal{R}$ RF tag reader, or transceiver.

$\mathcal{B}$ Back-end server, which has a database.

D A database of $\mathcal{B}$.

$\mathcal{A}$ An adversary.

$h()$ One-way hash function.

$PRNG$ PseudoRandom Number Generator.

$\oplus$ Exclusive-or (XOR) function.

ID Identification value of $\mathcal{T}$.

k Secret key shared between $\mathcal{R}$ and $\mathcal{B}$.

r_1 Pseudorandom number generated by $PRNG$ of $\mathcal{T}$.

r_2 Hashed value to authenticate $\mathcal{T}$ to $\mathcal{B}$.

r_3 Hashed value to authenticate $\mathcal{B}$ to $\mathcal{T}$.

s Pseudorandom number generated by $PRNG$ of $\mathcal{R}$.

r_b Pseudorandom number generated by $PRNG$ of $\mathcal{B}$.

r_r Pseudorandom number generated by $PRNG$ of $\mathcal{R}$.

r_t Pseudorandom number generated by $PRNG$ of $\mathcal{T}$.

$\stackrel{?}{=}$ Verification operator to check whether the left hand side is valid for the right side hand or not.

$\leftarrow$ Update operator from the right hand side to the left hand side.

IDF A field for the temporary identification value of $\mathcal{T}$ and used as a primary index.

K A field for the current shared random secret, k.

K_{last} A field for the previous shared random secret, k.

Chapter 1
Introduction

1.1 Radio Frequency Identification

Radio Frequency IDentification (RFID) is an automatic identification system, relying on storing and remotely retrieving data about objects we want to manage using devices called "RFID tag". In the near future, RFID technology is expected to play an important role for object identification as a ubiquitous infrastructure. RFID technology is one of next generation technologies which is mainly used to identify massive objects and will be a substitution for the existing optical barcode system in the near future. The micro-chip equipped on a tag has unique identification information and is applicable for various fields such as animal tracking, supply chain management, inventory control, *etc.*

Some widespread and commonly known applications of RFID are identification, tracking and real-time monitoring. RFID can help in providing real-time supply on location and status of goods. The ability to identify and track assets is critical for a retail store, a wholesale distributor, a manufacturer, or a hospital.

An RFID tag attached to any object contains a unique serial number that is used to identify the object. This application can be used in supply-chain management where each item can be identified and when it enters or leaves the warehouse. RFID can also be used to track the exact location of people or equipment and record events associated with their location.

1.2 Design Considerations on Authentication Protocol

RFID applications are growing in areas such as admission control, payment, ticketing, *etc.* that sophisticated security measures are needed. Without security, illegal activities cheating RFID systems, *e.g.*, breaking into a building or ticketing without payment, are not difficult because of using air interface between tags and readers of RFID system. In addition, user privacy is also issued since anyone can intercept communication between tags and readers and get information about a tag holder. We describe these security issues in the next chapter in detail.

To remove security vulnerabilities and protect user privacy, authentication protocol for RFID system can be considered as a measure of security. With the well-designed authentication protocol, a tag, a reader, and a back-end server authenticate each other and agree on the secret session key which will be used to secure the later session. As denoted in [3, 12, 13, 18, 26], one of important issues in providing security services under RFID environment is to design the authentication protocol to meet the low computational capabilities and restricted capacities. The detailed processes of our mutual authentication protocol will be described in Chapter 4.

When designing an RFID authentication protocol, we should consider the factors such as the properties of protocol environments and the resources of protocol entities. There are several factors specific to RFID systems. One of them is the characteristics of RFID communication channel. RFID communication channel is asymmetric in signal strength, which means it will be much easier for adversaries to eavesdrop on signals from reader to tag than on data from tag to reader since tags respond by passively modulating a carrier wave broadcasted by the reader. The insecure communication channel based on air interface between tags and reader is more vulnerable to attack than

the secure channel between readers and back-end servers. Another is limited resources to meet the minimum cost of RFID tags for general usage. The low-cost RFID tags are very limited in the computational capability compared with other entities such as readers and back-end servers.

We apply mutual authentication method to the RFID system in order to update secret information which is shared between a back-end and a tag. If secret information is fixed, an adversary may trace a previous event which a tag is participated by tapering with the tag. Therefore, the secret information in the tag must be frequently changed and synchronized with a database of the back-end. In order to update secret information securely with recognition of the back-end and the tag, mutual authentication is essential.

The cost of a RFID tag should be reduced under US$0.50 in practical use. In order to achieve this price, IC should be priced less than US$0.20 [27]. These price barrier for low-cost tags restrict the range of gates in a tag from 7.5 to 15 K and the number of gates for security purpose is limited to from 2.5 to 5 K [22]. For these reasons, it must be impractical to use the existing cryptographic algorithms [12].

1.3 Our Contributions

Recently, RFID applications have been implemented for various areas and the technology is also improved. Security concerns and user privacy issues are potential risks for RFID proliferation. In the mean time, security problems are not considered as a big barrier in the real world. The reason is that the applications based on RFID are mainly developed for supply chain management, and user friendly tags just start to emerge, which means the cost of RFID tag makes it difficult to apply as a substitute for the existing barcode. However, RFID technology is growing very fast, and the cost of tag for general purposes will be reasonable in the near future. Privacy and security issues should be protected for the admirable usage of RFID, and several

papers proposed security and privacy protections schemes for RFID.

In this thesis, some schemes for security and privacy protection for low-cost RFID are surveyed and their pros and cons are compared for the security requirements of RFID. Further, we propose a robust privacy preserving mutual authentication protocol that fits the low-cost RFID system environment. The proposed authentication protocol meets the privacy protection for a tag holder, which requires confidentiality, untraceability, anti-cloning, and integrity from the cryptographic point of view.

Therefore, we provide a protocol which can be realized in a low-cost tag. To protect users from tracing, we propose an RFID mutual authentication scheme which utilizes a hash function and synchronized secret information like others [11, 9, 17, 16]. We will show our scheme offers the most enhanced security feature in RFID mutual authentication scheme with respect to user privacy including resistance against tag cloning allowing one more hash operation in comparison with [17] which requires identical computational complexity at a back-end server.

1.4 Organization

The remainder of the thesis is organized as follows:

In Chapter 2, we introduce RFID system primer, cryptographic background to understand our scheme. We discuss about the current security problems concerning the characteristics on RFID technology as well as the potential privacy issues.

Several schemes and protocols are introduced in Chapter 3. Those schemes are mostly focused on how to guarantee security and protect user privacy in low-cost RFID environment. We analyze security problems of each protocol.

We propose our mutual authentication scheme in Chapter 4. To satisfy low computational power on the existing RFID, the proposed authentication protocol adapts simple cryptographic primitives, a one-way hash function

and random number generator. Mutual authentication is the basis for the proposed protocol, provides user privacy protection and protects a tag against various attacks. Then we analyze security and performance of our protocol.

In Chapter 5, we compare security and efficiency of the proposed protocol with previous work.

Finally, we conclude in Chapter 6.

Chapter 2
Preliminaries

2.1 RFID System Components

The RFID system has three main components: a RFID tag $\mathcal{T}$, a reader $\mathcal{R}$, and a back-end server $\mathcal{B}$ [27].

$\mathcal{T}$ carries object identifying data. $\mathcal{T}$ is attached to all objects in an RFID system. $\mathcal{T}$ is typically composed of a microchip for storage and computation, and a coupling element, such as an antenna coil for communication. $\mathcal{T}$ may also contain a contact pad. Tag memory may be read-only, write-once read-many or fully rewritable.

$\mathcal{R}$ not only queries $\mathcal{T}$ for its data, but also updates the contents of $\mathcal{T}$ through an RF interface. To provide additional functionality, $\mathcal{R}$ may contain internal storage, processing power or connections to $\mathcal{B}$. Computation, such as cryptographic calculations, may be performed by $\mathcal{R}$ on behalf of $\mathcal{T}$.

$\mathcal{B}$ stores records associated with $\mathcal{T}$. $\mathcal{R}$ may use contents of $\mathcal{T}$ as check information to find ID of $\mathcal{T}$. $\mathcal{B}$ may associate product information, tracking logs or key management information with a particular $\mathcal{T}$. An independent $\mathcal{B}$ may be built by anyone with access to tag contents. This allows $\mathcal{A}$ along the supply chain to build his own applications.

2.2 RFID Security

2.2.1 Security Problems

Privacy and cloning of $\mathcal{T}$ must be solved for proliferation of RFID technology. Because everyone can query to a low-cost tag (which doesn't have an access control function) without recognition of the tag holder, privacy must be considered.

One of privacy problems is the information leakage on user's belongings. We don't want that our personal things are known to others. For example, exposure of expensive products can make a tag holder be a victim of a robber. A personal medicine known to another throws the user into confusion. Even though the information leakage problem is significant, it's easy to solve. It can be solved just by using the anonymous ID that $\mathcal{B}$ only can match with the real product codes [6].

Another problem about the user privacy is a user tracing problem. By tracing $\mathcal{T}$, $\mathcal{A}$ can chase and identify the user. If $\mathcal{A}$ installs a vast amount of $\mathcal{R}$'s at a wide area, each individual person's location privacy is violated by $\mathcal{A}$. The user tracing problem is hard to solve, because we must update every response of $\mathcal{T}$ in order to evade a pursuer while a legitimate user can identify $\mathcal{T}$ without any inconvenience. Moreover, this job must be performed by $\mathcal{T}$ with small computational power.

Tag cloning also must not be ignored. $\mathcal{A}$ may try to clone a specific $\mathcal{T}$ to gain illegal benefit. For instance, if a protocol which is vulnerable to a replay attack is used, $\mathcal{A}$ can disguise an expensive product as cheap one by saving a response from $\mathcal{T}$ attached on cheap one and emitting the response while checking out.

2.2.2 Security Requirements

The most important security requirement for user privacy is *untraceability* [5]. Untraceability is the property that $\mathcal{A}$ can not trace $\mathcal{T}$ by using interactions with $\mathcal{T}$. This concept includes *ID anonymity*, which is satisfied when $\mathcal{A}$ can not know a real product ID of $\mathcal{T}$ and guarantees to prevent the leakage of information of user belongings. Even thought $\mathcal{A}$ doesn't know ID of $\mathcal{T}$, $\mathcal{A}$ can trace $\mathcal{T}$ if $\mathcal{A}$ can find specific patterns of outputs of $\mathcal{T}$, *e.g.*, a value increased by one for every response in [11].

For perfect untraceability, protocols must satisfy *indistinguishability* [18] and *forward security* [18] (or *forward untraceability* [5]). Indistinguishability means that values emitted by $\mathcal{T}$ must not be discriminated from the other $\mathcal{T}$. For forward security, $\mathcal{A}$ cannot trace the data back through previous events in which $\mathcal{T}$ was involved even if $\mathcal{A}$ acquires the secret data stored in $\mathcal{T}$.

Anti-cloning is an additional security requirement. This property means that $\mathcal{A}$ cannot clone $\mathcal{T}$ without tampering with $\mathcal{T}$. When $\mathcal{A}$ tampers with $\mathcal{T}$, $\mathcal{A}$ has same information as $\mathcal{T}$ itself, and then we cannot prevent tag cloning. However, there are many ways to cloning without tampering with $\mathcal{T}$, *e.g.*, the replay attack. Therefore, *anti-cloning* can be one of security requirements.

Availability is also one of security requirements. $\mathcal{T}$ should be available all the time when a user wants. Even though physical attacks such that jamming responses of $\mathcal{T}$ or blocking responses of $\mathcal{T}$ by using metal mesh or foil cannot be prevented by logical operations, an authentication protocol should provides the data recovery against the data loss such as DoS, message hijacking, power interruption, *etc.* during the authentication processes. Especially, the desynchronization attack by utilizing a man-in-the-middle attack must be prevented.

2.3 Cryptographic Background

2.3.1 Hash Function

The basic operation of hash functions is to map an element of larger domains to an element of smaller domains. This property is utilized in many non-cryptographic computer applications like storage allocation to improve performance. However, cryptographic hash functions (hereinafter, simply hash functions) has more important aspects then conventional ones, which makes them playing a fundamental role in modern cryptography.

The purpose of hash functions in cryptographic sense to provide data integrity and message authentication. For these usage, adopted hash functions $h()$ should satisfy the following requirements:

Compression. Given an input x of arbitrary finite bitlength, $h(x)$ maps to an output y of fixed bitlength n.

One-wayness. If $y = h(x)$ is given, it is computationally infeasible to compute x. This property has two folds. One is *preimage resistance*, which means for all outputs y, it is computationally infeasible to find any input x such that $h(x) = y$ given no corresponding input is known. Another is *2nd-preimage resistance*, which means given x, it is computationally infeasible to find $x' \neq x$ such that $h(x) = h(x')$.

Collision-avoidance. It is computationally infeasible to find a pair (x, x') satisfying $h(x) = h(x')$.

Efficiency. Given an input x, $h(x)$ is easy to compute.

A *one-way hash function* $OWHF$ is a hash function which offers preimage and 2nd preimage resistance. This may be thought of simply as being difficult to invert. A *collision resistant hash function* $CRHF$ is a hash function which is 2nd-preimage resistant and collision-freshness.

Hash functions should be resistant against the *Birthday attack* [25, 1], which is a powerful method to find colliding input pairs. Therefore, it is preferable that the output length of hash function is longer than 160 bits under current computing environments. We assume that hash functions in our protocols are secure and satisfy all above requirements.

The details on collision-freeness and one-wayness of hash functions are appeared in [8]. In terms of low-cost RFID environment, [27] examined and introduced about the study of low-cost hash functions.

Hash Chain

Hash chain is a variant of hash functions and utilized in various areas: RFID [18], authentication [15], micropayment [19] and auction [7], *etc.*

The generation of hash chain is done as follows:

$$\begin{aligned} \text{Seed}: &\quad s_0 \\ \text{1st round}: &\quad h^1 = h(s_0) \\ \text{2nd round}: &\quad h^2 = h(h(s_0)) \\ \ldots: &\quad \ldots \\ \text{n-th round}: &\quad h^n = h(h^{n-1}). \end{aligned}$$

The use of hash chain values is in reverse order, *i.e.* from h^n to h^1. From the one-wayness of hash functions, no one can predict the next value from the current value except only one has the knowledge on the seed.

2.3.2 Random Number Generator

Random number generation is used in a wide variety of cryptographic operations, such as key generation and challenge/response protocols. A random number generator is a function which outputs a sequence of 0 and 1 such that at any point, the next bit cannot be predicted based on the previous bits.

However, true random number generation is difficult to do on a computer, since all the computers are deterministic. Thus, if the same random generator is run twice, identical results are received.

True random number generators are in use, but they can be difficult to build. They typically take input from something in the physical world, such as the rate of neutron emission from a radioactive substance or a user's idle mouse movements. Because of these difficulties, random number generation on a computer is usually only pseudo-random number generation. A pseudo-random number generator $PRNG$ produces a sequence of bits which has a random-looking distribution. With each different seed (a typically random stream of bits used to generate a usually longer pseudo-random stream), the pseudo-random number generator generates a different pseudo-random sequence. With a relatively small random seed a pseudo-random number generator can produce a long apparently random string. Pseudo-random number generators are often based on cryptographic functions like block ciphers or stream ciphers. For instance, iterated DES encryption starting with a 56-bit seed produces a pseudo-random sequence.

2.3.3 Mutual Authentication

The general setting for an identification protocol involves a *prover* A and a *verifier* B. The verifier is presented with, or presumes beforehand, the purported identity of the claimant. The goal is to corroborate that the identity of the claimant is indeed A, *i.e.*, to provide entity authentication.

Definition *Entity authentication* is the process whereby one party is assured (through acquisition of corroborative evidence) of the identity of a second party involved in a protocol, and that the second has actually participated.

Challenge-response identification The idea of cryptographic challenge-response protocols is that one entity (the claimant) "proves" its identity to

another entity (the verifier) by demonstrating knowledge of a secret known to be associated with that entity, without revealing the secret itself to the verifier during the protocol. This is done by providing a response to a time-variant challenge, where the response depends on both the entity's secret and the challenge. The *challenge* is typically a number chosen by one entity (randomly and secretly) at the outset of the protocol. If the communications line is monitored, the response from one execution of the identification protocol should not provide an adversary with useful information for a subsequent identification, as subsequent challenges will differ.

Challenge-response by symmetric-key techniques Challenge-response mechanisms based on symmetric-key techniques require the claimant and the verifier to share a symmetric key. For closed systems with a small number of users, each pair of users may share a key. In larger systems employing symmetric-key techniques, identification protocols often involve the use of a trusted on-line server with which each party shares a key. The on-line server effectively acts like the hub of a wheel, providing a common session key to two parties each time one requests authentication with the other.

Challenge-response based on (keyed) one-way functions Challenge-response identification can be performed by utilizing a one-way function (not encryption function) with a shared key and challenge. This may be preferable in situations where encryption algorithms are otherwise unavailable or undesirable (*e.g.*, due to export restrictions or computational costs). The revised three-pass challenge-response mechanism based on a keyed one-way function provides mutual identification as follow:

$$\begin{aligned} A \longleftarrow B: &\quad r_B \\ A \longrightarrow B: &\quad r_A, h_K(r_A, r_B, B) \\ A \longleftarrow B: &\quad h_K(r_B, r_A, A) \end{aligned}$$

Chapter 3
Previous Work

We describe some mutual authentication schemes and other approaches for protecting tag holders against an adversary $\mathcal{A}$ in this chapter.

3.1 Mutual Authentication based on a Hash Function

In this section, we describe schemes based on a hash function and a pseudo-random number generator for mutual authentication of the RFID system. After showing authentication processes, we analyze security of the schemes. All schemes in this section assume the channel between $\mathcal{B}$ and $\mathcal{R}$ is secure, and $\mathcal{B}$ and $\mathcal{T}$ own information jointly. In addition, we asume that desynchronization occurred by data loss can be easily prevented by maintaining two records for each $\mathcal{T}$ like [11]. Notations r_b, r_r, and r_t mean random numbers generated by $\mathcal{B}$, $\mathcal{R}$, and $\mathcal{T}$, respectively.

3.1.1 Hash-based Enhancement of Location Privacy

Henrici *et al.* suggested a mutual authentication scheme that utilizes a counter for synchronization between $\mathcal{B}$ and $\mathcal{T}$ [11]. When the system is launched, $\mathcal{T}$ contains its current identifier ID, the current session number k (both are set up with random values), and k_{succ} which is equal to k. $\mathcal{B}$ contains ID and k_{succ} for each $\mathcal{T}$ it manages, which is initially equal to the

values stored in $\mathcal{T}$. An identification can be executed follows (Figure 3.1):

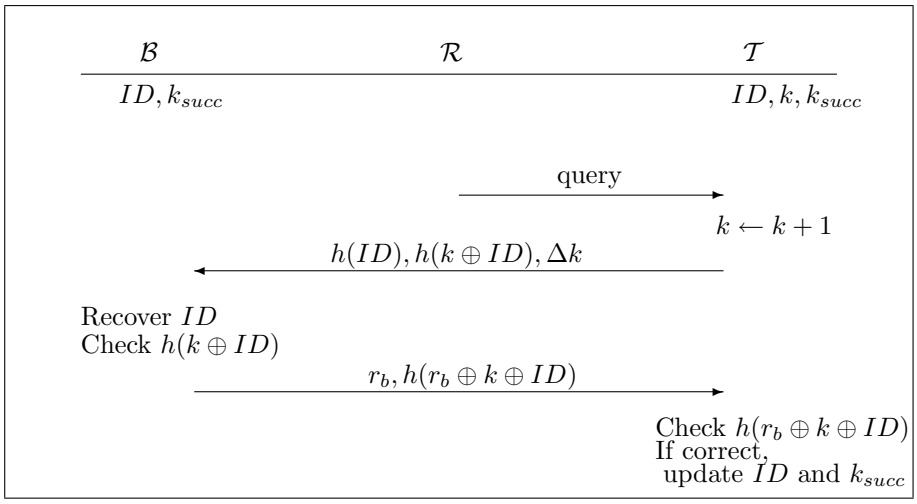

Figure 3.1: Hash based enhancement of location privacy

1. $\mathcal{R}$ sends a request to $\mathcal{T}$.

2. $\mathcal{T}$ increases its current session number k by one and then sends back $h(ID)$, $h(k \oplus ID)$ and $\Delta k := k - k_{succ}$. $h(ID)$ allows $\mathcal{B}$ to recover identity of $\mathcal{T}$; Δk allows $\mathcal{B}$ to recover k and thus to compute $h(k \oplus ID)$, and $h(k \oplus ID)$ aims at thwarting replay attacks.

3. $\mathcal{B}$ checks the validity of these values according to its recorded data. If all is fine, it sends a random number r_b and $h(r_b \oplus k \oplus ID)$ to $\mathcal{T}$ and stores the new values. Since $\mathcal{T}$ knows k and ID and receives r_b, it can check whether or not $h(r_b \oplus k \oplus ID)$ is correct. If this is case, it replaces its identifier by $r_b \oplus ID$ and k_{succ} by k. Otherwise it does not refresh its identifier.

Attack based on non-random information: This attack consists of tracking $\mathcal{T}$, taking advantage of the information supplied by Δk. Indeed, since $\mathcal{T}$ increases its value k every time it receives a request (Step 2) even if the

identification finally fails, while k_{succ} is updated only when the identification succeeds (Step 3), $\mathcal{A}$ may interrogate $\mathcal{T}$ several times to abnormally increase k and therefore Δk. Thanks to the fact that this value is sent in clear in the second message, $\mathcal{A}$ is then able to recognize its target later according to this value: an abnormally high Δk, *i.e.*, far from the expected Δk when no attack occurs.

Attack based on database desynchronization: A subtle and definitive attack consists of desynchronizing $\mathcal{T}$ and $\mathcal{B}$. For this, $\mathcal{A}$ performs the identification so that the random value r sent by $\mathcal{A}$ is the neutral element of $\oplus$: $\mathcal{A}$ replaces r_b by the null bit-string and replaces $h(r_b \oplus k \oplus ID)$ by $h(k \oplus ID)$ obtained from the second message of the current identification. We have trivially $h(0 \oplus k \oplus ID) = h(k \oplus ID)$. Thus, $\mathcal{T}$ cannot detect the attack. Then it replaces its identifier by $0 \oplus ID$ (which is equal to its "old" identifier) and it updates k_{succ}. In the next identification, $\mathcal{T}$ and $\mathcal{B}$ will be desynchronized, since $\mathcal{T}$ computes the hash value using the "new" k_{succ} whereas $\mathcal{B}$ checks the hash value with the "old" k_{succ} : the test fails and the received message is discarded. Consequently, $\mathcal{B}$ will never send the third message to refresh identifier of $\mathcal{T}$ and $\mathcal{T}$ is definitively traceable.

Attack based on tampering: Because ID is updated by XORing ID with r_b which is transmitted through an air interface, $\mathcal{A}$ can trace the previous output of $\mathcal{T}$. We can image an attack that after $\mathcal{A}$ obtains ID by tampering with $\mathcal{T}$ and has eavesdropped plenty of interactions between $\mathcal{B}$ and various $\mathcal{T}$'s, $\mathcal{A}$ tries to distinguish the response of $\mathcal{T}$ from others. If $\mathcal{A}$ collects interaction of the last authentication, $\mathcal{A}$ can know the previous ID by XORing the current ID with r_b of the interaction and test correctness of ID by hashing ID and comparing it with the first value of the response of $\mathcal{T}$, *i.e.*, $h(ID)$. Therefore, if $\mathcal{A}$ collects all interactions between $\mathcal{R}$ and $\mathcal{T}$ when ID is updated, then $\mathcal{A}$ can trace all previous events of $\mathcal{T}$.

3.1.2 A Lightweight RFID Protocol

Dimitriou proposed a mutual authentication scheme that uses random numbers generated by $\mathcal{R}$ and $\mathcal{T}$ for randomization [9]. An identification works as follows (Figure 3.2):

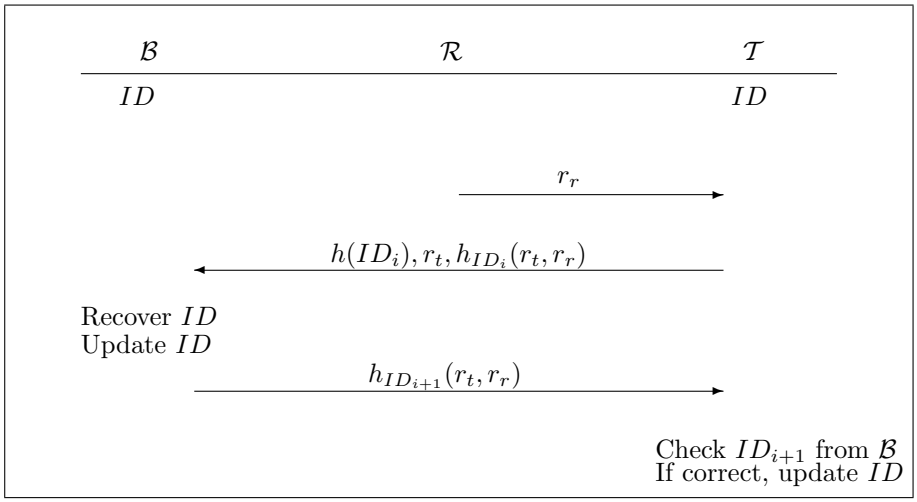

Figure 3.2: A Lightweight RFID Protocol

1. $\mathcal{R}$ transmits a random number r_r.

2. $\mathcal{T}$ generates a new random number r_t and sends back $h(ID_i), r_t$, and $h_{ID_i}(r_t, r_r)$ to $\mathcal{R}$ which then forwards these values to $\mathcal{B}$. $\mathcal{B}$ authenticates $\mathcal{T}$ and if everything holds good, $\mathcal{B}$ computes the new identity, ID_{i+1}.

3. $\mathcal{B}$ constructs the message $h_{ID_{i+1}}(r_t, r_r)$ using the new key ID_{i+1}. $\mathcal{B}$ then sends this message to $\mathcal{R}$ who forwards it to $\mathcal{T}$. Upon reception, $\mathcal{T}$ generates the new key on its own and computes the value $h_{ID_{i+1}}(r_t, r_r)$. If the value received is the same as the value computed, $\mathcal{T}$ accepts the response as authentic and only then $\mathcal{T}$ deletes the old key ID_i and r_t from its memory. Otherwise, $\mathcal{T}$ rejects the answer and keeps the old key ID_i.

Attack based on active query: The first value of the response of $\mathcal{T}$, *i.e.*, $h(ID_i)$ is changed per each successive mutual authentication because ID_i is updated, but $h(ID_i)$ does not vary between successive mutual authentications. Therefore, $\mathcal{A}$ can trace $\mathcal{T}$ by actively querying to $\mathcal{T}$.

Attack based on tampering: The authors doesn't describe in the detail way how to update ID. If the way that $\mathcal{A}$ can guess a previous ID from current ID is implemented, then $\mathcal{A}$ can trace the event that $\mathcal{T}$ is participated.

3.1.3 Efficient Authentication for Low-Cost RFID Systems

Lee *et al.* [16] suggested an efficient authentication for low-cost RFID systems. Their scheme prevents a desynchronization of ID between D of $\mathcal{B}$ and $\mathcal{T}$ by maintaining not only current data but also previous data which is updated by the current data. We briefly describe this scheme.

In their scheme, D of $\mathcal{B}$ and $\mathcal{T}$ commonly save ID.

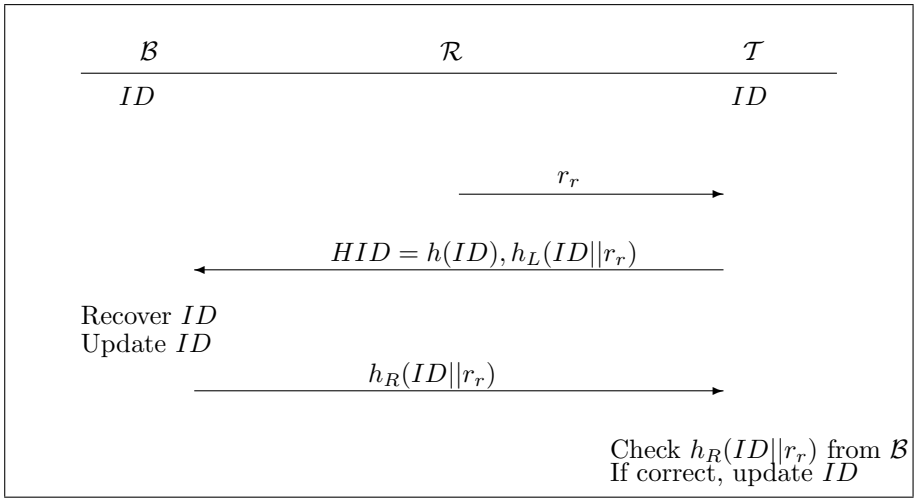

Figure 3.3: Efficient Authentication for Low-Cost RFID Systems

The scheme is operated as follows:

1. $\mathcal{R}$ picks a random number r_r and sends r_r to $\mathcal{T}$.

2. $\mathcal{T}$ computes $HID = h(ID)$ and $h(ID||r_r)$. $\mathcal{T}$ sends $h_L(ID||r_r)$ and HID to $\mathcal{R}$, where $h_L(ID||r_r)$ is a left half of $h(ID||r_r)$.

3. $\mathcal{R}$ sends $h_L(ID||r_r), r_r$, and HID to $\mathcal{B}$.

4. $\mathcal{B}$ finds HID in D. If successful, $\mathcal{B}$ computes $h_R(ID||r_r)$ using r_r received from $\mathcal{R}$ and ID in D, where $h_R(ID||r_r)$ is a right half of $h(ID||r_r)$. $\mathcal{B}$ sends $h_R(ID||r_r)$ to $\mathcal{R}$.

5. To update ID for the next session, $\mathcal{B}$ updates D as $HID = h(ID \oplus r_r)$ and $ID = ID \oplus r_r$.

6. $\mathcal{R}$ forwards $h_R(ID||r_r)$ to $\mathcal{T}$.

7. $\mathcal{T}$ checks a validity of $h_R(ID||r_r)$. If the message is valid, then $\mathcal{T}$ updates its own ID to $ID \oplus r_r$.

Attack based on active query: This scheme has identical venerability with [9]. The first value of the response of $\mathcal{T}$, *i.e.*, $h(ID)$ is changed per each successive mutual authentication because ID_i is updated, but $h(ID)$ does not vary between successive mutual authentications. Therefore, $\mathcal{A}$ can trace $\mathcal{T}$ by actively querying to $\mathcal{T}$.

Attack based on tampering: This scheme also has identical weakness against attack based on tampering like Dimitriou's scheme [9]. Because ID is updated by XORing ID with r_r which is emitted through a air interface, $\mathcal{A}$ can trace the previous output of $\mathcal{T}$. We can image an attack that after $\mathcal{A}$ obtains k by tampering with $\mathcal{T}$ and has eavesdropped plenty of interactions between $\mathcal{B}$ and various $\mathcal{T}$'s, $\mathcal{A}$ tries to distinguish the response of $\mathcal{T}$ from others. If $\mathcal{A}$ collects interaction of the last authentication, $\mathcal{A}$ can know the previous ID by XORing the current ID with r_r of the interaction and test correctness of ID by hashing ID and comparing it with the first value of the

response of $\mathcal{T}$, *i.e.*, HID. Therefore, if $\mathcal{A}$ collects all interactions between $\mathcal{R}$ and $\mathcal{T}$ when ID is updated, then $\mathcal{A}$ can trace all previous events of $\mathcal{T}$.

3.1.4 Privacy and Security in Library RFID

Molnar *et al.*'s scheme [17] provides mutual authentication of $\mathcal{B}$ and $\mathcal{T}$ in a private way if we don't consider tamper with $\mathcal{T}$. It shall prevent an attacker from impersonating, tracing or identifying tags. ID of $\mathcal{T}$ is stored in both D of $\mathcal{B}$ and $\mathcal{T}$. They also share a secret key k.

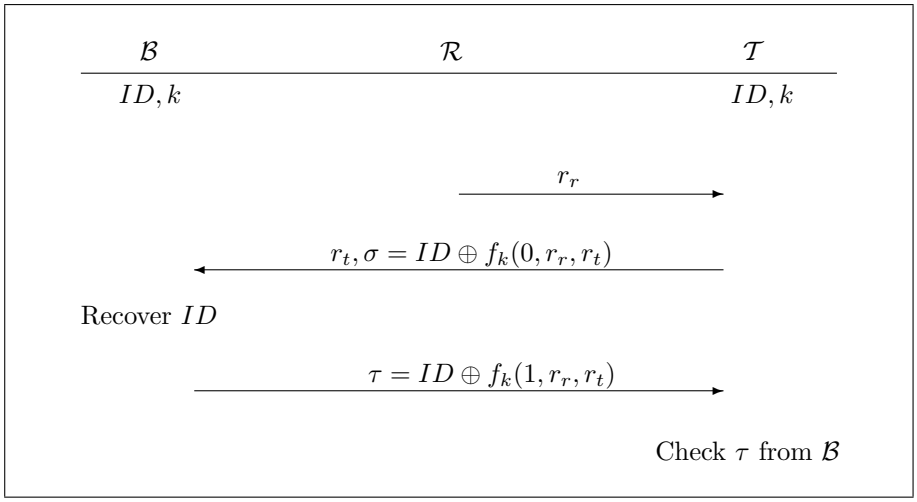

Figure 3.4: Privacy and Security in Library RFID

1. To initiate the authentication, $\mathcal{R}$ sends a random number r_r to $\mathcal{T}$.
2. $\mathcal{T}$ picks a random r_t and answers $\sigma := ID \oplus f_k(0, r_r, r_t)$, where f_k is a pseudorandom function.
3. $\mathcal{B}$ retrieves the identity of $\mathcal{T}$ by finding the pair (ID, k) in D such that $ID = \sigma \oplus f_k(0, r_r, r_t)$. This completes the authentication of $\mathcal{T}$.
4. In order to achieve mutual authentication, $\mathcal{B}$ sends back $\tau := ID \oplus f_k(1, r_r, r_t)$ to $\mathcal{T}$.

5. $\mathcal{T}$ can thus verify the identity of $\mathcal{R}$ by checking that $ID = \tau \oplus f_k(1, r_r, r_t)$.

Attack based on tampering: Because ID and k are not updated all the time, $\mathcal{A}$ can chaise the previous output of $\mathcal{T}$. We can image an attack that after $\mathcal{A}$ obtains k by tampering with $\mathcal{T}$ and has eavesdropped plenty of interactions between $\mathcal{B}$ and various $\mathcal{T}$'s, $\mathcal{A}$ tries to distinguish the response of $\mathcal{T}$ from others. $\mathcal{A}$ can make and test σ' by using the tampered ID and k with values, r_r and r_t, in communication. If σ' generated by $\mathcal{A}$ is the same value as σ, then $\mathcal{A}$ can conclude the tampered $\mathcal{T}$ participates the interaction eavesdropped by $\mathcal{A}$.

3.2 Other Approaches

There exist hardware-based schemes to protect user privacy such that kill command feature [3], blocker-tag [13], and Faraday case. Kill command feature is originally suggested by Auto-ID center [4]. Each $\mathcal{T}$ has a password. When $\mathcal{R}$ orders kill command to $\mathcal{T}$ with its password, $\mathcal{T}$ stops its operation permanently. This feature makes possible perfect security, but we cannot reuse $\mathcal{T}$. $\mathcal{T}$ must be killed when $\mathcal{T}$ goes to an insecure area. After that, $\mathcal{T}$ cannot be operated any more even if $\mathcal{T}$ returns to a secure area. The blocker-tag jams all $\mathcal{T}$ when tree-walking singulation is processed. The blocker-tag also disables legitimate user who wants to collect information from $\mathcal{T}$. The Faraday case prevents $\mathcal{T}$ from hearing the request by enclosing $\mathcal{T}$. This method is only suitable for limited application that we can enclose products using a Faraday case [6].

Several papers suggested schemes relying on the concept of universal re-encryption [10], that re-encryptions of a message m are performed neither requiring nor yielding knowledge of the public key under which m has been encrypted initially. The protocol of Golle *et al.* [10] proposed the concept of universal re-encryption and applied the concept to the RFID system. Saito

et al. [20] pointed out an attack against the protocol of Golle *et al.* [10] and subsequently suggested two protocols based on the Golle *et al.*'s protocol. The first protocol is an improvement of [10] where the operations carried out by $\mathcal{T}$ are modified. The difference between [10] and the second protocol of Saito *et al.* is that the re-encryptions are carried out by $\mathcal{T}$ itself and no longer by $\mathcal{R}$. All schemes based on universal re-encryption are nevertheless weak against eavesdropping. Previous re-encrypted data is the output of $\mathcal{T}$ of the next session, so an eavesdropper can link each session and trace $\mathcal{T}$.

Some schemes use hash functions to identify $\mathcal{T}$. Weis *et al.* [26] suggested a protocol that $\mathcal{T}$ sends $h(ID||r)$ and including a random number r, whenever $\mathcal{R}$ wants to know ID of $\mathcal{T}$. This scheme has a vulnerability that $\mathcal{A}$ can trace the previous outputs of $\mathcal{T}$ if $\mathcal{A}$ tampers with $\mathcal{T}$ [5]. Ohkubo *et al.*'s protocol [18] is a unique protocol which prevents traces by this time. In this protocol, the i-th $\mathcal{T}$ sends $G(H^{k-1}(s_i^1))$ for the k-th response, where G and H are different hash functions and s_i^1 is an initial value of the i-th $\mathcal{T}$. To find ID of $\mathcal{T}$, $\mathcal{B}$ must search all the hash chains of each $\mathcal{T}$, so this protocol is inefficient to be installed in a system which is not small.

Chapter 4
Our Protocol

In this section, we propose a RFID authentication protocol which guarantees mutual authentication in each session and almost secure privacy of a tag holder.

At first, we define some assumptions and the required security goals in RFID mutual authentication protocol. Starting from the design of basic protocol, security and robustness for risks are considered. Then, we place emphasis on the user privacy protection to our proposed protocol.

4.1 Main Idea

Basically, security vulnerabilities in RFID systems come from insecure air interface between $\mathcal{T}$ and $\mathcal{R}$. $\mathcal{A}$ can try to desynchronize identification information of $\mathcal{T}$ between $\mathcal{T}$ and $\mathcal{B}$. The replay attack is also enabled for $\mathcal{T}$ and $\mathcal{B}$, respectively. Moreover, $\mathcal{A}$ can trace location and behaviors of a tag holder without detection and impersonate a legitimate $\mathcal{T}$ or a legitimate $\mathcal{R}$. To remove these security and privacy problems, our protocol is based on mutual authentication between $\mathcal{T}$ and $\mathcal{B}$ guaranteeing freshness of identification information of $\mathcal{T}$.

However, we assume the communication channel between $\mathcal{R}$ and $\mathcal{B}$ is secure as most of previous work. $\mathcal{R}$ generates a random number and transmits it to make $\mathcal{T}$ anonymous when querying $\mathcal{T}$. This random number is integrated into the authentication message of $\mathcal{T}$ to protect the replay attack.

In our protocol, $\mathcal{T}$ does not have any real data and all authentication messages are random and hashed, so confidentiality is guaranteed even though the authentication messages are eavesdropped by $\mathcal{A}$ or $\mathcal{T}$ is tampered by $\mathcal{A}$.

Like [11], we adopt the similar database structure and a similar mechanism to prevent the data loss. In [11], a database D of $\mathcal{B}$ and manages a pair of records for each $\mathcal{T}$ in case the reply message from $\mathcal{B}$ to $\mathcal{T}$ is lost or intercepted. Instead of a pair of records, our scheme manages one additional field for each record K_{last} in D. K_{last} is a field to save the previous secret information which is replaced by the current secret information k.

4.2 Assumptions

4.2.1 General Assumptions

Our protocol works with the general assumption that $\mathcal{T}$ has a hash function $h()$, a pseudorandom number generator $PRNG$, XOR gate, and the capability to keep state during a single session. The widely acceptable low-cost RFID $\mathcal{T}$ would most likely require the usage of passive tags [21, 26]. To design our proposed protocol, we assume the low-cost $\mathcal{T}$ is passive and has a re-writable memory like EEPROM with reasonable size.

In our protocol, we assume $\mathcal{T}$ has a hash function. In [28], it is said that a hash function unit with block size of 64-bit can be implemented with only about 1.7 K-gate, so it is also assumed that there will be the practical implementation of hash function for the low-cost $\mathcal{T}$ with the desirable security. Like [11, 18], we assume that $\mathcal{T}$ only has its authentication related information. $\mathcal{T}$ also has a non-volatile memory for keeping values of k to process mutual authentication. The simple structures for the record of D and the tag memory are shown in Figure 4.2. Other required data of $\mathcal{T}$ for an application can be stored in D of $\mathcal{B}$, but we don't describe.

As the previous schemes [11, 26], we assumed $\mathcal{R}$ is a TTP and the com-

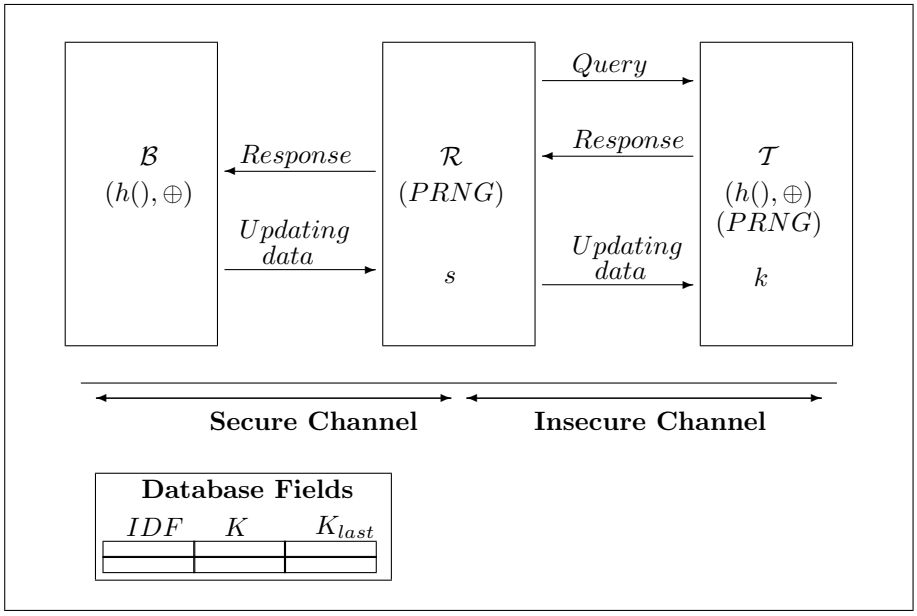

Figure 4.1: Overall architecture of our RFID system

munication channel between $\mathcal{R}$ and $\mathcal{B}$ is secure. Because both $\mathcal{R}$ and $\mathcal{B}$ have large computational power to implement traditional cryptographic primitives, *e.g.*, public key cryptosystem, the communication channel between $\mathcal{R}$ and $\mathcal{B}$ can be easily protected by well-known cryptographic protocols.

Only one cryptographic primitive which must be implemented in $\mathcal{R}$ is $SRNG$. s generated by $SRNG$ is used to verify the validity of a response of $\mathcal{T}$ by preventing the replay attack.

Figure 4.1 shows overall system architecture and message exchanges based on the assumptions.

4.2.2 Attacking Model

To solve the security risks and privacy issues, the following attacking model must be assumed and prevented.

Eavesdropping : $\mathcal{A}$ can easily eavesdrop a communications between $\mathcal{T}$ and

$\mathcal{R}$ without user's recognition because $\mathcal{T}$ and $\mathcal{R}$ emit their data through an air interface.

Replay Attack : $\mathcal{A}$ can eavesdrop the response message from $\mathcal{T}$, and retransmit the message to the legitimate $\mathcal{R}$.

Database Desynchronization : $\mathcal{A}$ tries to desynchronize identification information between $\mathcal{B}$ and $\mathcal{T}$. By desynchronizing identification information between $\mathcal{B}$ and $\mathcal{T}$, $\mathcal{A}$ can make $\mathcal{T}$ useless and trace $\mathcal{T}$ since $\mathcal{A}$ cannot update identification information.

Tampering : $\mathcal{A}$ should tamper with $\mathcal{T}$ because a low-cost $\mathcal{T}$ offers no and tamper-resistance mechanisms. It can induce information leakage problems of a tag holder. $\mathcal{A}$ tries to get knowledge of previous events in which $\mathcal{T}$ participate.

Forgery : When $\mathcal{A}$ tampers with $\mathcal{T}$, $\mathcal{A}$ has same information as $\mathcal{T}$ itself, and then we cannot prevent tag cloning. Therefore, we consider only the situation that $\mathcal{A}$ clones $\mathcal{T}$ without tampering with $\mathcal{T}$.

4.3 Security Requirements

To guarantee security and protect the privacy of a tag holder, we define the following requirements in cryptographic point of view [27, 18].

Data Confidentiality : The private information of $\mathcal{T}$ must be kept secure to guarantee user privacy. The information of $\mathcal{T}$ must be meaningless for its holder even though it is eavesdropped by an unauthorized $\mathcal{R}$.

Indistinguishability : Responses emitted by $\mathcal{T}$ must not be discriminated from the other $\mathcal{T}$. $\mathcal{A}$ can obtain responses of $\mathcal{T}$ by eavesdropping or actively querying. If $\mathcal{A}$ can discriminate $\mathcal{T}$ from other $\mathcal{T}$, $\mathcal{A}$ can trace a targeted $\mathcal{T}$ and then tag holder's privacy is violated.

Forward Security : $\mathcal{A}$ must not be able to trace the data back through previous events in which $\mathcal{T}$ was involved even if $\mathcal{A}$ acquires the secret data stored in $\mathcal{T}$. Because a low cost $\mathcal{T}$ doesn't have a tamper-resistance mechanism, $\mathcal{A}$ can get knowledge of data stored in $\mathcal{T}$ by tampering with $\mathcal{T}$. If $\mathcal{A}$ guesses a previous response of $\mathcal{T}$ from tampered data, $\mathcal{A}$ can discover an event participated by a tag holder.

Anti-cloning : Anti-cloning is an additional security requirement. This property means that $\mathcal{A}$ cannot clone $\mathcal{T}$ without tampering with $\mathcal{T}$. When $\mathcal{A}$ tampers with $\mathcal{T}$, $\mathcal{A}$ has same information as $\mathcal{T}$ itself, and then we cannot prevent tag cloning. However, there are many ways to cloning without tampering with $\mathcal{T}$, *e.g.*, the replay attack. Therefore, *anti-cloning* can be one of security requirements.

Availability : $\mathcal{T}$ must be available all the time when a user wants. Therefore, an authentication protocol should provides the data recovery against the data loss such as DoS, message hijacking, power interruption, *etc.* during the authentication processes. Especially, the desynchronization attack by utilizing a man-in-the-middle attack must be prevented. $\mathcal{A}$ may try to do a man-in-the-middle attack in order to desynchronize identification information between D and $\mathcal{T}$. $\mathcal{A}$ may try to do a desynchronizing attack by updating identification information of $\mathcal{T}$ while $\mathcal{B}$ is ignorant of this situation.

Besides, we must consider and evaluate the following security feature in the design of RFID authentication protocol.

Mutual Authentication : The mutual authentication between $\mathcal{T}$ and $\mathcal{B}$ must be provided as a measure of trust. By authenticating mutually, the replay attack and the cloning are prevented.

4.3.1 Protocol Setup

$\mathcal{B}$ and $\mathcal{T}$ can operate the XOR calculation and a common one-way hash function, $h : \{0,1\}^* \rightarrow \{0,1\}^l$. $\mathcal{R}$ has PRNG and a variable s whose length is l-bit. The role of s is to save pseudorandom number which is used in order to detect a replay attack. $\mathcal{T}$ also has a PRNG, which need not be the same as one of $\mathcal{R}$.

The variable r whose length is l-bit is saved in a volatile memory of $\mathcal{T}$. r is transmitted through RF and is tested for mutual authentication of components, *i.e.*, $\mathcal{B}$ and $\mathcal{T}$. Because r is saved in a volatile memory, the contents of r are automatically deleted at the end of the authentication process.

The variable k whose length is l-bit is saved in non-volatile memory of $\mathcal{T}$. k is used in order to identify ID of $\mathcal{T}$, so k must be different among all $\mathcal{T}$'s all the time. The initial value of k of each $\mathcal{T}$ is assigned by precalculation to guarantee each k of $\mathcal{T}$ to be always different. Let m be the number of $\mathcal{T}$'s in a system, and let n be the maximum number of authentication times of each $\mathcal{T}$. We construct a hash chain of secret information as follows: The hash chain starts from a secret seed t, the second one k_2 is $h(t)$, and the other x-th element k_x is $h(k_{x-1})$ where $3 \leq x \leq mn$. We must select t which makes a hash chain longer than mn. The initial value of k of each $\mathcal{T}$ is selected such that each one is far apart at least n in the hash chain. These initial values are saved in the each memory of $\mathcal{T}$ and maintained by $\mathcal{B}$ with the corresponding ID of $\mathcal{T}$. $\mathcal{B}$ and $\mathcal{T}$ update k to the next value of the hash chain synchronically when the authentication process is successfully done.

D of $\mathcal{B}$ has fields IDF, K, and K_{last}, which save the ID, the current k, the preceding k (the previous secret information which is replaced by the current k), respectively. Initially, IDF and K are set up with ID and initial k of each $\mathcal{T}$, respectively, and all values of the field K_{last} are null. The role of K_{last} is to prevent desynchronization. Even though $\mathcal{B}$ updated k of $\mathcal{T}$ (*i.e.*, the value of field K corresponding to $\mathcal{T}$) but $\mathcal{T}$ didn't receive that information under the

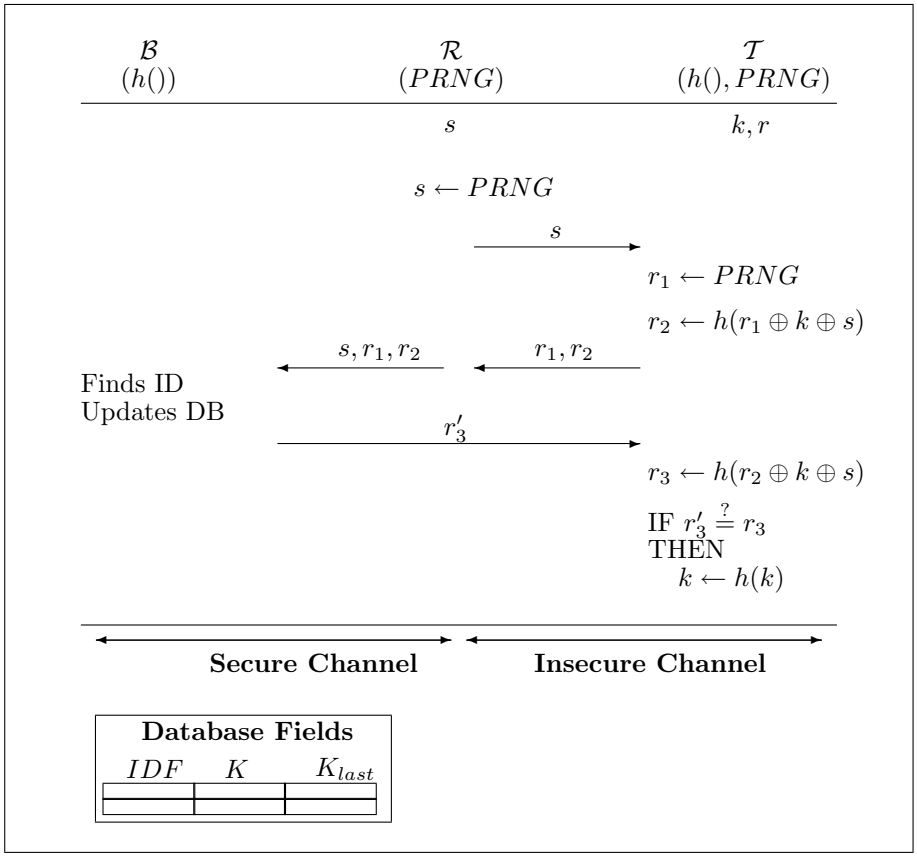

Figure 4.2: Our Protocol

influence of $\mathcal{A}$ or communication errors, $\mathcal{B}$ can know ID of $\mathcal{T}$ by referencing the field K_{last}.

4.3.2 Detailed Description

We describes the process of our authentication protocol as in Figure 4.2.

1. $\mathcal{R}$ generates and saves a new pseudorandom number s by utilizing $PRNG$, and sends s to $\mathcal{T}$.

2. $\mathcal{T}$ also generates a new pseudorandom number r_1 and sends r_1 to $\mathcal{R}$. After that, $\mathcal{T}$ generates r_2, *i.e.*, $h(r_1 \oplus k \oplus s)$, where s was sent by $\mathcal{R}$,

and sends r_2 to $\mathcal{R}$.

3. $\mathcal{R}$ delivers responses of $\mathcal{T}$ with the saved value s to $\mathcal{B}$, *i.e.*, s, r_1, and r_2.

4. In order to find ID of $\mathcal{T}$, $\mathcal{B}$ searches k' from the fields K and K_{last} which satisfies the following equation:

$$h(r_1 \oplus k' \oplus s) \stackrel{?}{=} r_2 \tag{4.1}$$

   where r_1, r_2, and s are values sent by $\mathcal{T}$. If only one k' satisfies Eq.(4.1), then we can know ID corresponding to k' in D is ID of $\mathcal{T}$ because Eq.(4.1) is true if k' and k is identical. If two more values satisfing Eq.(4.1) are found because of hash collisions (although the probability of this case is only about $m/2^{l-1}$), $\mathcal{B}$ informs the failure of searching ID of $\mathcal{T}$ to $\mathcal{R}$, and orders $\mathcal{R}$ to query again in order to restart the process from the first step.

5. $\mathcal{B}$ updates information of $\mathcal{T}$. If k' is found in the field K of a record, k' is copied to the field K_{last} of the record and the field K of the record is set to $h(k')$. If k' is found in the field K_{last}, we do not update D (because this situation means that $\mathcal{B}$ has already updated D at the previous authentication process but $\mathcal{T}$ didn't).

6. From s, r_2, and k', which are received values from $\mathcal{T}$ and the value found by testing Eq.(4.1), $\mathcal{B}$ calculates r'_3, *i.e.*, $h(r_2 \oplus k' \oplus s)$, and sends r'_3 to $\mathcal{R}$. $\mathcal{R}$ transfers r'_3 to $\mathcal{T}$ in order to inform the update.

7. In order to test the correctness of the value $\mathcal{R}$ sends, $\mathcal{T}$ tests the following equation:

$$r'_3 \stackrel{?}{=} r_3 \tag{4.2}$$

   If Eq.(4.2) is correct, $\mathcal{T}$ updates k to $h(k)$.

4.4 Analysis

4.4.1 Security Analysis

In this section, we analyze the security of our scheme. We show that our scheme is untraceable against all imaginable attacks except tampering with $\mathcal{T}$ in the random oracle model. Moreover, we describe that an attack based on tampering with $\mathcal{T}$ also limited by updating secure information. The followings are security analysis of our scheme against each attack.

Data Confidentiality

- *Attack based on Eavesdropping*: Our protocol guarantees the secure mutual authentication by utilizing only with the pseudorandom numbers s, r_1 and the hashed messages, $r_2 = h(r_1 \oplus k_j \oplus s)$, $r_3 = h(r_2 \oplus k_j \oplus s)$.

- *Attack based on Tampering*: $\mathcal{T}$ stores no privacy information of a tag holder. All other required data of $\mathcal{T}$ for an application are stored in D of $\mathcal{B}$. Although the information transmitted from $\mathcal{T}$ of the authentication is eavesdropped by $\mathcal{A}$, it is meaningless. Thus, data confidentiality of tag owners is guaranteed and the user privacy on data is strongly protected.

Indistinguishability

- *Attack based on Eavesdropping*: $\mathcal{A}$ can collect s, r_1, r_2, and r_3 during one authentication process by eavesdropping. Because s and r_1 are random values, they are useless to trace $\mathcal{T}$. Moreover, if $h()$ is a random oracle, $\mathcal{A}$ who doesn't get knowledge of k in $\mathcal{T}$ cannot distinguish r_2 and r_3 from a random value. Since the inputs of $h()$ to generate r_2 and r_3 are random because of s, the probability that the inputs are identical among each session is negligible. Because the random oracle is assumed to be a function with the property that if a value in its domain is not

queried before then the corresponding function value is a random value, r_2 and r_3 look like random value to $\mathcal{A}$. Therefore, $\mathcal{A}$ cannot trace $\mathcal{T}$ by eavesdropping.

- *Attack based on Controlling s*: $\mathcal{A}$ may try to make responses of $\mathcal{T}$ have some traceable patterns by controlling s as the role of $\mathcal{R}$. If $\mathcal{A}$ can guess r_1 before $\mathcal{A}$ sends s, $\mathcal{A}$ can make responses of $\mathcal{T}$ be always identical by sending the value r_1 as s repeatedly. In that case, r_2 is always $h(r_1 \oplus k \oplus r_1)$, *i.e.*, $h(k)$. Therefore, $\mathcal{A}$ can trace $\mathcal{T}$. However, because r_1 is a pseudorandom number, $\mathcal{A}$ cannot predict r_1. Therefore, $\mathcal{A}$ cannot trace $\mathcal{T}$ by means of controlling s.

- *Attack based on Power Interruption*: Let's assume a capacitor $\mathcal{T}$ doesn't have sufficient capacity in order to operate all stages all at once, and then let's assume $\mathcal{A}$ can halt a process of $\mathcal{T}$ any time when $\mathcal{A}$ wants by interrupting power. On this assumption, some schemes may be weak against tracking. For example, the Ohkubo's scheme [18] emits data and then updates contents of memory. $\mathcal{A}$ can obtain an identical response every time by stopping the authentication process of $\mathcal{T}$ just before $\mathcal{T}$ updates the contents of memory. However, our scheme is strong against power interruption because $\mathcal{T}$ always generates a new random number at the first stage and the random number guarantees that each session is different.

Forward Security

- *Attack based on Tampering*: We can image an attack that after $\mathcal{A}$ obtains k by tampering with $\mathcal{T}$ and has eavesdropped plenty of interactions between $\mathcal{B}$ and various $\mathcal{T}$'s, $\mathcal{A}$ tries to distinguish the response of $\mathcal{T}$ from others. $\mathcal{A}$ tests Eq.(4.1) by using the tampered k and collected interactions. If an interaction passes the test, $\mathcal{A}$ can guess the tampered $\mathcal{T}$ took part in that interaction. However, this forward trace is impossible

over successive authentication. Even though $\mathcal{A}$ knows the current k, $\mathcal{A}$ cannot know previous k of $\mathcal{T}$ because k is updated by a one-way function $h()$. Therefore, the forward trace is limited to a short period and is impossible whenever the successive authentication process is done.

Anti-cloning

- *Replay Attack*: $\mathcal{A}$ may try to do a replay attack by eavesdropping legitimate interactions, but $\mathcal{A}$ cannot success cloning by a replay attack because s is different for each session.

- *Attack by Generating a legitimate response*: Because $\mathcal{A}$ doesn't know k of $\mathcal{T}$, $\mathcal{A}$ cannot generate a legitimate r_2 from r_1 chosen by $\mathcal{A}$ and s given by $\mathcal{R}$.

Availability

- *Attack based on Updating k in D*: $\mathcal{A}$ may try to do a man-in-the-middle attack in order to desynchronize k between D and $\mathcal{T}$. We assume $\mathcal{T}$ is apart from $\mathcal{R}$ and $\mathcal{A}$ can relay the data between these two components. When $\mathcal{R}$ starts the authentication process, $\mathcal{A}$ transfers s from $\mathcal{R}$ to $\mathcal{T}$ and then transfers r_1 and r_2 to $\mathcal{R}$. Then, $\mathcal{B}$ updates k of $\mathcal{T}$ in D. After that, $\mathcal{B}$ sends r'_3 though $\mathcal{R}$ in order to update k in $\mathcal{T}$. At this time, if $\mathcal{A}$ doesn't deliver r'_3 to $\mathcal{T}$, k becomes different between D and $\mathcal{T}$. If $\mathcal{B}$ only uses the latest k to find ID, this situation makes $\mathcal{T}$ useless. However, our scheme saves another k which was replaced by the latest k in the field K_{last}, and $\mathcal{B}$ can identify $\mathcal{T}$ in this situation. Because $\mathcal{B}$ doesn't update k when $\mathcal{T}$ is identified using field K_{last}, two fields K and K_{last} are sufficient to defend the system from a desynchronization attack.

- *Attack based on Updating k in $\mathcal{T}$*: $\mathcal{A}$ may try to do a desynchronizing attack by updating secret information of $\mathcal{T}$ while $\mathcal{B}$ is ignorant of this

situation. However, without a help of $\mathcal{B}$, $\mathcal{A}$ cannot know r_3 because $\mathcal{A}$ doesn't know current k of $\mathcal{T}$ which is required in order to calculate r_3 from r_2. Therefore, $\mathcal{A}$ cannot success this attack.

Mutual Authentication

- *Attack based on Guessing r_2 or r_3*: $\mathcal{A}$ cannot generates r_2 from r_1 chosen by $\mathcal{A}$ and s given by $\mathcal{R}$ because $\mathcal{A}$ doesn't know secret information k. Therefore, $\mathcal{A}$ must randomly choose r_2 to be authenticated as $\mathcal{T}$. However, the probability that this attack succeeds is only $1/2^l$ where l is length of r_2, so this attack is negligible. For that same reason, $\mathcal{A}$ cannot be authenticated as a legitimate $\mathcal{B}$ or a legitimate $\mathcal{R}$ to $\mathcal{T}$.

- *Attack based on the Hash Chain*: The maximum number of authentication of each $\mathcal{T}$, n, should be big enough to stand against a brute-force attack based on tampering with $\mathcal{T}$. At the beginning, $\mathcal{A}$ reads k of $\mathcal{T}$, k_{tamper}, by tampering with $\mathcal{T}$. After that, $\mathcal{A}$ can know k of the other targeted $\mathcal{T}$, k_{target}, as follows. Let's assume $\mathcal{A}$ luckily collects the k_{tamper} which is a previous value than k_{target} in the hash chain. $\mathcal{A}$ queries with any s and receives r_1 and r_2 from k_{target}. After that, $\mathcal{A}$ can find the k_{target} by testing Eq.(4.1) with k' which are all values of the hash chain starting from k_{tamper}. Because the average length of the hash chain is about $2^{l/2}$, (not 2^l, because of the birthday paradox [25, 1]), this attack is much faster way to find the secret key than a brute-force searching from 0 to $2^l - 1$. This weakness is not only our problem but also the problem of schemes which utilize a hash chain to identify $\mathcal{T}$.

Chapter 5
Comparison

5.1 Security Comparison

Table 5.1: Efficiency and Security

Protocol	[11]	[9]	[16]	[17]	Our scheme
Computation at $\mathcal{T}$	$O(1)$	$O(1)$	$O(1)$	$O(m)$	$O(m)$
Data Confidentiality	O	O	O	O	O
Anti-cloning	O	O	O	O	O
Availability	X	O	O	O	O
Indistinguishability	X	△	△	O	O
Forward Security	X	-	X	X	△

m : the number of $\mathcal{T}$'s in a system
O : satisfy
△ : partially satisfy
X : do not satisfy
- : do not know

We analyze security of previous schemes in Chapter 3 and security of our scheme in Section 4.4. In Table 5.1, we summarize security and efficiency comparisons of our protocol with other mutual authentication schemes. The notation X means that a scheme doesn't satisfy a given security requirement.

The notation $\triangle$ means that $\mathcal{A}$ cannot trace $\mathcal{T}$ over successive authentication but can trace $\mathcal{T}$ within successive authentication. The notation O means that a scheme satisfies a given security requirement. The notation - means that we don't know whether a scheme satisfies a given security requirement or not because the scheme doesn't describe a detail process.

As in Table 5.1, all schemes satisfy anti-cloning. Because all values emitted by $\mathcal{T}$ are randomized and there is no data which reveals information of a object $\mathcal{T}$ is attached in memory of $\mathcal{T}$, all schemes including our scheme satisfy data confidentiality. Because $\mathcal{A}$ cannot generate a legitimate response without knowledge of secure information of $\mathcal{T}$, $\mathcal{A}$ cannot cloning $\mathcal{T}$ without tampering with $\mathcal{T}$.

Required computation of our scheme to find ID of a given $\mathcal{T}$ at $\mathcal{B}$ is $O(m)$, which means that required computation is increased as the number of $\mathcal{T}$'s in the system is increased. However, Table 5.1 shows our scheme offers the most enhanced security in RFID mutual authentication schemes with respect to user privacy. Our scheme is perfectly indistinguishable and almost forward secure. As compared with [17] whose computation at $\mathcal{B}$ is also $O(m)$, our scheme enhances user privacy with respect to forward security.

5.2 Efficiency Comparison

In Table 5.2, we compares our scheme with other schemes about efficiency at $\mathcal{T}$. The notation l means the number of bits of a data unit. At the line which shows required hash operations at $\mathcal{T}$, $+\alpha$ means that Dimitriou's scheme [9] may require additional hash operations to update ID because the scheme doesn't describe a method to update ID.

The hash operations of our scheme is one more than [16] and [17] which require the least number of hash operations. The communication complexity is the mount of data transmission from or to $\mathcal{T}$. The required communication complexity of our scheme is the middle of all schemes. Values which must be

Table 5.2: Efficiency in $\mathcal{T}$

Protocol	[11]	[9]	[16]	[17]	Our scheme
Hash operations	3	3+α	2	2	3
Communication complexity	$5l$	$5l$	$3l$	$4l$	$4l$
Non-volatile memory	$3l$	l	l	$2l$	l

l : the number of bits of a data unit

α : additional hash operations to update ID

preserved after authentication should be saved in non-volatile memory. Our scheme requires the smallest non-volatile memory among all schemes.

Chapter 6
Conclusion

In this thesis, we have studied the design and analysis of authentication schemes for low-cost RFID environment. We have reviewed previous works related to hash-based protocols as well as other security schemes which are used in construction of our protocol.

In order to protect user privacy, we proposed an RFID mutual authentication scheme which utilizes a hash function and synchronized secret information like many published schemes.

Our protocol is robust enough since it protects the eavesdropping, the replay attack, and the database desynchronization. Moreover, our protocol is almost secure against tampering with a tag because our protocol provides forward security between successive mutual authentications.

To the best of our knowledge, our scheme offers the most enhanced security feature in RFID mutual authentication scheme with respect to user privacy allowing one more hash operation in comparison with [17] which requires identical computational complexity at a back-end server. Moreover, our scheme reduces required non-volatile memory by half compared to [17]. Since non-volatile memory is an expensive unit in a tag, it will cut down the tag's cost. Therefore, our scheme can be one of good options for diverse systems.

As future work, we will prove security of our protocol by means of provable security. In the aspect of implementation, we need more specified complexity analysis to get the clearer security level.

동기화된 비밀정보를 이용한 RFID 시스템에서의 상호 인증 기법

이상신

RFID(Radio Frequency IDentification)는 자동 인식 시스템으로써, “RFID 태그”로 불리는 장치를 이용해 우리가 관리하고자 하는 물체에 대한 정보를 저장하고 원격으로 검색한다. RFID 시스템은 물리적인 접촉이나 광학적인 접촉이 없이 라디오 주파수를 이용하여 다량의 태그를 인식할 수 있어서 바코드 기술보다 여러모로 유용하다. RFID 시스템은 공급망 관리, 재고 관리, 창고 관리 등과 같은 산업의 다양한 분야에 사용될 수 있고 유비쿼터스 환경에서 개인에게 편의를 제공할 수 있다.

그러나 RFID 시스템은 접근제어와 메모리 읽기 방지 기법을 제공하지 않는 태그를 사용하기 때문에 안전성 문제를 내재하고 있다. RFID 태그가 공격자를 포함한 모든 이에게 정보를 방출함으로써, RFID 시스템은 회사의 정보를 유출할 수 있고 개인의 프라이버시 침해를 야기할 수 있다. 예를 들어, 부정직한 회사는 경쟁회사의 물류 정보를 수집하는 시도를 할 수 있다. 태그로부터의 응답을 활용함으로써, 공격자는 개인 사용자가 소지하고 있는 물품의 정보를 얻는 것을 시도하거나 사용자를 추적하는 것을 시도할 수 있다. 게다가 공격자가 위조된 태그를 이용해서 부당한 이득을 취하는 시도를 할 수 있다. 이런 취약성이 대중이 RFID 기술을 사용하는 것을 꺼리게 한다[2, 23].

비록 다른 시스템의 비슷한 취약성을 해결하기 위한 많은 암호학적 프리미티브가 존재하더라도, 저가형 태그의 적은 계산량으로 인해서 그것들을 RFID 시스템에 적용하지 못한다. 결과적으로 태그 내에서 적은 계산량으로 구현 가능한 새로운 프로토콜을 필요로 한다.

따라서 우리는 저가형 태그에서 구현 가능한 프로토콜을 제안한다. 추

적으로부터 사용자를 보호하기 위하여, 우리는 다른 여러 기법들[11, 9, 17, 16]과 같이 해쉬 함수와 동기화된 비밀 정보를 이용하는 RFID 상호인증 기법을 제안한다. 백엔드 서버에서 같은 정도의 계산량을 요구하는 [17]와 비교하였을 때, 우리의 기법은 한번의 추가적인 해쉬 연산을 함으로써 사용자 프라이버시 관점에서 가장 강력한 안전성을 제공한다.

References

1. Morton Abramson and W.O.J. Moser, "More Birthday Surprises", American Mathematical Monthly 77, pp. 856-858, 1970

2. C.A.S.P.I.A.N. http://www.nocards.org/.

3. Auto-ID Center, "860MHz-960MHz Class I radio frequency identification tag radio frequency & logical communication interface specification proposed recommendation Version 1.0.0", *Technical Report MIT-AUTOID-TR-007*, November 2002.

4. Auto-ID Center. http://www.autoidcenter.org/.

5. Gildas Avoine "Radio frequency identification: adversary model and attacks on existing protocols", *Technical Report LASEC-REPORT-2005-001*, EPFL, Lausanne, Switzerland, Septemper 2005.

6. Gildas Avoine and Philippe Oechslin, "RFID traceability: a mulilayer problem", *Financial Cryptography − FC*, pp. 125-140, March 2005.

7. Koji Chida, Kunio Kobayashi, and Hikaru Morita, "Efficient sealed-bid auction using hash chain, *International Conference on Information Security − ICISC*, pp. 183-191, 2000.

8. Ivan Bjerre Damgård, "Collision free hash functions and public key signature schemes", *Advances in Cryptology-Eurocrypt*, pp. 203-216, April 1987.

9. Tassos Dimitriou, "A lightweight RFID protocol to protect against traceability and cloning attacks", *Conference on Security and Privacy for Emerging Areas in Communication Networks − SecureComm*, September 2005.

10. Philippe Golle, Markus Jakobsson, Ari Juels, and Paul Syverson, "Universal re-encryption for mixnets", *The Cryptographers' Track at the RSA Conference − CT-RSA*, pp. 163-178, February 2004.

11. Dirk Henrici and Paul Müller, "Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers", *IEEE International Workshop on Pervasive Computing and Communication Security − PerSec*, pp. 149-153, March 2004.

12. Ari Juels, "Minimalist cryptography for low-cost RFID tags", *International Conference on Security in Communication Networks − SCN*, pp. 149-164, September 2004.

13. Ari Juels, Ronald Rivest, and Michael Szydlo, "The blocker tag: selective blocking of RFID tags for consumer privacy", *ACM Conference on Computer and Communications Security − ACM CCS*, pp. 103-111, October 2003.

14. Ari Juels and Ravikanth Pappu, "Squealing euros: privacy protection in RFID-enabled banknotes", *Financial Cryptography − FC*, pp. 103-121, January 2003.

15. Leslie Lamport, "Password authentication with insecure communications", *Communications of ACM*, pp. 770-772, November 1981.

16. Su-Mi Lee, Young Ju Hwang, Dong Hoon Lee, and Jong In Lim, "Efficient authentication for low-cost RFID systems", *International Conference on Computational Science and its Applications - ICCSA*, pp. 619-627, May 2005.

17. David Molnar and David Wagner, "Privacy and security in library RFID: issues, practices, and architectures", *ACM Conference on Computer and Communications Security − ACM CCS*, pp. 210-219, October 2004.

18. Miyako Ohkubo, Koutarou Suzuki, and Shingo Kinoshita, "Cryptographic approach to 'privacy-friendly' tags", *RFID Privacy Workshop*, November 2003.

19. Ronald Rivest and Adi Shamir, "Payword and micromint: two simple micropayment schemes", *International Workshop on Security Protocols*,, pp. 69-87, 1996.

20. Junichiro Saito, Jae-Cheol Ryou, and Kouichi Sakurai, "Enhancing privacy of universal re-encryption scheme for RFID tags", *Embedded and Ubiquitous Computing − EUC*, pp. 879-890, August 2004.

21. Sanjay Sarma, Stephen Weis, and Daniel Engels, "RFID systems, security and privacy implications", Technical Report MIT-AUTOID-WH-014, AutoID Center, MIT, 2002.

22. Sanjay Sarma, Stephen Weis, and Daniel Engels, "Radio-frequency identification: security risks and challenges", *Cryptobytes, RSA Laboratories*, pp. 2-9, Spring 2003.

23. Spychips. http://www.spychips.com/.

24. UHF wireless tag, Auto-ID Center, http://www.autoidcenter.org/research/mit-autoid-tr007.pdf.

25. David Wagner, "A generalized birthday problem", *Advances in Cryptology − Crypto*, pp. 288-303, 2002.

26. Stephen Weis, Sanjay Sarma, Ronald Rivest, and Daniel Engels, "Security and privacy aspects of low-cost radio frequency identification systems", *International Conference on Security in Pervasive Computing − SPC*, pp. 454-469, March 2003.

27. Stephen Weis, "Security and privacy in radio-frequency identification devices", *Masters Thesis, MIT*, May 2003.

28. Kaan Yüksel, "Universal hashing for ultra-low-power cryptographic hardware applications, *Master's Thesis, Dept. of Electronical Engineering, WPI*, 2004.

Acknowledgements

First, I would like to express my sincere gratitude to Prof. Kwangjo Kim, my academic advisor, for his constant direction and support. He always has shown his consistent affection and encouragement for me to carry out my research and life in ICU. Special thanks also goes to Prof. Jae Choon Cha and Ph.D HoWon Kim for their generosity and agreeing to serve as committee members of my thesis. I appreciate to Tomoyuki Asano for guidance when I was an internship student of Sony company.

I also would like to thanks to all members of cryptology and information security laboratory: Kyusuk Han, Zeen Kim, Hyunrok Lee, Jaemin Park, SungChul Heo, and Youngjoon Seo, Vo Duc Liem and Dang Nguyen Duc from Vietnam, Divyan Munirathnam Konidala from India, for giving me lots of interests and good advices during the course of my study.

In addition, I appreciate to the graduates, Jeongkyu Yang and Seok-kyu Kang, and Xiaofeng Chen and Ping Wang from China for their everlasting guidance in life and study of ICU.

Most of all, I should mention my parents for their endless concerns and devotional affection. I cannot forget their trust and encouragement on me. My sister also has given me warmhearted concerns.

Finally, I will always remember the life of ICU. It filled up my poor knowledge and made me a grown-up person.

Curriculum Vitae

Name : Sangshin Lee

Date of Birth : Jul. 1. 1980

Sex : Male

Nationality : Korean

Education

2004.3–2006.2 Cryptology and Information Security, Engineering
Information and Communications University (M.S.)

1999.3–2004.2 Information and Computer Engineering
Ajou University (B.E.)

Career

2005.07–2005.12 Graduate Research Assistant
Samsung-ICU Research Center (Embedded Security 3-4)
Samsung Electronics

2005.07–2005.12 Graduate Research Assistant
A Study on the Security of RFID Gen2
Electronics and Telecommunications Research Institute(ETRI)

2005.06–2005.08 Apprentice Researcher
Information Technology Laboratories. Secure System Group, SONY, Japan

2005.05– Graduate Research Assistant
Research on a light-weight Protocol in RFID Application Environment
Electronics and Telecommunications Research Institute(ETRI)

2005.01–2005.12 Graduate Research Assistant
Research on Link Layer Security
Electronics and Telecommunications Research Institute(ETRI)

2004.12–2005.08 Graduate Research Assistant
Development & Implementation of Link Protection System Technology between Set-top Box and Handheld Device
Samsung Electronics

2004.06–2004.12 Graduate Research Assistant
Research on RFID Privacy and Security Protection
Electronics and Telecommunications Research Institute(ETRI)

2004.04–2004.11 Graduate Research Assistant
A Study on the Security for Special Digital Signature
Security Research Center(SERC), Hannam University

2004.03–2005.12 Teaching Assistant
Institute for IT-gifted Youth

2004.03–2005.02 Graduate Research Assistant
Research on A Group-Aware Middleware Infrastructure for Active Surroundings
Institute of Information Technology Assessment(IITA)

Publications

(1) 2006.01 Sangshin Lee, Tomoyuki Asano, and Kwangjo Kim, RFID Mutual Authentication Scheme based on Synchronized Secret Information, To be appeared *In the Proceedings of SCIS'06*, Hiroshima, Japan.

(2) 2005.12 이상신, Tomoyuki Asano, 김광조, 상호인증을 통한 RFID 프라이버시 보호 기법, 2005년 한국정보보호학회 동계정보보호학술대회, Vol.15, No.2, pp. 241-245, 서울대학교, 서울

(3) 2005.06 이상신, 김진, 김광조, 저가형 RFID를 위한 효율적인 프라이버시 보호 기법, Vol.15, No.1, pp. 569-573, 2005년 한국정보보호학회 하계정보보호학술대회, 조선대학교, 광주