A Thesis for the Degree of Master of Science

Security and Privacy on Authentication Protocol for Low-cost Radio Frequency Identification

Jeongkyu Yang

School of Engineering

Information and Communications University

2005

Security and Privacy on Authentication Protocol for Low-cost Radio Frequency Identification

Security and Privacy on Authentication Protocol for Low-cost Radio Frequency Identification

Advisor : Professor Kwangjo Kim

by

Jeongkyu Yang
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. 23. 2004
Approved by

(signed)

Professor Kwangjo Kim
Major Advisor

Security and Privacy on Authentication Protocol for Low-cost Radio Frequency Identification

Jeongkyu Yang

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. 23. 2004

Approved:

Chairman of the Committee
Kwangjo Kim, Professor
School of Engineering

Committee Member
Jae Choon Cha, Assistant Professor
School of Engineering

Committee Member
Jong Suk Chae, Ph.D
Electronics & Telecommunications Research Institute (ETRI)

M.S. Jeongkyu Yang
20032047

Security and Privacy on Authentication Protocol for Low-cost Radio Frequency Identification

School of Engineering, 2005, 60p.
Major Advisor : Prof. Kwangjo Kim.
Text in English

Abstract

In the near future, radio frequency identification (RFID) technology is expected to play an important role for object identification as a ubiquitous infrastructure. One of its main objectives is the next generation technology that is mainly used to identify massive objects and will be a substitution for the existing optical bar code system.

On the other hand, low-cost RFID tags are highly resource-constrained, cannot support its long-term security and it is very restricted to implement the existing cryptographic algorithms. Thus, they have potential risks and may violate privacy for their bearers, and user privacy issues would be a big barrier for the admirable usage of RFID in the future ubiquitous society. As far as user privacy issues concerned, we should consider the data leakage illegally from a tag and the malicious tracking for the unique ID of a tag.

Due to the characteristics of RFID interface, the identification data from RFID tags must guarantee their integrity with other entities, readers or back-end servers. In addition to it, these entities must trust each other, be seamlessly integrated with their authentication messages, and anonymously interact protecting user location privacy without revealing any information of tag bearers.

To protect user privacy and remove security vulnerabilities, we propose a robust and privacy preserving mutual authentication protocol that fits the low-cost RFID environment. Different from the previous works, our protocol firstly provides reader authentication and prevent active attacks based on the assumption that a reader is no more a trusted third party and the communication channel between the reader and the back-end server is insecure like wireless channel. In addition, the proposed protocol exhibits forgery resistance against simple copy, or counterfeiting prevailing RFID tags. As tags only have hash function and exclusive-or operation, the proposed protocol is very feasible for low-cost RFID system compared to the previous works. We also firstly adapt GNY logic-based formal proof for the correctness of the proposed authentication protocol.

Contents

List of Tables

List of Figures

List of Abbreviations

BAN-logic An authentication logic introduced by M. Burrow, M. Abadi, and R. Needham

CRHF Collision Resistant Hash Function

DoS Denial-of-Service

EEPROM Electrical Erasable Programmable Read Only Memory

EPC Electronic Product Code

GNY-logic A authentication logic by L. Gong, R. Needham, and R.yahalom

IC Integrated Circuit

NH A universal hash function of UMAC

OWHF One-Way Hash Function

PH NH-Polynomial, a variation of NH

PR NH-Polynomial with Reduction, a variation of NH

PRNG Pseudo Random Number Generator

RFID Radio Frequency Identification

RNG Random Number Generator

TTP Trusted Third Party

UMAC Unconditionally Secure MAC

WH Weighted NH-Polynomial with Reduction, a variation of NH

XOR Exclusive-or

List of Notations

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

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

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

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

C Chip serial number that is embedded into $\mathcal{T}$ during manufacturing.

$E_k()$ Symmetric-key encryption function with the secret key, k.

$D_k()$ Symmetric-key decryption function with the secret key, k.

$h()$ One-way hash function.

$h_k()$ Keyed hash function with the secret key k.

ID Temporary identification value of $\mathcal{T}$, it is used to make the shared secret k_2 randomized.

ID' Temporary value to be used to make the shared secret k_1 randomized.

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

k_1 Shared random secret between $\mathcal{T}$ and $\mathcal{B}$.

k_2 Shared random secret between $\mathcal{T}$ and $\mathcal{B}$.

RNG Random Number Generator in $\mathcal{R}$.

r Random number generated by RNG of $\mathcal{R}$.

S Keyed one-way hash value of $h_k(r)$.

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

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

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

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

T_1 A field for the shared random secret, k_1.

T_2 A field for the shared random secret, k_2.

AE A field for the pointer linking a pair of records each other to counteract for the data loss.

CN A field for the chip serial number, C, of $\mathcal{T}$.

$DATA$ A field for all other application related data of $\mathcal{T}$.

Chapter 1
Introduction

1.1 Radio Frequency Identification

Radio Frequency Identification (RFID) technology is expected to take an important role for object identification as a ubiquitous infrastructure, and is currently considered as the next generation technology that is mainly used to identify massive objects and will be a substitution for the existing optical bar code system in the near future. The micro-chip equipped on a tag has a unique identification information and is applicable for various fields such as animal tracking, supply chain management, inventory control, *etc.*

RFID has been already used in many applications. In 2003, Mark & Spencer, the largest retailer of clothing in the U. K. developed Radio Frequency Identification (RFID) tagging for a trial on individual garments [22]. This follows the success of trials on RFID tagging on 3.5 million produce delivery trays in Marks & Spencer's food supply chain. Michelin also embedded RFID tags in tires for its tire tracking system [23]. For more sensitive area, European Central Bank (ECB) determined to embed RFID tags on its banknotes from 2005 for special purposes like banknote tracking and strong forgery resistance as well as user privacy protection [6]. After Exxon Mobile developed the mobile payment system known as Speedpass, many service providers like SONY and Philips has been tried to develop mobile payment system using radio frequency radiation [24].

1.2 Design Considerations on Authentication Protocol

RFID applications is getting growing in areas such as admission control, payment, ticketing, *etc.* that sophisticated security measures are needed. Without security, illegal activities cheating RFID systems, for example 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 are also issued since anyone can intercept communication between tags and readers and get information about RFID tag bearers. We describe these security issues in the next section 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, tags, readers, and back-end servers authenticate each other and agree on the secret session key which will be used to secure the later session. As denoted in [1, 3, 4, 6, 14, 18], 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 the mutual authentication protocol will be described in section 3.

When designing a 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.

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. Meanwhile, security problems are not considered as a big barrier in the real world at this moment. 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. In this context, the expected privacy and security problem should be protected for the admirable usage of RFID, and several papers proposed security and privacy protections schemes for RFID

In this thesis, various 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 tag bearers, which requires confidentiality, anonymity, and integrity in the cryptographic point of view.

Our protocol is robust enough against the active attacks such as the man-in-the-middle attack, and the replay attack as well as the data loss [14, 15, 17]. The protocol is based on mutual authentication between a tag and a back-end server, and provides authentication for the reader in case the reader is

not regarded as the trusted third party (TTP). Our protocol is also forgery resistant against the attacker who copies or counterfeits a prevailing RFID tag. The formal proof of correctness is firstly provided for the proposed authentication protocol based on GNY logic.

1.4 Outline of the thesis

The remainder of the thesis is organized as follows.

In section 2, we introduce RFID system primer, cryptographic primitives, and related works. In terms of RFID, we discuss about the current security problems concerning the characteristics on RFID technology as well as the potential privacy issues. Together with this, several schemes and protocols are introduced in this section. Those schemes are mostly focused on how to guarantee security and protect user privacy in low-cost RFID environment.

Then, we propose authentication scheme in section 3. To satisfy low computational power on the existing RFID, the proposed authentication protocol adapts simple cryptographic primitives, one-way hash function, and random number generators. Mutual authentication is the basis for the proposed protocol, provides user privacy protection and protects attacks.

In section 4, we show security proof and analyze its security and performance. To show the correctness of our proposed protocol, GNY logic is introduced. GNY logic is an extension of BAN logic that is one of the best known modal logic to prove and analyze the security and correctness on authentication protocol. Then, the security analysis is given and we compare the computational loads and the required memory on the proposed protocol with previous works. Finally, we conclude in section 5.

Chapter 2
Preliminaries

2.1 RFID Background

2.1.1 Overview

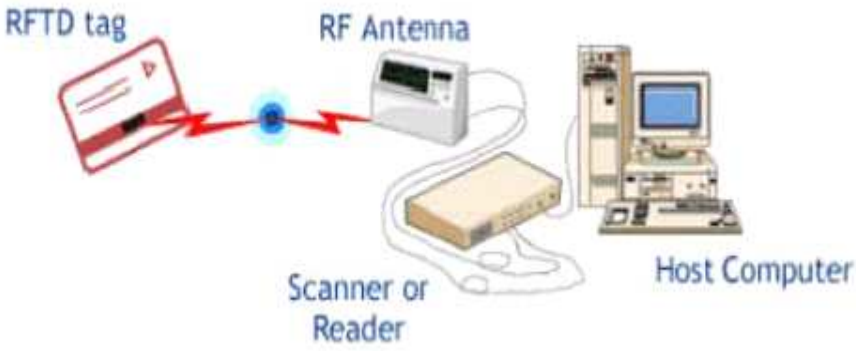

Figure 2.1: Typical RFID System

As shown in Figure 2.1, typical RFID system consists of RF tags (or transponders), and RF tag readers (or transceivers) [15]. In addition, a back-end server is usually working together in RFID system as a separate component [14, 18].

A tag consists of IC chip and antenna, and transmits its stored data to a reader as response for radio frequency interrogation of readers. A reader sends a radio signal to tags, receives the data transmitted from a tag, and sends the data to a back-end server. The back-end server is a secure server and has

a database which stores the various information of each tag like the identification information, all other application related data of tags and location information of readers. The back-end server determines each tag's identification from the information responded from the tag by way of an authorized reader. The back-end server replies the data from its database to the reader. The transmitted data is totally determined by a specific application.

For the privacy of tag bearers, the unique ID of each tag must be anonymous to protect the location privacy, and all messages to process the authentication must be secure to guarantee the user data privacy. A reader is generally considered as a TTP. The insecure communication channel through the air interface between tags and reader is more vulnerable to an attack than the secure channel between readers and back-end servers. The range of radio frequency from the reader is much stronger than that from the tag when the tag is passive and receives power from the reader. Thus, an adversary can eavesdrop for interrogation of the reader from the much longer distance [17].

The cost of a RFID tag should be reduced under US$0.50 for most applications. In order to achieve this price, IC should be priced less than US$0.20 [17]. This price barrier for low-cost tags restricts the range of gates in a tag number from 7.5 to 15 K, and the number of gates for security purpose is limited to from 2.5 to 5 K [16]. Due to this, it must be infeasible to use the existing cryptographic algorithm [5].

2.1.2 RFID System Components

Generally the typical RFID system components are tag, readers and back-end servers. Those three entities are seamlessly integrated each other. In addition to them, we add one more entity, operating frequency as a basic component.

Tags (or Transponders)

RFID tags or transponders are either *passive* or *active* devices. Passive RFID tags do not have a dedicated power supply. They derive their operation power from the electrical field generated by the reader, and do not operate unless they are in very close proximity to the reader [33]. As they do not have internal batteries, passive transponders are quite small and can be installed unobtrusively.

Active tags or transponders have an internal battery and therefore have significantly greater read range, as much as 50 feet in some cases. However, they are significantly larger than passive tags and lend themselves to a different set of applications.

Tags are categorized into several types according to their physical characteristics and their purposes for applied applications. ISO/IEC categorizes RFID tag into *type A* and *type B* according to air interface since the characteristics of tags is mostly very different according to used radio frequency. On the other hand, EPC Global divides it into six categories, *Class 0-1, Class 2, Class 3, Class 4, Class 5* as a defacto standard; Class 0 and Class 1 are types of read only passive identity tags; Class 2 is type of passive tags with additional functionality like memory or encryption; Class 3 is type of semi-passive tags and may support broadband; Class 4 is type of active tags and may be capable of broadband peer-to-peer communication with other active tags in the same frequency band and with readers; Class 5 is type of active tags, and can support power Class 0-3 tags and communicate with Class 4 tags and with each other wirelessly.

It is also convenient to classify tags by their functionality [17]. Table 2.1 shows five classes based on functionality defined by MIT Auto-ID Center.

Table 2.1: Tag Functionality Classes

Class	**Nickname**	**Memory**	**Power Source**	**Features**
0	Anti-Shoplift Tags	None	Passive	Article Surveillance
1	EPC	Read-Only	Any	Identification Only
2	EPC	Read-Write	Any	Data Logging
3	Sensor Tags	Read-Write	Semi-Passive or Active	Environmental Sensors
4	Smart Dust	Read-Write	Active	Ad Hoc Networking

Readers (or Transceivers)

Readers or Transceivers have a number of varied responsibilities including powering transponders, identifying them, reading data from them, writing to them, and communication with back-end servers [33]. Computations, such as cryptographic calculations, may be carried out by the reader on behalf of a tag [17].

Generally, the communication channel between tags and readers is insecure since the channel is based on air interface. On the other hand, the communication channel between readers and back-end servers is consider as secure channel. Readers might be handheld devices and will be mobile devices based on wireless network.

Back-end Servers (or Back-end Databases)

Back-end servers receive data from readers, enter the data into a database of their own, and provide access to the data in a number of forms that are useful to the sponsoring organization [33].

Sometimes, many applications and authentication protocol describe that a reader takes a role of data processing. However, a back-end server is typically

Table 2.2: Characteristics of Frequencies for RFID System

Devices	**Bandwidth**	**Typical Freq.**	**Application Example**
Low	30 - 300 KHz	125 - 134 KHz	Short Range Applications: Live Stock Identification, Antitheft Systems
High	3 - 30 MHz	13.56 MHz	Smart Card, Smart Card Label Applications: Baggage Tracking, Small Product Labeling
Very High	300 MHz - 3 GHz	U.S.A.: 902 - 928 MHz or 2.45 GHz, EU: 865 - 868 MHz or 5.8 GHz, Japan: 950 - 956 MHz, Korea: 908.5 - 914 MHz	Toll Collection Applications

considered as a single entity which has its own database and stores associated product information, transaction logs with a particular tag.

In many cases, it is assumed that the communication channel between readers and back-end severs is considered as a secure channel like the existing VPN or SSL.

Operating Frequencies

As a key interface for RFID system, operating frequency should be considered according to purpose of real world applications since each country has regulation of the electromagnetic spectrum. RFID tags and readers operate within several distinct frequency ranges, each of which is intended for specific application characteristics. According to the application purposes [33], table 2.2 shows the characteristics of frequencies that are available for RFID system.

2.1.3 Security and Privacy Issues

In the viewpoint of real world applications, the technical design of RFID readers and tags determines many risks [35]. A common technology is used in both retail [35, 38] and library applications [37]. Retail 915MHz tags can be read at ten times the distance (20-30 feet) of library 13.56MHz tags (2-4 feet). In addition, retail users of RFID will use the Electronic Product Code (EPC), a 96-bit number designed to uniquely label individual items. EPC users will have access to the EPC Discovery Service, an aggregate database of tag collected from independent readers. Anyone with access EPC Discovery can monitor or track the movement of a particular RFID-tagged item. Commercial information good producers will likely use the EPC format on their RFID tags.

Together with the high-lightening aspects of RFID technology, the existing RFID systems are vulnerable to many security risks and imply potential privacy problems, since it is very hard to implement the existing cryptographic algorithms due to the restricted computational power and the memory size of low-cost RFID tags [6, 7, 9, 14, 15, 18]. User privacy issue is considered as a big barrier for the proliferation of RFID system applications since the data of a tag can be transmitted by an illegal interrogation without its bearer's notification.

We mainly consider two privacy issues when using RFID. One is the data leakage illegally from a tag. Another is the malicious tracking for the unique ID of a tag [12]. A tag bearer has various objects that they do not want to make others know what they currently keep and what those objects are. If the tags are attached to those objects, the private information of tag bearers can be revealed regardless of their attention. The location privacy of tag bearers can be revealed through the response information from the tag although the tag information is securely protected. Especially, the location privacy can be more significant when a certain tag is exposed to the long-term tracking.

In RFID-labeled society, the value for commodities or products is mostly identified by the RFID. Thus, simple forgery such as copying information of a tag or even more sophisticated measure will be very attractive for the malicious users and the adversaries to disguise or impersonate [9]. From this reason, the forgery resistance is also strictly required for desirable usage of RFID.

2.1.4 Security Considerations

As generally required security aspects for RFID system, we can consider the followings [17, 25].

Confidentiality. RFID tags must not get involved in processing personal data. In addition to it, data stored in a tag should not be gathered to trace the relationship between the tag and the tag bearer by illegitimate readers. The private information of a tag must be kept secure to guarantee user privacy. The tag information must be meaningless for its bearer even though it is eavesdropped by an unauthorized reader.

Anonymity. Although a tag's data is encrypted, the tag's unique identification information is exposed since the encrypted data is constant. An attacker can identify each T with its constant encrypted data. Therefore, it is important to make the tag's information anonymous.

Integrity. Integrity in terms of RFID environment as a security requirement is usually for data integrity between entities, tags, readers, and back-end servers. Especially, not fault-tolerable is the air interface of communication channel and data synchronization between entities could be failed. Thus, integrity among entities must be guaranteed and data recovery mechanism should be provided in case data loss is occurred. In addition to it, if a tag's memory is rewritable, forgery is possible, so integrity for the tag's information also must be guaranteed.

We will revisit these aspects to define security requirements in section 3.

2.2 Cryptographic Background

2.2.1 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($\mathcal{H}$) should satisfy the following requirements:

Compression. Given an input x of arbitrary finite bitlength, $\mathcal{H}(x)$ maps to an output y of fixed bitlength n.

One-wayness. If $y = \mathcal{H}(x)$ 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 $\mathcal{H}(x) = y$ 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 $\mathcal{H}(x) = \mathcal{H}(x')$.

Collision-avoidance. It is computationally infeasible to find a pair (x, x') satisfying $\mathcal{H}(x) = \mathcal{H}(x')$.

Efficiency. Given an input x, $\mathcal{H}(x)$ 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 [26, 27], which is a powerful method to find colliding input pairs. So it is preferable that the output length of hash 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 [28]. In terms of low-cost RFID environment, [17, 36] 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 [14], authentication [29], micropayment [30] and auction [31], *etc.*

The generation of hash chain is done as follows:

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

The use of hash chain values is in reverse order, *i.e.* from $\mathcal{H}^n$ to $\mathcal{H}^1$. 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.

2.2.2 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 that outputs a sequence of 0s and 1s 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 computers are deterministic devices. 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 that 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 Related Works

2.3.1 Hash-lock Scheme

In 2003, Weis *et al.* [18] proposed two simple hash-based protocols, *hash-lock scheme* and *extended hash-lock scheme*. Both use a hash function which is enabled to implement by low-cost.

In hash-lock scheme, as shown in Figure 2.2, a back-end server stores keys k in its database for all tags and each tag has $metaID = h(k)$ for its key. When a reader queries access request to a tag, the tag transmits $metaID$ to

the reader as response. The reader sends $metaID$ received from the tag and the associated key k to the tag. At this moment, the tag itself calculates the hash value for the key from the reader and compares it with the stored $metaID$. If those two values are matched, the tag sends its own ID to the reader.

Hash-lock scheme uses $metaID$ as the unique ID of each tag for every read attempt. Thus, data privacy of tag bearers is protected and the protocol can meet confidentiality. However, $metaID$ is always constant so that attackers can eavesdrop it, identify each tag, and trace the tag. Therefore, location privacy of tag bearers is compromised.

2.3.2 Extended Hash-lock Scheme

Figure 2.3 shows *extended hash-lock scheme*. In extended hash-lock scheme [18], they proposed another method to overcome the tracing problem.

The difference is that a tag has a random number generator to make its constant variable randomized. Each tag has its own ID and random number generator. The tag picks pseudo random number r uniformly and calculates $c = hash(ID||r)$ as the tag's unique identification for every session. The tag transmits its c and r to a back-end server by way of the reader. The server finds the unique identifier of the tag comparing c with the construction

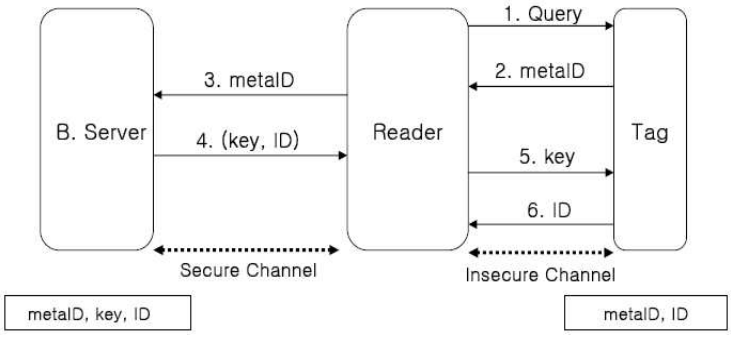

Figure 2.2: Hash-lock Scheme

of r and all IDs that is stored in database of the server, then the server authenticates itself by sending the unique identifier, ID back to the tag.

This scheme prevents tracing problem of tag bearers since the tag's output is switched randomly. This scheme is also strong for the replay attack. However, the tag can be traced if the tag's ID is exposed. In addition, an adversary can query a tag to get a tag's valid message pair (c, r). Later on, the attacker can impersonate that tag to a legitimate reader. The response from the reader will identify the tag. Also, the implementation issue for the random number generator is occurred.

2.3.3 Hash-based Varying Identifier

Another hash-based approach is *hash-based varying identifier* proposed by Henrici and Müller [7]. Their scheme also adopts a hash function and a random number generator, but a pseudo random number is generated by a back-end server and transmitted to the tag for every interrogation to make the tag's queried identifier random and preserve location privacy.

They assume the communication channel between tags and readers is insecure RF channel and secure is that between readers and back-end databases. A tag has only a unique identifer and remaining original data used for applications stored and controlled in a back-end database.

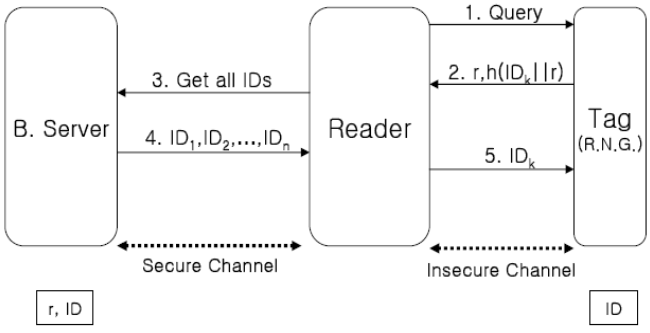

Figure 2.3: Extended Hash-lock Scheme

As shown in Figure 2.4, the overall system architecture is very similar to that of other previous works above. In their scheme [7], the DB-ID is set according to the database which will be in charge of the tag and ID, TID and LST of a tag are set to a random value initially. A corresponding row in the database ID, TID and LST that are same as the tag. Its HID is $h(ID)$ used as a primary index. The database manages a pair of record to guarantee message recovery for any data loss using AE fields which point each other.

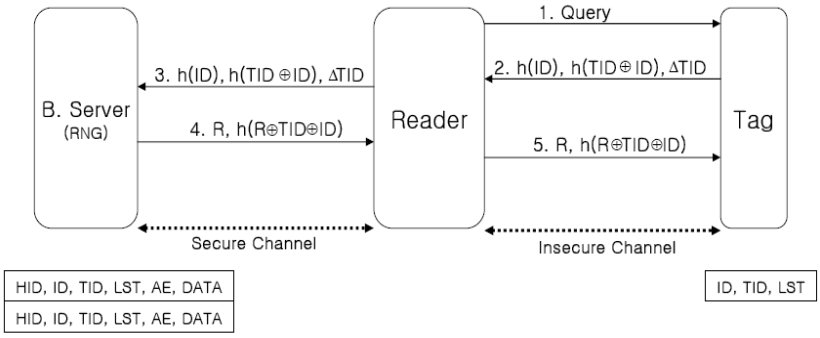

Figure 2.4: Hash-based Varying Identifier

Starting with the singularization of a tag, the protocol manages TID and LST to counteract the replay attacks. The tag output $h(TID \oplus ID)$ is changed in every read attempt. To authenticate the tag, $h(ID)$ is used as a primary index for the database to find a record comparing HID. In the last successful transaction, $\triangle TID$ is $\triangle TID = TID - LST$. After finding $HID = h(ID)$, the stored LST is achieved and the received $\triangle TID$ are added together [7]. This value is the current TID^* and now the hash $h(TID^* \oplus ID)$ is calculated. To do so the message from the tag can be authenticated comparing TID^* with stored TID. If the TID^* is not higher than the TID, the replay attack is in progress and the message is discarded [7].

If the authentication step is successful, the back-end server creates a random number RND and updates ID of the current record with a new $ID^* = RND \oplus ID$, then replies RND and $h(RND \oplus TID \oplus ID)$ with tag

data. Doing together with updating the current ID field with ID^* and HID with $h(ID^*)$, the back-end server manages the pair of records updating AE fields to reference to the other record and vice versa as well. The reader gets tag data and forwards the reply message to the tag. The tags can check whether the reply message is valid or not processing the same calculation $h(RND \oplus TID^* \oplus ID)$ from the new $ID^* = RND \oplus ID$ and TID. If it is correct, the tag also updates its ID and TID with $RND \oplus ID$ and its last successful transaction number, LST, respectively.

This scheme protects location privacy problem making a tag's ID randomized in every interrogation. However, location privacy of tag bearers is compromised since the response of tag is constant until the next authentication session. Adversaries can track tag bearers whose tags are long-distance from readers and scarcely have chance to be queried. Using $TIDs$ the replay attacks cannot compromise the scheme since tags and back-end servers are mutually authenticated in every interrogation. Errors in message transfer can be detected and the scheme is reliable for data loss since it can provides the data from the previous record. While authors claims that the scheme protects the man-in-the-middle attack, it can be compromised. The attacker can query any tag before the tag is interrogated by the legitimate reader, and he can be authenticated with the obtained data.

2.3.4 Improved Hash-based Varying Identifier

Hwang *et al.* [32] proposed an improved authentication protocol of *Hash-based Varying Identifier*. In their scheme, the main difference is that a reader has a random number generator (RNG) to protect the man-in-the-middle attack. Figure 2.5 shows the overall protocol of *Improved Hash-based Varying Identifier*.

In every query, the reader sends a pseudo-random number, S, to the tag. Then the tag replies $h(ID)$ for finding the record of a back-end server

and half of a new identifier, $half_L(R)$ ($R = h(ID||S)$. Then, the reader forwards $h(ID)$, $half_L(R)$, and S. In authentication phase in the back-end server, $h(ID)$ is used to find the corresponding record and ID is obtained. With stored ID and S received from the reader, the back-end server can calculate $R' = ID||S$ and the tag can be authenticated comparing $half_L(R')$ with $half_L(R)$ received from the tag. If the authentication is successful, the remaining job is updating ID of the record to a new $ID = R'$ and $h(ID)$ to $h(R')$, and then updating AE fields of the pair of record to reference each other. Then, the back-end server replies $half_R(R')$ with tag data to the tag by way of the reader. With $half_R(R)$, the tag can check whether the reply message is valid or not. If the process is successful, the tag and the back-end database updates its $ID \longleftarrow ID \oplus (R||R)$ since they assume the hash function of this protocol is $h : \{0,1\}^* \longrightarrow \{0,1\}^{\frac{1}{2}L}$ and R generated by this hash function is $\frac{1}{2}L$ bits.

This idea changing the location of a R.N.G. from a back-end server to a reader is cleaver and makes the protocol neat. Their proposed scheme needs only $1l$- field for a unique ID and its challenge and response phase uses a half length of R ($R = h(ID||S)$) so that its communication performance is more efficient than [7]. The scheme protects the location privacy as a tag's unique identifier is changed in every read attempts. The replay attacks

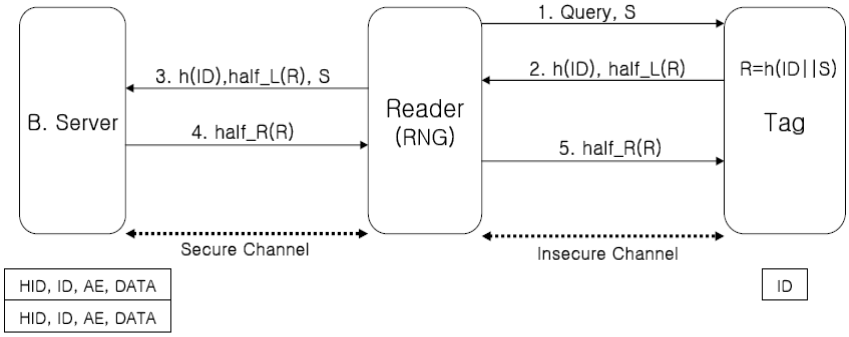

Figure 2.5: Improved Hash-based Varying Identifier

cannot compromise the scheme since tags and back-end servers are mutually authenticated.

However, this scheme is still vulnerable to the man-in-the-middle attack if a reader is not a trusted third party. An attacker can disguise as a legitimate reader, then he can interrogate any tag with a pseudo random number and the man-in-the-middle attack can be enabled. While they claim their scheme can protect location privacy of users, location privacy of tag bearers, similar to the problem of [7], is also compromised since the response of tag is constant until the next authentication session. Adversaries can track tag bearers whose tags are long-distance from readers and scarcely have chance to be queried.

In addition to it, both schemes, [7] and [32], did not denote the tag's ownership, so forgery is easily enabled with just passively attacks like eavesdropping a tag's response.

2.3.5 Other Approaches

Kill Command Approach

This method is initially suggested by MIT Auto-ID Center [1]. In this approach, each tag has its unique password of 8 bits and eases its functionality by itself when it receives its password. However, it is difficult to check whether kill command is successfully enabled or not. Another problem is that the applied method is restricted. In addition to them, attacker can determine the exact password of tag within computation of 2^8 since the length of password of each tag is only 8 bits.

One-time Pad based on XOR

This method [3] needs only XOR operation, so low computational cost is needed. A reader (or a back-end server) has the common list of randomly generated key for each tag. The reader and the tag find that both of them have the same key of the key list with several message exchanges between

them. Then, the tag transmits its ID to the reader. However, this method needs several message exchange for authentication between the tag and the reader. Besides, the common key list must be refreshed to guarantee the security. These are problems for implementation and system efficiency.

External Re-Encryption Scheme

This method [6] uses public key cryptosystem. Tag data is re-encrypted when a user requires using the data transferred from an external unit. As public key encryption needs high computation cost, a tag cannot process for itself. Thus, this job is generally processed by a reader. Each tag data is randomly shown until next session, the attacker eavesdropping the tag data cannot trace the tag for long-term period. However, this method has difficulty to frequently refresh each tag's data since the encrypted ID stored on tag is constant so that user location privacy is compromised. This job is processed by users (or tag bearers) and is considered impractical.

Hash Chain-based Scheme

Okubo *et al.* [14] proposed hash-chain based authentication protocol which protects users' location privacy and anonymity. They claims that their scheme provides strong forward security. However, hash-chain calculation must be burden on low-cost RFID tags and gives back-end servers heavy calculation loads.

Blocker Tag

This approach [4] uses a individual tag, namely *blocker tag* for each tag and according to its purpose. To protect a tag's data, the blocker tag responses for attacker's request to get the tag's data. The response from the blocker tag is not for the tag but all tags. Thus, the attacker cannot distinguishes the tag's data. This method basically uses binary tree walking protocol as

a collision-avoidance mechanism. Using the binary tree-based protocol, this method has advantages that the range of protecting tags can be efficiently specified into specific area of the binary tree. Doing so, the area of protecting tags is divided into *multiple privacy zone* and the performance of tree walking can be efficient. This method also provides *zone policy* to apply protection policy according to various purposes. This method is currently considered as a practical solution for the existing RFID privacy and security protection. Problem is that additional blocker tag is needed for every tag and it is susceptible whether tag bearers strictly follow to attach the additional tag.

2.3.6 Comparison of Related Works

Table 2.3 shows comparison between previous results according to their cryptographic feature, their advantages, and disadvantages that is described in the previous section. We denote hash-lock scheme by [18]-A, extended hash-lock scheme by [18]-B, and other schemes by the number of reference, respectively.

Related works are schemes for security in low-cost RFID environment, but each approach is different. Those scheme are compared with each other in the aspects of data protection, tracking prevention, and forward security. Under constrained resources of low-cost RFID tag, data protection simply does not mean confidentiality but imply data privacy of tag bearers since most of schemes just use identification information when challenging and responding during authentication or identification.

Tracking prevention also implies protection location privacy of tag bearers. Concerning this goal, some of them [3, 6, 7, 14, 18, 18, 32] try to provide tag anonymity using cryptographic primitives such as hash, hash-chain, or random number generator, but others [1, 4] apply simple way based on some characteristics of RFID interfaces such as RFID tag's command or tag singularization.

Forward security means security of the past is preserved even after the

Table 2.3: Comparison between Schemes

Schemes	Data Protection	Tracking Prevention	Forward Security	Required Computation
[1]	△	◯	-	-
[3]	◯	◯	×	XOR
[4]	◯	◯	-	-
[6]	◯	◯	◯	-
[7]	◯	△	◯	hash, RNG, XOR
[14]	◯	◯	◯	hash
[18]-A	◯	×	×	hash
[18]-B	◯	◯	×	hash, RNG
[32]	◯	△	◯	hash, RNG, XOR

†† Notation

◯ satisfied △ partially satisfied

× not satisfied - not required

secrets or keys has been exposed. Once the secret in the tag is stolen, all past activities can be traced by searching past logs. Forward security ensures that the latest memory in the tag does not give a hint to guess past outputs [14]. Thus, the past activities can be protected from tampering. [1, 4] do not need to consider this aspect since they do not use a session key or a secret for every session. Rather, their purpose is to make tag's functionality stop permanently or temporarily. Except for them, most schemes consider forward security. More security aspects and requirements will be discussed in the next section.

Chapter 3
Proposed Scheme

In this section, we propose a RFID authentication protocol that guarantees mutual authentication in each session and the privacy of tag bearers.

At first, we define some assumptions and the required security goals in RFID 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.

3.1 Main Idea

Basically, security vulnerabilities in RFID systems are resulted from the asymmetric communication channel of air interface between tags and readers. It can violate synchronization of tag's identification information between tags and back-end servers. The replay attack is also enabled for tags and back-end servers, respectively. Moreover, adversaries can trace location and behaviors of tag bears without detection and impersonate a legitimate tag or a legitimate reader. To remove these security and privacy problems, our protocol is based on mutual authentication between tags and back-end servers guaranteeing freshness of tag's identification information.

In addition, we firstly assume the communication channel between a reader and a back-end server is insecure. Different from the previous works, a reader is no more a trusted third party. An attacker can disguise as a legitimate reader. Thus, in our protocol, a reader is also authenticated by a back-end

server. The reader generates a random seed and transmits it to make tags anonymous when querying tags. This random seed is integrated into the tag's authentication message to protect the man-in-the-middle attack.

In our protocol, tags do not have any real data and all authentication messages are random, hashed, and encrypted, so confidentiality is guaranteed even though the authentication messages are eavesdropped by adversaries.

We use the notations as defined in *List of Notations* to describe the protocol throughout the thesis. Like [7], we adopt the similar database structure and the same mechanism to prevent the data loss. A back-end server has a database D and manages a pair of records for each tag in case the reply message from the back-end server to the tag is lost or intercepted. Each record consists of fields like $\langle T_1, T_2, AE, CN, DATA \rangle$. The detailed descriptions of the fields are also shown in *List of Notations*.

3.2 Assumptions

3.2.1 General Assumptions

Our protocol works with the natural assumption that $\mathcal{T}$ has a hash function, XOR gate, and the capability to keep state during a single session. The widely acceptable low-cost RFID tags would most likely require the usage of passive tags [15, 18]. To design our proposed protocol, we assume the low-cost RFID tag is passive and has a re-writable memory like EEPROM with reasonable size like EPC Class 2 of EPC Global [17]. In Crypto 2004, Biham *et al.* [2, 8, 19] showed that collision of SAH0, MD4, MD5, HAVAL-128, and RIPEMD in a special case is easily found. With this in mind, we expect that the cryptographic hash function used in our protocol has the desirable security like preimage resistance, second preimage resistance, and collision avoidance. In our protocol, we assume $\mathcal{T}$ has a hash function. In [13], 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 RFID tag with the desirable security. Like [7, 14], we assume that $\mathcal{T}$ only has its authentication related information. A tag also has a memory for keeping values of ID, k_1, and k_2 to process mutual authentication. The simple structures for the record of D and the tag memory are shown in Figure 3.3. Other required data of $\mathcal{T}$ for an application are stored in D of $\mathcal{B}$.

In the previous schemes [7, 18], they assumed $\mathcal{R}$ is a TTP and the communication channel between $\mathcal{R}$ and $\mathcal{B}$ is secure. However, we assume that $\mathcal{R}$ is not a TTP and the communication channel is insecure like the today's wireless network. With this assumption, their schemes are easily compromised with the man-in-the-middle attack. To verify the validity of $\mathcal{R}$, $\mathcal{R}$ has a RNG, $r \in_U \{0,1\}^l$, and both $\mathcal{R}$ and $\mathcal{B}$ have, $h_k()$, a keyed one-way hash function, $h_k : \{0,1\}^* \rightarrow \{0,1\}^l$. To secure $DATA$ in the reply message from $\mathcal{B}$, $\mathcal{B}$ encrypts, $E_{h_k(S)}(DATA)$, and $\mathcal{R}$ decrypts, $D_{h_k(S)}(DATA)$. $h_k(S)$ is the secret key and is randomized for each session with the random number r from $\mathcal{R}$. We assume that k is the secret key shared between $\mathcal{R}$ and $\mathcal{B}$, and $\mathcal{R}$ and $\mathcal{B}$ has enough capability to manage the symmetric-key cryptosystem and sufficient computational power for encryption and decryption.

Figure 3.1 shows overall system architecture and message exchanges based on the assumptions.

3.2.2 Attacking Model

To solve the security risks and privacy issues, the following attacking model must be assumed and prevented [7, 15, 17, 18]. However, in our protocol, we do not consider a physical attack like detaching RFID tag physically from a product because it is hard to carry out in public or on a wide scale without detection. We consider the following attacks and describe:

Man-in-the-middle attack : The attackers can impersonate as a legitimate reader and get the information from $\mathcal{T}$, so he can impersonate as the legitimate $\mathcal{T}$ responding to $\mathcal{R}$. Thus, the attacker easily can be authenticated by the legitimate $\mathcal{R}$ before the next session.

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

Forgery : The simple copy for the information of $\mathcal{T}$ by eavesdropping is enabled by the adversary.

Data loss : The protocol can be damaged from the denial-of-service (DoS) attack, power interruption, and hijacking.

3.3 Security Requirements

To guarantee security and protect the privacy of tag bearers, we define the following requirements in cryptographic point of view [17, 14].

Data Confidentiality : The private information of $\mathcal{T}$ must be kept secure to

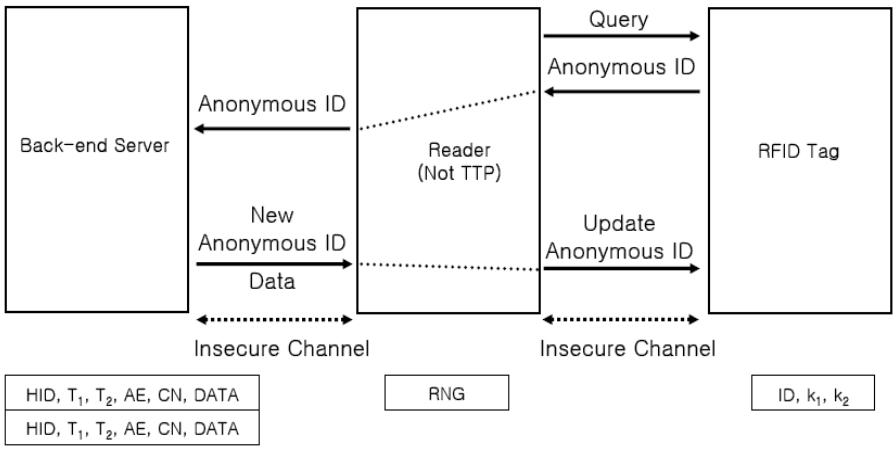

Figure 3.1: Overall architecture of our RFID system

guarantee user privacy. The information of $\mathcal{T}$ must be meaningless for its bearer even though it is eavesdropped by an unauthorized $\mathcal{R}$.

Tag Anonymity : Although the data of $\mathcal{T}$ is encrypted, the unique identification information of $\mathcal{T}$ is exposed since the encrypted data is constant. An attacker can identify each $\mathcal{T}$ with its constant encrypted data. Therefore, it is important to make the information of $\mathcal{T}$ anonymous.

Data Integrity : If the memory of $\mathcal{T}$ is rewritable, forgery and data modification will happen. Thus, the linkage between the authentication information and $\mathcal{T}$ itself must be given in order to prevent the simple copy for $\mathcal{T}$. In addition to this, the data integrity for the authentication information between $\mathcal{T}$ and $\mathcal{B}$ must be guaranteed. On the other hand, there is the possible data loss coming from the DoS attack, power interruption, message hijacking, *etc.* Thus, the authentication information between $\mathcal{T}$ and $\mathcal{B}$ must be delivered without any failure, and the data recovery must be provided.

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

Mutual authentication and reader authentication : In addition to access control, the mutual authentication between $\mathcal{T}$ and $\mathcal{B}$ must be provided as a measure of trust [11]. By authenticating mutually, the replay attack and the man-in-the-middle attack to both $\mathcal{T}$ and $\mathcal{B}$ is prevented. $\mathcal{B}$ also must authenticate $\mathcal{R}$ to avoid the man-in-the-middle attack by an illegitimate $\mathcal{R}$ on the insecure channel.

3.4 Design of Authentication Protocol

First of all, we design the basic authentication protocol that provides mutual authentication between a tag and a back-end server as shown in Figure 3.2.

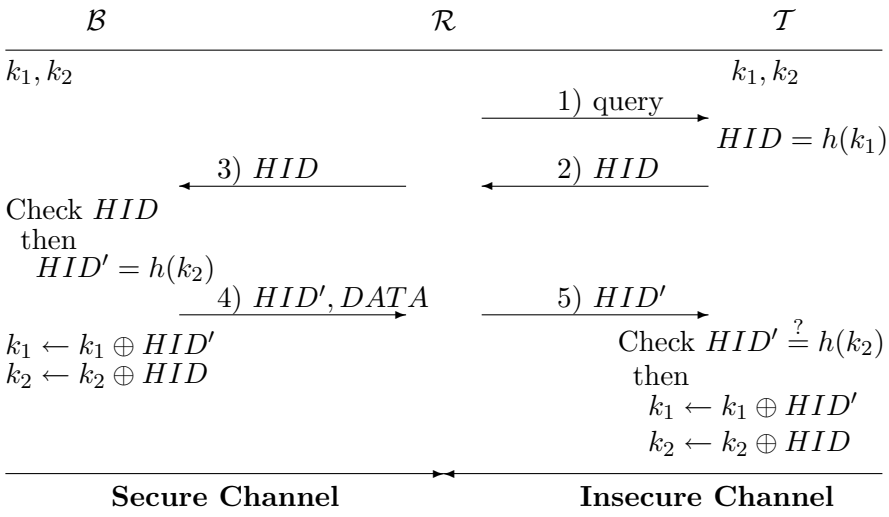

Figure 3.2: Basic Authentication protocol

Initially, the tag and the back-end server store two fresh random nonce, k_1 and k_2 as shared secrets. The basic protocol yet counteracts attacks, but it provides mutual authentication between a tag and a back-end server and anonymity on the tag. The tag has a hash function and transmit HID as a reply for a query from a reader. In the basic protocol, HID is used as a index so that the back-end server can find the record corresponding the tag and calculate $h(k_1)$. The tag response is forwarded by the reader. When the back-end server succeed to authenticate the tag, it sends HID' and corresponding data as a reply to the reader and update the shared two secret information, $k_1 = k_1 \oplus HID'$ and $k_2 = k_2 \oplus HID$, respectively. To make the response anonymous, the back-end server uses $HID' = h(k_2)$. Then, the reader forwards the authentication reply to the tag. The tag can authenticate HID' with its shared secret k_2. After finishing mutual authentication successfully, the tag updates the shared two secret information, $k_1 = k_1 \oplus HID'$ and $k_2 = r_2 \oplus HID$, respectively.

We adapt RNG on $\mathcal{R}$ to protect the man-in-the-middle attack. In this

protocol, $\mathcal{R}$ generates a pseudo random number, r, and queries with $S = h_k(r)$ to $\mathcal{T}$. $\mathcal{R}$ also transmits S and r to $\mathcal{B}$ with the response message from $\mathcal{T}$. S is the keyed hash value from $\mathcal{R}$ and is verified by $\mathcal{B}$. Thus, the protocol can prevent the man-in-the-middle attack even though an attacker can query $\mathcal{T}$ before $\mathcal{T}$ is interrogated by the legitimate $\mathcal{R}$ and it can be authenticated with its corresponding response.

To make this protocol forgery resistant, we exclusive-or a RFID chip's original serial number in ID. During the authentication process in $\mathcal{B}$ side, $\mathcal{B}$ can check ID with the chip serial number that is stored on the corresponding record of D. The overall protocol is shown in Figure 3.3. The subordinate procedures for each step are described.

3.4.1 Protocol Setup

1) Each $\mathcal{T}$ is given two fresh random secrets and D of $\mathcal{B}$ also stores them as the shared secret. The temporary used two shared secrets are k_1 and $k_2 \in_U \{0,1\}^l$. $\mathcal{T}$ has a hash function and a XOR function. $\mathcal{T}$ does not need to have the additional storage for its serial number, C, since C is unique and permanently embedded into each $\mathcal{T}$ [12]. k_1, k_2, and the initial identification data, $ID = h(k_1)$, are initially stored into ID, k_1, and k_2 of memory of each $\mathcal{T}$, respectively.

2) $\mathcal{R}$ has a RNG and a keyed hash function, generates a fresh random nonce, $r \in_U \{0,1\}^l$, and calculates $h_k(r)$ for every session. $\mathcal{R}$ and $\mathcal{B}$ manage the secret key k for keyed hash function. We simply denote $h_k(r)$ by S.

3) $\mathcal{B}$ has a database D and manages a record pair for each tag consisting of $\langle T_1, T_2, AE, CN, DATA \rangle$ like [7]. AE is not set since no associated entry exists initially at this moment. CN, keeps the unique chip serial number, C, for each $\mathcal{T}$. $\mathcal{B}$ has a hash function and a keyed hash function

to verify $\mathcal{T}$ and $\mathcal{R}$, respectively. The pair of records point each other with the pointer field, AE. Thus, the record for the previous session can be used to recover the data for the current session when the data loss is occurred.

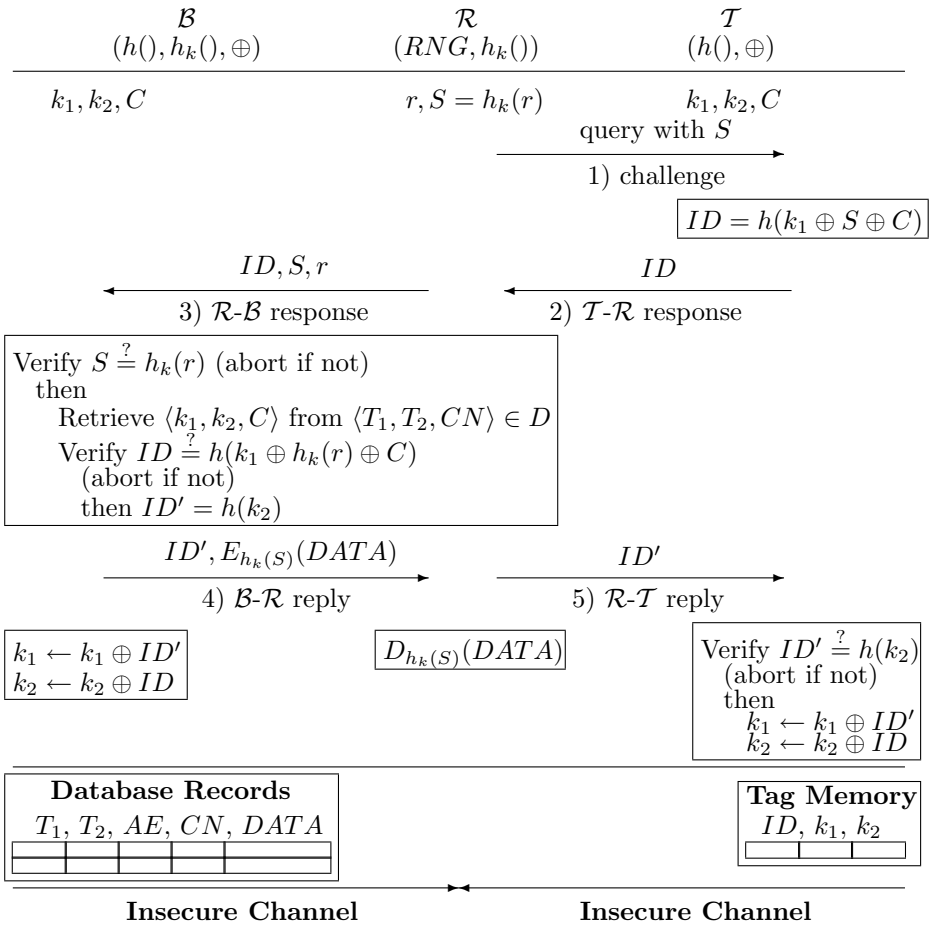

Figure 3.3: Proposed Authentication Protocol

3.4.2 Detailed Description

As shown in Figure 3.3, we describe the proposed protocol according to the sequence of message exchange and also discuss the security goals which can be achieved during the execution of each protocol message. The one session of mutual authentication is processed from step 1 to step 5 challenging and responding the valid authentication messages.

Step 1 (Challenge) In this step, $\mathcal{R}$ usually applies a collision-avoidance protocol like the secure binary tree walking [4, 17] or the standard protocols of ISO 18000-3 MODE [10] to singularize $\mathcal{T}$ out of many. $\mathcal{R}$ generates a fresh random nonce, r, and randomizes it with the keyed one-way hash function, $S = h_k(r)$. $\mathcal{R}$ sends S to the queried $\mathcal{T}$. The key, k, is shared by $\mathcal{R}$ and $\mathcal{B}$, and S is used to authenticate the validity of $\mathcal{R}$. With S, the man-in-the-middle attack is prevented against an active attacker. It is also used to detect the illegitimate $\mathcal{R}$ by $\mathcal{B}$ after step 3.

Step 2 ($\mathcal{T}$-$\mathcal{R}$ Response) When queried, $\mathcal{T}$ sends ID to $\mathcal{R}$. ID is the output of one-way hash function. ID is used as the identification information and has two purposes; One is to verify the legitimate $\mathcal{R}$ with S, and another is to prevent the forgery with C by the passive eavesdropping. ID is randomized with the shared secrets, k_1 and k_2 for every read attempt.

Step 3 ($\mathcal{R}$-$\mathcal{B}$ Response) $\mathcal{R}$ simply forwards ID to $\mathcal{B}$. At the same time, $\mathcal{R}$ also transmits S and r to prevent the man-in-the-middle attack and to detect the illegal $\mathcal{R}$. Within this step, $\mathcal{B}$ authenticates $\mathcal{R}$ and $\mathcal{T}$ consequently with ID, respectively.

- At first, $\mathcal{B}$ verifies whether the forwarded r is valid or not by comparing S with $h_k(r)$. k is the shared secret key only between $\mathcal{R}$ and $\mathcal{B}$, so $\mathcal{B}$ can detect the illegal $\mathcal{R}$ and discards the forwarded

message. Thus, the man-in-the middle attack by the illegitimate $\mathcal{R}$ and a passive eavesdropper can be prevented.

- If $\mathcal{R}$ is valid, $\mathcal{B}$ retrieves the record corresponding to ID and get $\langle k_1, k_2, C\rangle$ from $\langle T_1, T_2, CN\rangle \in D$, respectively. Then, $\mathcal{B}$ authenticates $\mathcal{T}$ with ID. $\mathcal{B}$ calculates $h(k_1 \oplus h_k(r) \oplus C)$ and compares with ID.
- Since $\mathcal{B}$ initially stores the chip serial number, C, $\mathcal{B}$ can evaluate the linkage between the forwarded authentication information ID and $\mathcal{T}$ itself in order to prevent forgery. Forgery can be detected and prevented by $\mathcal{B}$ at this moment.
- At the same time, $\mathcal{B}$ can detect and prevent the man-in-the-middle attack since S is used as the factor of the man-in-the-middle attack detection. Similarly, the replay attack can be also detected and prevented simultaneously.
- If $\mathcal{B}$ successfully finishes the authentication process, $\mathcal{B}$ generates ID' with its one of shared random secrets k_2. ID' will be used to make the shared secret, k_1, anonymous in the remaining steps.
- D of $\mathcal{B}$ generates a new record to consist of a pair of records and updates with the corresponding record. AE have the value to point the pair of records each other. When errors or the data loss in message for the current session are occurred, D of $\mathcal{B}$ can refer the record of the previous session pointed by AE of the current session. Thus, the protocol is reliable for the data recovery against the data loss.

Step 4 ($\mathcal{B}$-$\mathcal{R}$ Reply) $\mathcal{B}$ encrypts the corresponding $DATA$ using $h_k(S)$, the randomly created shared secret key between $\mathcal{B}$ and $\mathcal{R}$. Then, $\mathcal{B}$ replies ID' and $E_{h_k}(S)(DATA)$. Then, $\mathcal{B}$ makes its shared two keys, k_1 and k_2, randomized simply by exclusive-oring. The same process will be

applied to the next step for making the corresponding shared secrets of $\mathcal{T}$ anonymous. After this step, the corresponding decryption process, $D_{h_k}(S)(DATA)$, is processed by $\mathcal{R}$ to get $DATA$. Thus, $DATA$ of $\mathcal{T}$ is securely obtained only by the legitimate $\mathcal{R}$ although the adversary eavesdrops the reply messages on the insecure channel.

Step 5 ($\mathcal{R}$-$\mathcal{T}$ Reply) Like step 3, $\mathcal{R}$ forwards ID' to the corresponding $\mathcal{T}$. Then, $\mathcal{T}$ processes the mutual authentication. $\mathcal{T}$ verifies the forwarded ID'. $\mathcal{T}$ calculates $h(k_2)$ and compares it with ID'. If matched, the mutual authentication is finally succeeded, and $\mathcal{T}$, as the last process, updates the shared secrets k_1 and k_2 simply exclusive-oring with ID and ID', respectively. Otherwise, $\mathcal{T}$ will not updates them in case the replay attack to $\mathcal{T}$ occurs.

Chapter 4
Analysis

4.1 Correctness Proof

In this section, we prove the correctness of the proposed protocol based on GNY logic. Specifically, the correctness means that after the protocol execution, the communication parties believe that they are sharing two fresh secrets, k_1 and k_2, and random nonce, r, and are sure that this belief is confirmed by the other side. In addition to this, two entities, $\mathcal{R}$ and $\mathcal{B}$ should believe that they share the secret keys in case the communication channel between the two entities is insecure. The logic of Gong, Needham and Yahalom [20], usually referred to as the GNY logic, is used to formally verify cryptographic protocols. we apply the reasoning process of GNY logic to our protocol.

In the forthcoming description, we use the conventional notations as follows: T, R, and B are entities, $\mathcal{T}$, $\mathcal{R}$, and $\mathcal{B}$, respectively; K_1^i and K_2^i are shared secrets for i-th session between $\mathcal{T}$ and $\mathcal{B}$. $H()$ is a one-way hash function and $H_K()$ is a one-way keyed hash function; N_R is a random nonce generated by $\mathcal{R}$; K is a shared secret for $H_K()$ and K_{RB} is a shared secret for conventional encryption; m is data; other notations like T1, P1, F1, *etc.* follow the logical postulates of GNY logic [20].

4.1.1 Used GNY Constructs

Table 4.1 shows the constructs of GNY that are used throughout the thesis. For a more detailed and complete description we refer to [20].

4.1.2 Protocol Formalization

In this section, we, at first, simplify the protocol and describe it as a generic type. Then, we idealize and formalize the generic type of the protocol for verification goals.

Generic Type of Protocol.

1) Message 1 $R \rightarrow T : H_K(N_R)$

2) Message 2 $T \rightarrow R : H(K_1^i \oplus H_K(N_R)), H(K_1^i \oplus H_K(N_R) \oplus C)$

3) Message 3 $R \rightarrow B : H(K_1^i \oplus H_K(N_R)), H(K_1^i \oplus H_K(N_R) \oplus C), H_K(N_R), N_R$

4) Message 4 $B \rightarrow R : H(K_2^i), \{m\}_{K_{RB}}$

5) Message 5 $R \rightarrow T : H(K_2^i)$

Idealized Protocol.

Authentication messages of the proposed protocol consist of the outputs of hash function or keyed hash function. Two fresh shared secrets between T and B are updated for every session and used as session keys for mutual authentication between two entities. The authentication messages are transferred by way of legitimate R. The validity for R should be guaranteed by B with shared keys since we assume R is not a TTP and the communication channel between R and B is insecure. Therefore, we add some conditions for those keys to the generic type of protocol and omit unnecessary components. We follow the authentication steps and verify whether the two parties, T and

Table 4.1: Used GNY constructs

(X, Y)	Concatenation of formulae	$\{X\}K$ $\{X\}K^{-1}$	Symmetric-key encryption and decryption
$P \ni X$	P possesses or is capable of possessing formula, X	$P \mid\sim X$	P conveyed X.
$P \models\!\equiv X$	P believes X.	$\sharp(X)$	The formula X is fresh. X has not been before the current run of the protocol.
$P \triangleleft X$	P is told X. P has a received a message containing X and P can read and repeat X.	$P \triangleleft\!\star (X)$	P is told formula X, not conveyed by P during the current protocol run.
$X \rightsquigarrow C$	Message X has the extension C. The precondition for X being conveyed is C.	$P \Rightarrow X$	P has jurisdiction over X. The principal P is an authority on X.
ϕX	Formula X is recognizable	$P \overset{K}{\longleftrightarrow} Q$	K is a suitable secret for P and Q. It may be used as a key or as a proof of identity.
$P \overset{K}{=\!=} Q$	K is a secret known only to P and Q, and possibly to principals trusted by them. Only P and Q may use X to prove their identities to one another. Often, K is fresh as well as secret.		

B, can believe each other that they send and reply the shared secrets, K_1 and K_2 with each other.

1) Message 1 $R \rightarrow T : H_K(N_R) \rightsquigarrow R \models R \overset{K}{\leftrightarrow} B$

2) Message 2 $T \rightarrow R : H(K_1^i \oplus H_K(N_R)) \rightsquigarrow T \models \phi(H(X))$

3) Message 3 $R \rightarrow B : H(K_1^i \oplus H_K(N_R)) \rightsquigarrow B \models R \overset{K}{\leftrightarrow} B$

4) Message 4 $B \rightarrow R : H(K_2^i), \{R \overset{K_{RB}}{\longleftrightarrow} B\}_{K_{RB}} \rightsquigarrow B \models R \overset{K_{RB}}{\longleftrightarrow} B$

5) Message 5 $R \rightarrow T : H(K_2^i) \rightsquigarrow T \ni K_2^i$

Formalized Protocol.

The conventional notations above is not convenient for manipulation in a logic [21]. To process the formal proof, we introduce the logical formula for our authentication protocol that is idealized version of the original message. The asterisks, $\star$, denote the ability of each principal to recognize that it did not send the received message at an earlier stage in the protocol, which means the following term was not originated by the party who receives it. The message extension, $\rightsquigarrow$, means that K and K_{RB} are intended to be shared secret keys for use between entities, R and B.

1) Message 1 $T \triangleleft \star(H_K(N_R)) \rightsquigarrow R \models R \overset{K}{\leftrightarrow} B$

2) Message 2 $R \triangleleft \star(H(K_1^i \oplus H_K(N_R))) \rightsquigarrow T \models \phi(H(X))$

3) Message 3 $B \triangleleft \star(H(K_1^i \oplus H_K(N_R))) \rightsquigarrow B \models R \overset{K}{\leftrightarrow} B$

4) Message 4 $R \triangleleft \star(H(K_2^i), \{R \overset{K_{RB}}{\longleftrightarrow} B\}_{K_{RB}}) \rightsquigarrow B \models R \overset{K_{RB}}{\longleftrightarrow} B$

5) Message 5 $T \triangleleft \star(H(K_2^i)) \rightsquigarrow T \ni K_2^i$

Table 4.2: Goals of the Correctness Proof

1. $B \models T \mid\sim \sharp(H(K_1^i \oplus H_K(N_R)))$	2. $T \models B \mid\sim \sharp(H(K_2^i))$
3. $R \models R \overset{K}{\leftrightarrow} B$	4. $B \models R \overset{K}{\leftrightarrow} B$
5. $R \models R \overset{K_{RB}}{\leftrightarrow} B$	6. $B \models R \overset{K_{RB}}{\leftrightarrow} B$

4.1.3 Proof Goals

T transmits its fresh hashed output to B for every session using two fresh shared secrets, K_1^i and K_2^i between T and B. B replies its fresh hashed output to T also using two fresh shared secrets, K_1^i and K_2^i between T and B. In addition to them, K and K_{RB} are shared keys between R and B. The proof goals of correctness are shown in Table 4.2.

The first two goals, (1) and (2), are for the shared secrets. Those beliefs is to state that two entities shared secrets each other exchange fresh messages. In the real world, the output from tag should be anonymous. Thus, it is hashed and transmitted to back-end server, then verified whether the tag is valid or not. The message from the back-end server to the tag is also hashed and verified by the tag for mutual authentication. The goals (3-6) are about shared keys between two entities. (3) and (4) are for a keyed hash function to guarantee the validity of reader, and (5) and (6) are for message encryption and decryption based on the symmetric key cryptosystem. Those beliefs stat that two entities, a reader and a back-end server, share those keys for their own purpose.

4.1.4 Initial Assumptions

Table 4.3 shows the initial assumptions for our protocol. Assumptions (1-4) state that $\mathcal{T}$ has a hash function, $\mathcal{B}$ has a hash functions and a keyed hash function, $\mathcal{R}$ has a RNG and a keyed hash function, and the random nonce N_R of $\mathcal{R}$ and the keyed hash value $H_K(N_R)$ are fresh. The next six assump-

Table 4.3: Initial Assumptions for Proof

1. $T \ni H(X)$	2. $R \ni H_K(X)$
3. $B \ni (H(X), H_K(X))$	4. $T \models \sharp(N_R)$
5. $T \ni (K_1^i, K_2^i)$	6. $B \ni (K_1^i, K_2^i)$
7. $T \models \sharp(K_1^i, K_2^i)$	8. $B \models \sharp(K_1^i, K_2^i)$
9. $T \models T \overset{K_1^i, K_2^i}{\leftrightharpoons} B$	10. $B \models T \overset{K_1^i, K_2^i}{\leftrightharpoons} B$
11. $T \models B \ni (K_1^i, K_2^i, C)$	12. $B \models T \ni (K_1^i, K_2^i, C)$
13. $T \models B \mapsto T \overset{K_1^i}{\rightleftharpoons} B$	14. $T \models \sharp(H(K_2^i))$
15. $T \models R \mapsto B \mid\sim H(K_2^i)$	16. $B \models T \mapsto T \overset{K_2^i}{\rightleftharpoons} B$
17. $R \ni (K, K_{RB})$	18. $R \models R \overset{K, K_{RB}}{\longleftrightarrow} B$
19. $B \ni (K, K_{RB})$	20. $B \models R \overset{K, K_{RB}}{\longleftrightarrow} B$
21. $B \models R \mapsto R \overset{K, K_{RB}}{\longleftrightarrow} B$	22. $R \models B \mapsto R \overset{K, K_{RB}}{\longleftrightarrow} B$

tions (3-8) are for two fresh shared secrets, K_1 and K_2, between $\mathcal{T}$ and $\mathcal{B}$, which means those shared secrets are fresh and are kept in both entities, $\mathcal{T}$ and $\mathcal{B}$, during the single session. Assumptions (9) and (10) are based on the assumptions (1-8) and $\mathcal{R}$ must be a trusted entity in the viewpoint of $\mathcal{B}$ since the authentication messages from $\mathcal{T}$ are transmitted via $\mathcal{R}$. The abilities for verifying the hashed authentication message transmitted from $\mathcal{T}$ by $\mathcal{B}$ and from $\mathcal{B}$ by $\mathcal{T}$, respectively are based on assumptions (11-14). Assumptions (15-20) mean that K and K_{RB} are shared secrets between entities, $\mathcal{R}$ and $\mathcal{B}$, and are managed by both entities. K_{RB} is for the symmetric key cryptosystem for message encryption, and K is for the keyed hash function. Those assumptions also mean that both entities trust each other with those keys.

4.1.5 Verification

In this section, the formal proof of our protocol is stated. The proof based on GNY logic is processed with the assumptions of Table 4.3. We strictly follow the logical postulates of [20]. We refer n is the number of list and denote the list of proof goals of Table 4.2 by Gn, the list of assumptions of Table 4.3 by An, and the verification steps by Vn.

Message 1 $T \triangleleft \star(H_K(N_R)) \rightsquigarrow R \models R \overset{K}{\leftrightarrow} B$

1) The extension to the message, $R \models R \leftrightarrow KB$, is the precondition for the shared key K between R and B. It is valid because it holds when the message is sent as is evident from the initial assumptions, A18.

2) Applying T1, Being-Told Rule.
   $T \triangleleft H_K(N_R)$

3) Applying P1, Possession Rule.
   $T \ni H_K(N_R)$

4) Applying F1, Freshness Rule.
   $$\frac{T \models \sharp(N_R)}{T \models \sharp(H(N_R))}$$

5) Applying V3, and applying F10, Freshness Rule.
   $$\frac{T \models \sharp(H_K(N_R)) \wedge T \ni H_K(N_R)}{T \models \sharp(H_K(N_R))}$$

Message 2 $R \triangleleft \star(H(K_1^i \oplus H_K(N_R))) \rightsquigarrow T \models \phi(H(X))$

6) The extension to the message, $T \models \phi(H(X))$, is valid since it holds when the message is sent as is evident from the initial assumptions, A1. The followings show the validity of it.

   - Applying R6, Recognizability Rule.
     $$\frac{T \ni H(X)}{T \models \phi(X)}$$

- Applying R5, Recognizability Rule.
$$\frac{T \models\!\!\models \phi(X) \wedge T \ni X}{T \models\!\!\models \phi(H(X))}$$

7) Applying T1, Begin-Told Rule.

$R \triangleleft H(K_1^i \oplus H_K(N_R))$

8) Applying P1, Possession Rule.

$R \ni H(K_1^i \oplus H_K(N_R))$

9) Applying F10, Freshness Rule.
$$\frac{R \models\!\!\models \sharp(K_1^i \oplus H_K(N_R)) \wedge R \ni H(K_1^i \oplus H_K(N_R))}{R \models\!\!\models \sharp(H(K_1^i \oplus H_K(N_R)))}$$

10) For V9, applying R6, Recognizability Rule.
$$\frac{R \ni H(K_1^i \oplus H_K(N_R))}{R \models\!\!\models \phi(H(K_1^i \oplus H_K(N_R)))}$$

11) For V9, applying A18, V7, V8, V10, and applying I1, Message Interpretation Rule.
$$\frac{R \triangleleft \star(H(K_1^i \oplus H_K(N_R))) \wedge R \ni (K_1^i \oplus H_K(N_R)) \wedge R \models\!\!\models \sharp(H(K_1^i \oplus H_K(N_R))) \wedge R \models\!\!\models \phi(H(K_1^i \oplus H_K(N_R)))}{R \models\!\!\models \sharp(H(K_1^i \oplus H_K(N_R)))}$$

12) Applying A20, A22, and applying J1, Jurisdiction Rule. This is the proof for G3, $R \models\!\!\models R \overset{K}{\longleftrightarrow} B$.
$$\frac{P \models\!\!\models B \mapsto R \overset{K}{\longleftrightarrow} B \wedge B \models\!\!\models R \overset{K}{\longleftrightarrow} B}{R \models\!\!\models R \overset{K}{\longleftrightarrow} B}$$

Message 3 $B \triangleleft \star(H(K_1^i \oplus H_K(N_R))) \rightsquigarrow B \models\!\!\models R \overset{K}{\longleftrightarrow} B$

13) The extension to the message, $B \models\!\!\models R \overset{K}{\longleftrightarrow} B$, is valid because it holds when the message is sent as is evident from the initial assumptions, A20.

14) Applying T1, Being-Told Rule.

$B \triangleleft H(K_1^i \oplus H_K(N_R))$

15) Applying P1, Possession Rule.

$B \ni H(K_1^i \oplus H_K(N_R))$

16) For V15, applying A3, A6, and applying F10, Freshness Rule.
$$\frac{B \models\!\!\models \sharp(K_1^i \oplus H_K(N_R)) \wedge B \ni H(K_1^i \oplus H_K(N_R))}{B \models\!\!\models \sharp(H(K_1^i \oplus H_K(N_R)))}$$

17) For V15, applying R6, Recognizability.

$$\frac{B \ni H(K_1^i \oplus H_K(N_R))}{B \models \phi(H(K_1^i \oplus H_K(N_R)))}$$

18) For V16, applying A3, A6, A20, V14, V16, V17, and applying I1, Message Interpretation Rule.

$$\frac{B \triangleleft \star(H(K_1^i \oplus H_K(N_R))) \wedge B \ni (K_1^i, H_K(N_R)) \wedge B \models R \overset{K}{\longleftrightarrow} B \wedge B \models \sharp(H(K_1^i \oplus H_K(N_R))) \wedge B \models \phi(H(K_1^i \oplus H_K(N_R)))}{B \models R \mid\sim H(K_1^i \oplus H_K(N_R))}$$

19) For assumptions, A18 and A21, applying J1, Jurisdiction Rule. This is the proof for G4, $B \models R \overset{K}{\longleftrightarrow} B$.

$$\frac{B \models R \mapsto R \overset{K}{\longleftrightarrow} B \wedge R \models R \overset{K}{\longleftrightarrow} B}{B \models R \overset{K}{\longleftrightarrow} B}$$

20) For V16, applying A3, A6, A10, V14, V16, and applying I3, Message Interpretation Rule.

$$\frac{B \triangleleft \star(H(K_1^i \oplus H_K(N_R))) \wedge B \ni (K_1^i, H_K(N_R)) \wedge B \models T \overset{K_1^i}{\longleftrightarrow} B \wedge B \models \sharp(H(K_1^i \oplus H_K(N_R)))}{B \models T \mid\sim H(K_1^i \oplus H_K(N_R))}$$

21) For V20, applying V16, and applying F1, Freshness Rule. This is the proof for G1, $B \models T \mid\sim \sharp(H(K_1^i \oplus H_K(N_R)))$.

$$\frac{B \models \sharp(H(K_1^i \oplus H_K(N_R))) \wedge B \models T \mid\sim H(K_1^i \oplus H_K(N_R))}{B \models T \mid\sim \sharp(H(K_1^i \oplus H_K(N_R)))}$$

Message 4 $R \triangleleft \star(H(K_2^i), \{R \overset{K_{RB}}{\longleftrightarrow} B\}_{K_{RB}}) \rightsquigarrow B \models R \overset{K_{RB}}{\longleftrightarrow} B$

22) The extension to the message, $B \models R \overset{K_{RB}}{\longleftrightarrow} B$, is valid because it holds when the message is sent as is evident from the initial assumptions, A20.

23) Applying T1, Being-Told Rule.

$R \triangleleft (H(K_2^i), \{R \overset{K_{RB}}{\longleftrightarrow} B\}_{K_{RB}})$

24) Applying T2, Being-Told Rule.

$R \triangleleft H(K_2^i)$

25) Applying P1, Possession Rule.

$R \ni H(K_2^i)$

26) For V25, applying P1, Possession Rule.
$$\frac{R \models \sharp(K_2^i) \wedge R \ni H(K_2^i)}{R \models \sharp(H(K_2^i))}$$

27) For V23, applying P1, Possession Rule.
$$R \ni (H(K_2^i), \{R \overset{K_{RB}}{\longleftrightarrow}\}_{K_{RB}})$$

28) For V23, applying A18, V27, and the freshness $\sharp(H(K_2^i), K_{RB})$ is straightforward, and applying I1, Message Interpretation Rule.
$$\frac{R \triangleleft \star(H(K_2^i), \{R \overset{K_{RB}}{\longleftrightarrow}\}_{K_{RB}}) \wedge R \ni (H(K_2^i), \{R \overset{K_{RB}}{\longleftrightarrow}\}_{K_{RB}}) \wedge R \models R \overset{K_{RB}}{\longleftrightarrow} B \wedge R \models \sharp(H(K_2^i), K_{RB})}{R \models B \mid\sim (H(K_2^i), R \overset{K_{RB}}{\longleftrightarrow} B)}$$

29) Applying I7, Message Interpretation Rule.
$$R \models B \mid\sim R \overset{K_{RB}}{\longleftrightarrow} B$$

30) Applying A20, and applying J1, Jurisdiction Rule. This is the proof for G5, $R \models R \overset{K_{RB}}{\longleftrightarrow} B$.
$$\frac{R \models B \mid\sim R \overset{K_{RB}}{\longleftrightarrow} B \wedge B \models R \overset{K_{RB}}{\longleftrightarrow} B}{R \models R \overset{K_{RB}}{\longleftrightarrow} B}$$

31) We omit the proof for G6 since, for the encrypted message with the key, K_{RB}, there is no further message exchange after this step. That is, the encrypted message of the entity, B, is replied to R and decrypted by R. Thus, the proof is not needed at this moment.

Message 5 $T \triangleleft \star(H(K_2^i)) \rightsquigarrow T \ni K_2^i$

32) The extension to the message, $T \ni K_2^i$, is valid because it holds when the message is sent as is evident from the initial assumptions, A5.

33) Applying T1, Being-Told Rule.
$$T \triangleleft H(K_2^i)$$

34) Applying P1, Possession Rule.
$$T \ni H(K_2^i)$$

35) Applying A7, and applying F10, Freshness Rule.
$$\frac{T \models \sharp(K_2^i) \wedge T \ni H(K_2^i)}{T \models \sharp(H(K_2^i))}$$

36) Applying A5, A9, V33, and applying I3, Message Interpretation Rule.

$$\frac{T \triangleleft \star(H(K_2^i)) \wedge T \ni K_2^i \wedge T \models T \overset{K_2^i}{\longleftrightarrow} B \wedge T \models \sharp(H(K_2^i))}{T \models B \mid\sim H(K_2^i)}$$

37) Applying V35, and applying F1, Freshness Rule. This is the proof for G2, $T \models B \mid\sim \sharp(H(K_2^i))$.

$$\frac{T \models \sharp(H(K_2^i)) \wedge T \models B \mid\sim H(K_2^i)}{T \models B \mid\sim \sharp(H(K_2^i))}$$

As shown above, the proof goals G1-G6 are accomplished by verification steps V12, V19, V21, V30, and V37, respectively.

4.2 Evaluation

4.2.1 Security Analysis

We evaluate our protocol in the view point of the security requirement.

Data Confidentiality

- *On Data Privacy of Tag Bearers*: Our protocol guarantees the secure mutual authentication only with the hashed messages, $S = h_k(r)$, $ID = h(k_1 \oplus S \oplus C)$, and $ID' = h(k_2)$, and $\mathcal{T}$ does store no privacy information of tag bearers. 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 step 2 is eavesdropped by an adversary, it is meaningless. Thus, data confidentiality of tag owners is guaranteed and the user privacy on data is strongly protected.

- *On Application Data*: Based on the assumption that the communication channel between $\mathcal{R}$ and $\mathcal{B}$ is insecure, the application-specified data of $\mathcal{B}$, $DATA$, is easily eavesdropped by adversaries during the authentication step 4. It makes a serious security problem since $DATA$ is in real

world domain. As $\mathcal{R}$ and $\mathcal{B}$ have enough computational power, in our scheme, we adapt symmetric-key cryptosystem to protect $DATA$. The $DATA$ of the corresponding $\mathcal{T}$ is not compromised since it is encrypted by $\mathcal{B}$ and decrypted only by the authorized $\mathcal{R}$ with the randomly generated secret key, $h_k(S)$, from S of $\mathcal{R}$.

Tag Anonymity

From the initial point of authentication processes, we use two fresh random secrets, k_1 and k_2, as the shared secrets between entities, $\mathcal{T}$ and $\mathcal{B}$. These secrets are randomized and anonymous since they are updated for every read attempt. To make tag's authentication information randomized, $\mathcal{T}$ does hash calculation of its initial key, k_1, with random value, S, that is generated and given by $\mathcal{R}$ when querying. When the mutual authentication is successfully completed, $\mathcal{T}$ changes its two secrets for next session. The proposed protocol seamlessly integrates one-wayness of hash function into mutual authentication processes. Thus, tag anonymity is guaranteed and the location privacy of a tag owner is not compromised, either.

Data Integrity

Based on the mutual authentication, our protocol guarantees the data integrity between $\mathcal{T}$ and $\mathcal{B}$. In every session, $\mathcal{B}$ check the validity of the authentication message transmitted from $\mathcal{T}$ after the authentication step 3. $\mathcal{B}$ replies only for valid tag and updates its shared secrets for next session. $\mathcal{T}$ also changes its shared secrets only for the valid authentication message of $\mathcal{B}$. Even though there is possible data loss for the reply message to $\mathcal{T}$, $\mathcal{B}$ guarantees data integrity recovering the data of previous session using a pair of records in D.

In addition, the linkage between the authentication information and $\mathcal{T}$ itself is given for guaranteeing integrity against forgery and data modifica-

tion for $\mathcal{T}$. When $\mathcal{T}$ uses its the chip serial number, C, when it generates its authentication message. $\mathcal{B}$ can check the validity of the authentication information of $\mathcal{T}$, thus, integrity of the linkage is guaranteed.

Availability

- *Man-in-the-middle Attack Prevention*: Through the authentication step 1 to step 3, $\mathcal{R}$ sends S to $\mathcal{T}$, and S and r to $\mathcal{B}$ for preventing the man-in-the-middle attack. $\mathcal{B}$ can verify S with the calculation of the keyed hashed value of r transmitted from $\mathcal{R}$. Only an authorized reader shares the key for the keyed hash function of $\mathcal{R}$ and $\mathcal{B}$. An illegitimate $\mathcal{R}$ cannot generate the valid S and the invalid authentication messages are detected by $\mathcal{B}$. Thus, the man-in-the-middle attack is strongly prevented even though an adversary tries to impersonate as a legitimate reader and get the information from $\mathcal{T}$ and impersonate as the legitimate $\mathcal{T}$ responding to $\mathcal{R}$.

- *Unauthorized Reader Detection*: The proposed protocol is designed based on the insecure communication channel between $\mathcal{R}$ and $\mathcal{B}$. Similar to the way of man-in-the-middle attack prevention, the man-in-the-middle attack by $\mathcal{R}$ as an illegitimate reader is detected and can be prevented by $\mathcal{B}$ on the insecure channel between $\mathcal{R}$ and $\mathcal{B}$.

- *Replay Attack Prevention*: The replay attack for $\mathcal{B}$ is detected and prevented in the authentication step 3 when the attacker retransmits the message of $\mathcal{T}$ to the legitimate $\mathcal{R}$. The freshness of shared secrets between $\mathcal{T}$ and $\mathcal{B}$ is guaranteed for each session. Thus, $\mathcal{B}$ can evaluate the validity of the authentication message of $\mathcal{T}$ by checking the randomized secrets, k_1 in ID. The replay attack to $\mathcal{T}$ also can be enabled to violate the synchronization between $\mathcal{T}$ and $\mathcal{B}$ for each session. Similarly, the replay attack for $\mathcal{T}$ is detected and prohibited through the authentication step 5 for $\mathcal{T}$ by checking ID' transmitted from $\mathcal{B}$.

- *Forgery Resistance*: To give the forgery resistance feature, we simply exclusive-or the embedded chip serial number, C, of $\mathcal{T}$ to the authentication information, ID. C is initially embedded during the chip manufacturing. $\mathcal{B}$ also initially keeps each tag's chip serial number initially and authenticates the ownership of the authentication information for $\mathcal{T}$. Whenever $\mathcal{T}$ generates ID, it refers to C, so we can come up with the linkage between ID and $\mathcal{T}$ itself. Thus, the forgery like simple copy or counterfeiting of $\mathcal{T}$ is prevented.

- *Data Recovery*: By using the pair of records in D and managing AE for each $\mathcal{T}$ as we described in the authentication `step 3`, our protocol provides the data recovery against the data loss such as DoS, message hijacking, power interruption, *etc.* during the authentication processes.

Table 4.4 shows the comparison of the security-related features and the possible attacks for our protocol with the previous results in the next section. We denote the hash-lock scheme [18], the extended hash-lock scheme [18], and the hash-based varying Identifier [7] by HLS, EHLS, and HBVI, respectively.

As shown in Table 4.4, the proposed protocol satisfies the security requirements and provides the privacy protection features. Our protocol exhibits much secure and more reliable compared to other previous results.

4.2.2 Performance Analysis

We analyze the performance of the proposed scheme in forms of the following overheads: 1) computation, 2) storage, 3) communication , and 4) cost.

- **Computational Overhead**: $\mathcal{T}$ requires only a hash calculation and a XOR operation and needs three hash calculation. However, the cost of hash calculation at the server side is $2n$, where n is the number of tags. Compared to [7], the cost of our protocol has overheads for $\mathcal{B}$. However, in [7], the anonymity of tag is guaranteed only after the authentication

Table 4.4: Security Comparison

Protocol	HLS [18]	EHLS [18]	HBVI [7]	Our Scheme
User data confidentiality	×	△	△	◯
Tag anonymity	×	△	△	◯
Data integrity	△	△	◯	◯
Mutual authentication	△	△	△	◯
Reader authentication	×	×	×	◯
Man-in-the-middle attack prevention	△	△	×	◯
Replay attack prevention	△	△	◯	◯
Forgery Resistance	×	×	×	◯
Data Recovery	×	×	◯	◯

†† Notation

◯ satisfied △ partially satisfied × not satisfied

is successfully completed. Therefore, the location privacy of tag bearers is compromised until the next session is successfully started. To make the output of $\mathcal{T}$ anonymous for the current session, $\mathcal{B}$ should check for every records of D to authenticate each tag like EHLS [18]. [14] also shows that the computational cost of $\mathcal{B}$ is also $2n$ at the same condition, which means the cost of our scheme is reasonably feasible. However, note that the reduction of this cost should be needed for the admirable performance.

On the other side, our protocol seems to have encryption and decryption overheads for $\mathcal{R}$ and $\mathcal{B}$. However, those cryptographic tools are needed to secure $DATA$ on the insecure channel. We assume that $\mathcal{R}$ and $\mathcal{B}$ have enough computational power to process encryption and decryption based on the symmetric-key cryptosystem.

- **Storage Overhead**: To compare with the previous protocols, we assume the sizes of all components are L bits, and a RNG and a hash function

are $h, h_k : \{0,1\}^* \rightarrow \{0,1\}^{\frac{1}{2}L}$ and $r \in_U \{0,1\}^L$, respectively. In our protocol, $\mathcal{T}$ only has a hash function and XOR function, and the size of the memory is $2\frac{1}{2}L$. Thus, the proposed protocol is light-weight and practical. We exclude the comparison for the application-specified data, $DATA$ since the size of $DATA$ depends on applied applications.

- **Communication Overhead**: The proposed protocol accomplishes mutual authentication between $\mathcal{T}$ and $\mathcal{B}$ requiring five rounds. As we denote in the previous section, some protocols [18, 14] requires three or six rounds. However, their protocol have synchronization problem on authentication data between $\mathcal{T}$ and $\mathcal{B}$. Five rounds is mostly acceptable for a minimum number of mutual authentication in RFID environment. Therefore, the proposed protocol is feasible in the sense of communication overheads.

- **Cost Overhead**: [17, 16, 39] claimed that generally acceptable low-cost RFID tag should not exceed 5 cents and the IC cost should not exceed 2 cents to achieve a 5 cent tag. They said the limitation of the number of gates is 7.5-15 K-gate and a 100-bit EPC chip requires approximately 5-10 K-gate. Consequently, the number of gates available for security generally cannot exceed 2.5-5 K-gate [14].

  The proposed protocol requires only a hash function and a XOR unit. Currently, [13] introduced the implementation results of three variations, PH (NH-Polynomial), PR (NH-Polynomial with Reduction), WH (Weighted NH-Polynomial with Reduction) of NH (a universal hash function of UMAC). Their results shows a hash function unit for 64-bit output of WH, can be implemented with only about 1.7 K-gate and low-power consumption. In our protocol, only one hash function unit and temporary gates for XOR operation are needed. If we assume the temporary gates for XOR operation needs several tens of gates, the

Table 4.5: Performance Comparison

Protocol	Entities	HLS [18]	EHLS [18]	HBVI [7]	Our Scheme
No. of Hash Operation	$\mathcal{T}$	1	2	3	2
	$\mathcal{B}$	¬	n	3	2n
No. of Keyed Hash Operation	$\mathcal{R}$	¬	¬	¬	1
	$\mathcal{B}$	¬	¬	¬	1
No. of RNG Operation	$\mathcal{T}$	¬	1	¬	¬
	$\mathcal{R}$	¬	¬	¬	1
	$\mathcal{B}$	¬	¬	1	¬
No. of Encryption	$\mathcal{B}$	¬	¬	¬	1
No. of Decryption	$\mathcal{R}$	¬	¬	¬	1
Number of Authentication Steps		6	5	5	5
Required Memory Size	$\mathcal{T}$	$1\frac{1}{2}L$	$1L$	$3L$	$2\frac{1}{2}L$
	$\mathcal{R}$	¬	¬	¬	$1\frac{1}{2}L$
	$\mathcal{B}$	$2\frac{1}{2}L$	$1\frac{1}{2}L$	$9L$	$8L$

†† Notation ¬ not required

n number of tags L size of required memory

number of expected gates is less than 2 K-gate. Therefore, the proposed protocol is feasible and practical for low-cost RFID environment.

Table 4.5 shows the comparison of the computational loads and the required memory size for a single session with previous results [7, 18].

Chapter 5
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.

We proposed a new mutual authentication protocol for the low-cost RFID environment that is computationally light-weight and anonymously interact between entities. The proposed protocol basically fits the low-cost RFID system environment. The tag only has a hash function with the shared two fresh random secrets of small memory size. With this minimal cryptographic primitive, our protocol provides the mutual authentication between the tag and the back-end server and anonymously interacts.

Our protocol is robust enough since it protects the replay attack and man-in-the-middle even when the reader is not a trusted third party and the communication channel is insecure. We add the linkage feature between the tag and its authentication data, so forgery is prohibited. As all authentication messages are randomized and the tag only has its unique identification data, the user data privacy and the location privacy is guaranteed.

Different from the previous works [7, 18], in our protocol, reader authentication and prevention of active attacks are firstly provided based on the assumption that a reader is no more a trusted third party and the communication channel between the reader and the back-end server is insecure. As tags only have hash function and exclusive-or unit, the proposed protocol is

practical for low-cost RFID environment.

Furthermore, the formal proof of the proposed authentication protocol was given based on GNY logic to show correctness for message exchanges throughout the mutual authentication processes.

As a future work, we will consider public key infrastructure (PKI) to check whether a reader is legitimate or not instead of the simple keyed hash-based verification with the symmetric key between a reader and a back-end server. In the aspect of implementation, we need more specified complexity analysis to get the clearer security level. In addition to this, we would like to extend our work to the real world system preventing forgery without the physical access as well as protecting user privacy under the low-cost RFID system environment.

저가의 RFID를 위한 안전한 상호인증 프로토콜

양정규

RFID(Radio Frequency Identification)는 초소형 반도체에 식별 정보를 넣고 무선주파수를 이용하여 이 칩을 지닌 객체를 판독, 추적 및 관리할 수 있는 기술을 말한다. 이는 기존의 바코드를 대체할 차세대 인식 기술로 간주되고 있으며, 향후 도래할 유비쿼터스 사회의 중요한 하부구조를 형성할 것으로 전망되고 있다. 아직까지는 RFID 칩의 높은 가격으로 인하여 RFID 기술의 사용이 보편화되지 못하고 있지만, 칩 가격이 급속히 낮아지고 있어 빠른 시일 내에 RFID 기술의 사용이 전 산업분야로 확대되어 나갈 것으로 전망된다.

그러나 향후 보편적인 RFID 태그의 사용에 있어서 기존의 보안위협과 더불어 프라이버시 침해라는 중요한 문제를 내재하고 있으며, 이는 바람직한 RFID 환경의 구축에 있어 향후 커다란 장애물로 대두될 것으로 예상된다. 이 문제는 RFID 태그의 식별 정보가 무선 주파수라는 인터페이스를 사용함으로써 제 3자에 의해서 쉽게 식별될 수 있다는 RFID 시스템의 기본적인 특성으로 인하여 발생한다. 즉, 태그의 소유자가 알지 못하는 사이에 태그의 정보가 전송됨으로써 개인의 신상과 관련된 정보의 유출 및 개인의 행적에 대한 추적과 같은 프라이버시 침해 요소를 유발 시킨다. 이러한 RFID 시스템의 특성은 태그가 부착된 제품에 대한 위변조 및 서비스거부 공격과 같은 위협에 쉽게 노출되게 된다. 따라서 RFID의 향후 성공적인 산업화를 위해서는 저가의 태그에 대한 가격적, 기술적 측면을 만족시킴과 동시에 위협 요소의 제거 및 프라이버시 문제를 해결해야 하는 것이 우선 과제가 되고 있다. 저가의 RFID 환경에 대하여 적은 계산량을 요구하는 해쉬 및 랜덤넘버 생성기 등의 암호학적 도구에 기반을 둔 효과적인 기법들이 제시되고 있으며, Henrici 등 [7], Okubo 등 [14], Weis 등 [18]이 제안한 프로토콜들

이 대표적이다.

본 논문에서는 RFID 시스템의 특성 및 위협요소를 알아보고 이에 따른 보안요구사항을 만족하는 저가의 RFID 환경에 적합한 상호인증 프로토콜을 제안한다. RFID 시스템에서의 상호인증이란, 태그, 리더 및 백-엔드 서버로 구성된 전형적인 RFID 시스템 상에서 태그의 식별을 위한 유일한 식별자(ID)에 대하여 리더 또는 백-엔드 서버의 인증과 동시에 성공적인 세션의 종료를 위한 태그에서의 리더 또는 백-엔드 서버의 인증을 말한다. 제안 프로토콜은 개체간의 상호인증에 기반을 두고 태그와 백-엔드 서버가 동일한 두개의 랜덤한 비밀값을 공유하며, 이를 통해 해당 세션에서 개체들 간의 상호인증 및 비밀값에 대한 무결성을 보장한다. 태그 소유자의 프라이버시 문제를 해결하기 위하여 태그는 상호인증을 위한 태그의 유일한 식별 정보만 가지게 하고, 해당 세션에서 인증을 위한 메시지 교환 시 리더가 생성한 랜덤 값을 함께 공유하여 매 세션마다 태그의 익명성을 유지함으로써 태그 소유자의 위치정보에 대한 노출을 보호한다. 또한, 제안 프로토콜은 공격자중간공격(Man-in-the-middle Attack), 재생공격(Replay Attack) 및 위조(Forgery) 등의 RFID 시스템에 대표적인 위협에 강하도록 설계되었다.

[7, 14, 18]에서와 달리, 제안 프로토콜에서는 처음으로 리더와 백-엔드 서버 간의 통신 채널이 안전하지 않고, 리더 또한 신뢰할 수 없는 개체라는 가정에 기반을 두어 프로토콜을 설계하였다. 이는 향후 리더가 모바일 기기를 사용한 무선 네트워크 환경에서 동작할 경우에도 태그 소유자의 프라이버시 및 안전한 상호인증을 보장하며, 외부의 공격에 대한 안전성을 제공한다. 이와 더불어 본 논문에서는 제안한 상호인증 프로토콜에 대해서 GNY 로직 [20]에 기반을 두어 그 안전성을 증명하였다.

References

1. 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*, Nov. 2002.

2. A. Joux, Collisions in SHA-0, Presented at the rump session of *Crypto 2004*.

3. A. Juels, Privacy and Authentication in Low-Cost RFID Tags, Available at *http://www.rsasecurity.com/rsalabs/staff/bios/ajuels/*.

4. A. Juels, R.L. Rivest, and M. Szydlo, The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy, *In the Proceedings of Thenth ACM Conference on Computer and Communications Security(CCS 2003)*, Oct. 2003.

5. A. Juels, Minimalist Cryptography for Low-Cost RFID Tags, Available at *http://www.rsasecurity.com/rsalabs/node.asp?id=2033*.

6. A. Juels and R. Pappu, Squealing euros: Privacy protection in RFID-enabled banknotes, In Rebecca N. Wright, editor, *In the Proceedings of Financial Cryptography - FC'03*, LNCS, Vol.2742, pages 103-121, Le Gosier, Gaudeloupe, French West Indies, IFCA, Springer-Verlag, Jan. 2003.

7. D. Henrici, and P. Müller, Hash-based Enhancement of Location Privacy for Radio-Frequency Identification Devices using Varying Identifiers, *In the Proceedings of PerSec'04 at IEEE PerCom*, pp.149-153, Mar. 2004.

8. E. Biham and R. Chen, New results on SHA-0 and SHA-1, Presented at the rump session of *Crypto 2004*.

9. G. Avoine, Privacy issues in RFID banknotes protection schemes, *In the Proceedings of the Sixth Smart Card Research and Advanced Application IFIP Conference - CARDIS*, Kluwer, Toulouse, France, Aug. 2004.

10. ISO/IEC JTC 1/SC 31/WG 4, Information technology AIDC techniques - RFID for item management Air interface, Part3: Parameters for air interface communications at 13.56 MHz, Version N681R, Apr. 2004.

11. I. Vajda and L. buttyan, Lightweight Authentication Protocols for Low-Cost RFID Tags, *In the Proceedings of the Second Workshop on Security in Ubiquitous Computing , in conjunction with Ubicomp 2003*, Seattle, Washington, USA, Oct. 12, 2003.

12. K. Finkenzeller, RFID Handbook Second Edition, *Wiley & Sons*, 2002.

13. K. Yüksel, Universal Hashing for Ultra-Low-Power Cryptographic Hardware Applications, *Master's Thesis, Dept. of Electronical Engineering, WPI*, 2004.

14. M. Ohkubo, K. Suzuki, and S. Kinoshita, Cryptographic Approach to Privacy-Friendly Tags, *RFID Privacy Workshop 2003*, MIT, MA, USA, Nov. 2003.

15. S. Sarma, S. Weis, and D. Engels, RFID Systems and Security and Privacy Implication, *Technical Report MIT-AUTOID-WH-014*, AutoID Center, MIT, 2002.

16. S. Sarma, S. Weis, and D. Engels, Radio-Frequency Identification: Security Risks and Challenges, *CryptoBytes, 6(1), 2003.*

17. S. Weis, Security and Privacy in Radio-Frequency Identification Devices, *Master's thesis, MIT*, 2003.

18. S. Weis, S. Sarma, R. Rivest, and D. Engels, Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems, *In the Proceedings of the First Security in Pervasive Computing*, LNCS, Vol.2802, pp.201-212, 2004.

19. X. Wang, X. Lai, D. Feng, and H. Yu, Collisions for hash functions MD4, MD5, HAVAL-128 and RIPEMD, Presented at the rump session of *Crypto 2004*.

20. L. Gong, R. Needham and R. Yahalom, Reasoning about Belief in Cryptographic Protocols, *1990 IEEE Computer Scociety Synopsis on Research in Security and Privacy*, pp.234-248, 1990.

21. M. Burrow, M. Abadi, and R. Needham, A Logic of Authentication, *In the Proceedings of the Twelfth ACM symposium on Operating systems principles*, pp.1-13, ISBN:0-89791-338-8, 1989.

22. Marks & Spencer Develops Intelligent Clothing , Available at *http://www2.marksandspencer.com /thecompany/mediacentre*.

23. RFID Journal. Michelin embeds RFID tags in tires, Available at *http://www.rfidjournal.com/article/view/269*.

24. Texas Instruments ISO 14443 payment platform promises faster data transfer rates and more security, Available at *http://http://www.rfidjournal.com/article/articleview/327/1/1/*.

25. S. Kinoshita, F. Hoshino, T. Komuko, A. Fujimura, and M. Okubo, Nonidentifiable Anonymous-ID Scheme for RFID Privacy Protection, *In the Proceedings of CSS 2003*, pp.497-502,

IPSJ, 2003 Oct. In Japanese. See English version available at *www.autoidlabs.com/whitepapers/KEI-AUTOID-WH004.pdf*.

26. D.Wagner, A Generalized Birthday Problem, *In the Proceedings of Advances in Cryptology - Crypto'02*, LNCS, Vol.2442, pp.288-303, Springer-Verlag, 2002.

27. G.Yuval, How to swindle Rabin, *Cryptologia*, 3:187-189, pp.187-190, Jul. 1979.

28. I.B.Damgård, Collision Free Hash Functions and Public Key Signature Schemes, *In the Proceedings of Advances in Cryptology-Eurocrypt '87*, LNCS, Vol.304, pp.203-216, Springer-Verlag, 1988.

29. L.Lamport, Password Authentication with Insecure Communications, *Communications of ACM*, Vol.24, No.11, pp.770-772, Nov. 1981.

30. R.Rivest and A.Shamir, PayWord and MicroMint: Two Simple Micropayment Schemes, *In the Proceedings of International Workshop on Security Protocols*, LNCS, Vol.1189, pp.69-87, Springer-Verlag, 1996.

31. K.Suzuki, K.Kobayashi and H.Morita, Efficient Sealed-bid Auction using Hash Chain, *In the Proceedings of ICISC'00*, LNCS, Vol.2015, pp.183-191, Springer-Verlag, 2000.

32. Y. Hwang, S. Lee, D. Lee, and J. Lim, An Authentication Protocol for Low-Cost RFID in Ubiquitous, *In the Proceedings of CISC S'04*, pp.109-114, in Korean, Jun. 2004..

33. S. Shepard, RFID Radio Frequency Identification, *MacGraw-Hill*, 2005.

34. T. Scharfeld, An Analysis of the Fundamental Constraints on Low Cost Passive Radio-Frequency Identification System Design, *Master's Thesis, Dept. of Mechanical Engineering, MIT*, Cambridge, 2001.

35. N. Good, D. Molnar, J. M. Urban, D. Mulligan, E. Miles, L. Quilter, and D. Wagner, Radio Frequency Id and Privacy with Information Goods, *In the Proceedings of the 2004 ACM worksop on Privacy in the electronic society*, Oct. 2004.

36. S. Bakhtiari, R. Safavi-Naini, and J. Pieprzyk, Cryptographic Hash Function: A Survey, *Technical Report 95-09, Department of Computer Science*, University of Wollongong, Jul. 1995.

37. D. Molnar, D. Wagner, Privacy and security in library RFID: issues, practices, and architectures, *In the Proceedings of the 11th ACM conference on Computer and communications security*, pp.210-219, Washington DC, USA, 2004.

38. T. Ishikawa, Y. Yumoto, M. Kurata, M. Endo, S. Kinoshita, F. Hoshino, S. Yagi, M. Nomachi, Applying Auto-ID to the Japanese Publication Business: To Deliver Advanced Supply Chain Management, Innovative Retail Applications, and Convenient and Safe Reader Services, *White Paper, Auto-ID Center*, 242-8520, Japan, Oct. 2003, Available at *http://whitepapers.silicon.com/0,39024759,60090119p-39000512q,00.htm.*

39. S. E. Sarma, Towards the Five-cent Tag, *Technical Report, MIT-AUTOID-WH-006, MIT Auto-ID Center*, 2001, Available at *http://www.autoidcenter.org.*

Acknowledgements

First and foremost I would like to thank my academic advisor Prof. Kwangjo Kim for his constant direction and support. Without his guidance, I could never have carried out my research in ICU. Special thanks are also due to Prof. Jae Choon Cha and Dr. Jong Suk Chae for their generosity and agreeing to serve as advisory committee members. I also express my gratitude to Prof. SeongHyun Myaeng for his inspiration and kind advice. I am grateful to Prof. Adi Shamir of the Weizmann Institute in Israel for his precious comments and kind advice on our protocol during his visit to ICU.

Particularly, I would like to thank staffs in my company, Korea Minting and Security Printing Corporation (KOMSCO), for giving me a invaluable chance to do research: special gratitude to HaeYoun Jung, JuTae Um, HwangBong Kim, JongSeong Kim, and many others. I am specially thankful to all coworkers of Management Information Team.

I would also like to thank all members of Cryptology and Information Security Laboratory: Hyunrok Lee, Zeen Kim, Seokkyu Kang, SungChul Heo, Sangshin Lee, JaeMin Park, Vo Duc Lim, and Dang Nguyen Duc from Vietnam, and Ping Wang from China, and Divyan from India, for giving me lots of interests and good advices during the course of my study. I also thank Sunhye Mok for helpful support as a staff member. I also appreciate to the graduates: Byunggon Kim, Songwon Lee, HwaSun Chang, Chuljoon Choi, SungJoon Min, SangWon Lee, JaeHyrk Park, JoongMan Kim, SuGil Choi, Kyusuk Han, Jungyeon Lee, and Yan Xie, Xiaofeng Chen from China for their everlasting guidance in life and study of ICU. I also give my special gratitude for his aid on every aspect to Kui Ren who is in Worcester Polytechnic Institute (WPI). We all have shared unforgettable time and enjoy ICU life together.

In addition, I would like to give my thanks to SaKwang Song, and SeongSu Kim of IRNLP lab., Jihyun Lee of GTC lab., and Chan-Kook Park of V&S lab. for their deep interest and affection.

My biggest gratitude goes to my parents, and to my parents-in-law for their endless concerns and devotional affection. I cannot forget their trust and encouragement on me. Also many thanks to my brother, sisters and brother-in-laws for their generosity and warm-hearted help.

My love and thanks go to my wife Eunha Hwang for her endless encouragement and devotion, and to my lovely daughter SeoHyun. They were medicine whenever I had hard time, are strong belief for my way of life, and will be permanent happy in my entire life. I dedicate this work to them.

Curriculum Vitae

Name : Jeongkyu Yang

Date of Birth : Mar. 1. 1973

Sex : Male

Nationality : Korean

Education

2003.3–2005.2 Cryptology and Information Security, Engineering
Information and Communications University (M.S.)

1991.1–1999.2 Computer Engineering
Hannam University (B.A.)

Career

2004.6–2004.12 Graduate Research Assistant
Research on RFID Privacy and Security Protection
Electronics and Telecommunications Research Institute (ETRI)

2004.5–2004.12 Graduate Research Assistant
Research on the Security for Special Digital Signature
Security Engineering Research Center (SERC)

2004.1–2004.12 Graduate Research Assistant
Research on Link Layer Security Technology
Electronics and Telecommunications Research Institute (ETRI)

2003.7–2004.2 Graduate Research Assistant
Ubiquitous System Security Technology: Protecting Digital Contents from Illegal Use
NITZ Co.

2003.12–2004.4 Graduate Research Assistant
Support for Running the International Research Center for Information Security
The Ministry of Information and Communications (MIC)

1999.2– Member of Engineering Staff
Management Information Team
Korea Minting & Security Printing Corporation (KOMSCO)

Publications

(1) 2005.1 Jeongkyu Yang, Kui Ren, and Kwangjo Kim, "Security and Privacy on Authentication Protocol for Low-cost RFID", To be appeared in *In the Proceedings of SCIS'05*, Maiko Kobe, Japan, January 25-28, 2004.

(2) 2004.12 Jeongkyu Yang, Kui Ren, SuGil Choi, and Kwangjo Kim, "Mutual Authentication: A Privacy Preserving Method for Low-cost RFID", *In the Proceedings of CISC-W'04*, Chung-Ang University, Seoul, Korea, December 4, 2004.

(3) 2004.10 Jeongkyu Yang and Kwangjo Kim, "Privacy Preserving Mutual Authentication Protocol for Low-cost RFID", *In the Proceedings of KIISC Conference Region Chungcheong 2004*, pp.55-68, Gongju University, Gongju, Korea, October 2, 2004.

(4) 2004.6 Jeongkyu Yang, Kwangjo Kim, and Cheol Sig Pyo, "A Study on Low-cost RFID Schemes", *In the Proceedings of CISC-S'04*, Vol.14, No.1, pp.605-609, Kyung Dong Univ., Korea, January 24-26, 2004.

(5) 2003.10 Songwon Lee, Jeongkyu Yang, and Kwangjo Kim, "Threshold Password-Based Authentication Using Bilinear Pairings", *In the Proceedings of CSS2003*, pp.385-390, Kitakyushu, Japan, October 29-31, 2003.