A Thesis for the Degree of Master of Science

A Study on Scalable and Untraceable Authentication Protocol of RFID tags

Youngjoon Seo

School of Engineering

Information and Communications University

2007

A Study on Scalable and Untraceable Authentication Protocol of RFID tags

A Study on Scalable and Untraceable Authentication Protocol of RFID tags

Advisor : Professor Kwangjo Kim

by

Youngjoon Seo
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
12. 26. 2006
Approved by

(signed)

Professor Kwangjo Kim
Major Advisor

A Study on Scalable and Untraceable Authentication Protocol of RFID tags

Youngjoon Seo

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

12. 26. 2006

Approved:

Chairman of the Committee
Kwangjo Kim, Professor
School of Engineering

Committee Member
Jae-Choon Cha, Assistant Professor
School of Engineering

Committee Member
Byoungcheon Lee, Assistant Professor
Dept. of Information Security, Joongbu University

M.S. Youngjoon Seo
20052049

A Study on Scalable and Untraceable Authentication Protocol of RFID tags

School of Engineering, 2007, 46p.
Major Advisor : Prof. Kwangjo Kim.
Text in English

Abstract

RFID (Radio Frequency IDentification) is recently becoming popular, and plays definitely an important role in moving to ubiquitous society due to deploying its convenience and economic efficiency. Furthermore, RFID nowadays comes into the spotlight as a technology to substitute the barcode system since RFID can solve several problems in the barcode system: (1) to require line of sight for scanning, (2) no read/write capability including limited capacity for encoding information, (3) opportunities of human error, and more problems in [48, 45]. RFID is expected to achieve unlimited economic gain. For example, in case that one billion of tagged items are sold by consumer product manufacturers, then a difference of one cent between tags and the barcode can give ten million dollar economic gain since tags can be attached to any kinds of items [42].

RFID technology, on the other hand, is jeopardized from various attacks and problems preventing widespread RFID deployment: replay attack, spoofing, traceability, desynchronization, scalability, and tag cloning. We focus ourselves on untraceabily and scalability in this thesis. We give the reason why untraceability and scalability is important to deploy RFID widely. To prevent an adversary from tracing tagged item is most important in RFID

system since it infringes personal privacy. For example, in [49], Albrecht who organized a Benetton boycott called RFID tags "spy chips" due to the traceability of tags. And moreover, tags with unique ID can be associated with a personal identity. Garfinkel *et al.* [12] dealt with personal privacy threats as follows: action, association, location, preference, constellation, transaction, and breadcrumb threat.

There must be a trade-off between scalability and untraceability. However, we try to keep the constant computational time in back-end server regardless of the number of tags when designing an untraceable protocol. If a response from T, as an example, does not include information about ID of T, which is dynamic or incomputable, these protocols are likely to be unscalable since readers are supposed to exhaustively search in database to find ID of T. If a response from T, on the contrary, includes information about ID of T, which is static or computable, tagged items are likely to be traceable because an adversary also can find ID of T as an authorized one does.

The previous protocols [14, 44, 24, 8, 39] and hash lock scheme [41] are scalable, but traceable. Rhee *et al.* [30], Ohkubo *et al.* [28] and randomized hash lock [41] schemes are untraceable, but unscalable. Therefore, we try to design a scalable and untraceable protocol that any other literatures have not dealt with before.

In this thesis, we propose two RFID authentication protocols that guarantees untraceability and scalability together.

Our protocol without proxies (1) supports ownership transfer[1], (2) considers multi-tag-reader environment, (3) receives messages from the tags what a reader wants. In addition, we address the reason why the item privacy is so important, and a way to keep it securely.

Under the strong assumption that all the channels are insecure, our proto-

[1]Ownership transfer should be supported since the owner of RFID tags could never be unchanged through whole life cycle of RFID tags, but only Molnar *et al.* [27] dealt with ownership transfer to the best of our knowledge.

col using a proxy for individual and the universal re-encryption has several advantages: (1) ownership transfer, (2) untraceability against the compromised tags, and (3) data access authorization level-based service by the back-end server.

Contents

List of Tables

List of Figures

List of Abbreviations

RFID Radio Frequency Identification

DoS Denial-of-Service

ECC Elliptic Curve Cryptosystem

PUF Physical Unclonable Function

UR Universal Re-encryption

EPC Electronic Product Code

PKI Public Key Infrastructure

PRNG Pseudorandom Number Generator

XOR Exclusive-or

List of Notations

R RFID tag reader, or transceiver.

T RFID tag, or transponder.

$\mathcal{T}_k$ Set of T which has same secret key k.

$\mathcal{T}'_k$ Set of T which has secret key k' where $k' \neq k$.

S Back-end Server.

P Proxy.

$\mathcal{A}$ Adversary.

$h()$ One-way hash function.

ID Identifier.

ID_i Pseudo-EPC of T at i-th query(i=0,1,$\cdots$).

k Shared secret key between R and $\mathcal{T}_k$.

TS Timestamp.

TS_{last} last TS sent by an authorized R.

t_i Temporal storage (i=0,1,2).

C Ciphertext.

C' Re-encrypted ciphertext.

SK Private key.

PK Public key corresponding to SK.

SK_M Private key of M.

PK_M Public key of M corresponding to SK_M.

$Cert_M$ Certificate of M.

Sig_M Signature of M.

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

M_1, M_2 Concatenation of messages M_1 and M_2.

PIN Access PIN written into a reserved memory of T.

$l \leftarrow r$ Operator which updates l with r.

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

$\stackrel{?}{>}$ Comparison operator to check whether the left hand side is greater than the right hand side or not.

n Number of tags.

γ Number of tags within an operating range.

β Number of tags that have same k within an operating range.

Chapter 1
Introduction

1.1 Radio Frequency Identification

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

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

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

1.2 RFID Research Issues

Since its invention in the 1940s, RFID has been an obvious target for abuse [33]. The first paper [36] related to RFID security is written by Sanjay Sarma, Stephen Weis, and Daniel Engels in 2002, which includes overview and security threats of RFID technology. After the paper [36], there have been many papers which are hash-based [14, 7, 38, 2, 30, 24, 28, 41, 8, 11, 46], pseudonym-based [18, 13, 26], zero knowledge-based [39] using PUF(Physical Unclonable Function), tree-based protocol[27] using pseudonym generator, off-tag-based schemes [20, 21, 17, 6, 32, 31, 22], and yoking proof [19, 35, 29]that attempt to address the security concerns raised by the use of RFID tags, but it is believed that there is no protocol that provides all of security requirements (See Section 2.1) with low cost as reasonable as applicable until now.

Reference site has been maintained by Gildas Avoine is in [1] and survey paper related to RFID written by Ari Juels is in [15]. The master's thesis of Steven Weis [42] describes early work in the area of RFID privacy, and provides good technical background.

1.3 Motivation

When designing an untraceable protocol completely, we are faced with scalability problem; that is, it increases computational complexity in back-end server. In other words, there is a trade-off between scalability and untraceability. If the responses from T, as an example, do not include information about ID of T, which is dynamic or incomputable, these protocols are likely to be unscalable since readers are supposed to exhaustively search in database to find ID of T. If the response from T, as an opposite example, includes information about ID of T, which is static or computable, tagged items are likely to be traceable because attackers also can find ID of T by computing information about ID of T. The previous protocols [14, 44, 24, 8, 39] and

hash lock scheme [41] are scalable, but traceable. Rhee *et al.* [30], Ohkubo *et al.* [28] and randomized hash lock scheme [41] are untraceable, but unscalable. Accordingly, we try to design two scalable and untraceable protocols that any other literatures have never tried before.

1.4 Our Contributions

In this section, we show our contributions.

We propose the reason why we have to write pseudo-EPC into the memory of T, not a code itself. Writing EPC itself into the memory of T causes infringing item privacy after an adversary eavesdrops EPC or tampering T. Shortly after an adversary finds out what EPC is in the particular tag, he/she can learn the types of items and whether tagged items are expensive or cheap. In other words, item privacy can be violated. It is clear that item privacy causes user privacy and incentive to steal valuable items. On the other hand, writing pseudo-EPC into the memory of T guarantees item privacy even after an adversary comprises tags. It couldn't matter as long as back-end server converts pseudo-EPC into a valid EPC and points to a right entry for retrieving relevant product information.

In the protocol without proxies, our contribution is to design a scalable and untraceable protocol which is more secure than Dimitriou [8] (TD05); we use only four hash operations while TD05 uses five and more hash operations. We make it using a shared secret k; when a reader sends a query, a shared secret k needs to be authenticated by T. This is totally different approach in comparison with the previous literatures. The only tags stored with the same secret k respond to the query from R; a reader gets the message from particular tag what the reader wants. It reduces computational time in tags and back-end sever, especially in multi-tag-reader environment.

In the protocol with a proxy, our proxy supports granular data access and maintains Server Location field which makes readers connect directly the

appropriate back-end server. In other previous protocols, the back-end server has to do some extra works to find the proper server which has the server location for tags. In other words, we alleviate the work in back-end servers.

In our two protocols, we deal with ownership transfer which is one of advanced security requirements. Ownership transfer should be supported since the owner of RFID tags should be changeable through whole life cycle of RFID tags, Molnar *et al.* [27] dealt with ownership transfer to the best of our knowledge.

1.5 Organization

The rest of the thesis is organized as follows. In Chapter 2, we introduce universal re-encryption, security requirements, the previous work and case study which analyze the security of other work contributions. In Chapter 3, we propose our RFID authentication protocol without proxies based on hash function which is a scalable and untraceable protocol enabling ownership transfer. In Chapter 4, we propose our scalable and untraceable protocol enabling ownership transfer of RFID tags. We finally conclude our results in Chapter 5.

Chapter 2
Preliminaries

2.1 Research Goals in RFID systems

Several threats in RFID systems are obstacles to make RFID more popular, familiar, and widespread then before. In this section, we look into design goals (security and other) of RFID protocol. Security goals can be broadly classified into five categories: protection against tracing, eavesdropping, spoofing, and DoS; additionally, ownership transfer should be supported since RFID tags' owner never could be unchange through whole life cycle of RFID tags. The remaining goals can be broadly divided into performance and cost goals.

The followings are the security goals in RFID systems.

Security Goal 1: ***Protection against tracing.*** This is the most important goal. The adversary can collect the responses from all the tags using device like the ghost and leech [23] with the aim of tracing or spoofing which is used for the man-in-the-middle attack; a ghost communicates with a reader and leech; a leech communicates with a tag and ghost. Unwanted tracking may cause a social problem since unwanted tracking leaves room for producing more burglars, rapists, and stalkers than now; so, those who can not protect themselves are likely to reluctant to carry RFID tags unless tracing problem is solved. Hash lock scheme [41] proposed by Weis *et al.* emits same responses all the time, which means *Hash Lock Scheme* is traceable. Relabeling (or refreshing) the identifier of a tag or secret value between tags and readers was suggested not to

be traced. However, relabeling require synchronization between a tag and a reader.

A higher criterion from the level of security compared to the than protection against tracing is indistinguishability which is the same definition with the semantic security; Juels *et al.* [16] presented some of indistinguishability under chosen-plaintext attack (IND-CPA) and chosen-ciphertext attack (IND-CCA) cryptosystem security experiments.

Security Goal 2: ***Protection against eavesdropping***. Any compliant reader can read the response from T without any knowledge of the tag owner. So, the response from T should be encrypted to be recognizable only to authorized ones and not to give any information gain to adversaries. Current EPCglobal Class-1 Generation-2 UHF tag standard [47] sends access and kill password in unencrypted form. RFID tags with this standard are hard to be carried by individual which is vulnerable to even by eavesdropping.

Security Goal 3: ***Protection against spoofing***. An adversary can spoof tags (or readers) to readers (or tags). Spoofing readers is the same meaning with cloning the tag, which gives adversaries a chance to substitute genuine with counterfeit; Bono *et al.* [5] recently have succeeded in breaking the car immobilization system through tag cloning. Auto-ID Labs currently have focused on anti-counterfeit flagship project from every stakeholder in supply chain to customs.

An adversary who spoofs a tag can get information about the tag since the adversary can be regarded as authorized parties; so, spoofing tags is related to *privacy* which is defined that no unwanted information about the tag has to be leaked from the system.

Security Goal 4: ***Protection against DoS***. RFID tags are inherently subject to DoS attack since the endless queries can be made to the nearest

RFID tags to make it inoperable. In addition, DoS occurs such as when synchronization between tags and readers is failed and battery is totally consumed; in that case, external device like proxy or active tags are used. DoS can be very serious when RFID tags are used for medical purpose.

The goal of the blocker [20, 17] tag, noisy tag [6], and selective RFID jamming is privacy, however these can be used for DoS by the malicious person.

Security Goal 5: ***Ownership transfer.*** Molnar *et al.* [27] dealt with sophisticated ownership transfer at the first time. The owner of RFID tags can not be unchangeable. For example, ownership transfer happens from manufacturers to wholesalers, wholesalers to repackagers, repackagers to retailers, and from retailers to consumers in the supply chain; furthermore, consumers can sell or give RFID tags to the other consumers. In case that the protocol of RFID tags does not support ownership transfer, the previous owners can be regarded as the current owner.

The following items are performance and cost goals in RFID systems.

Performance Goal: ***Scalability.*** Scalability means that how many RFID tags can be accommodated in the back-end server. To accommodate a large number of RFID tags, computational time of identification should be reasonable, which means $O(n)$ or $O(mn)$ computational time in the back-end server is not scalable where n is the number of tags in the system and m is the number of read operations. We define *completely scalable* that the back-end server can find identifer of a tag with constant computational time regardless of the number of tags. Protection against tracing and scalability has a trade-off relationship. For example, previous protocols [14, 44, 24, 8, 39] and hash lock scheme [41]

are scalable, but traceable. Rhee *et al.* [30], Ohkubo *et al.* [28] and randomized hash lock [41] schemes are untraceable, but unscalable.

Economic Goal: ***Cost.*** The cost of RFID tags should be as cheap as possible since RFID tags can be attached to all kinds of items even a personal pen or toothbrush which is very low-cost. RFID tags may have 1000-10000 gate equivalents and the number of gates available for security features is 200-2000 [43]. Asymmetric cryptosystems like RSA, ECC, or NTRU and symmetirc cryptosystems like DES and AES are not applicable in terms of gate count so far. For example, even standard cryptographic hash functions like SHA-1 require roughly 20,000 gates [42]; more seriously, SHA-1 was broken by Wang *et al.* [40] in 2005. Lehtonen *et al.* [25] provides good background on the cost of RFID tags.

2.2 Previous Work

There have been many papers which are hash-based [14, 2, 38, 30, 24, 28, 41, 8, 46], pseudonym-based [18, 13], zero knowledge-based [39] using PUF (Physical Unclonable Function), and tree-based protocol [27] using pseudonym generator that attempt to address the security concerns raised by the use of RFID tag, but it is believed that there is no perfect protocol that avoids all of the threats with low cost as reasonable as applicable until now.

2.2.1 Hash Lock Scheme and Randomized Hash Lock Scheme

Hash Lock Scheme [41] (HLS) is based on one-way hash function. It requires implementing a hash on the tag to achieve low-cost tag and scalability. On the other hands, tracking is possible through the shared secret from T which

is the first response from T. And moreover, it is possible to spoof reader and tag. For example, after an adversary eavesdrops the response from T, $metaID = hash(key)$, he/she can spoof the tag; an adversary can spoof the tag since readers send key with unecrypted form. *Randomized Hash Lock scheme* [41] (RHLS) is an extended version of HLS03 to remove traceability, but RHLS03 causes scalability and spoofing problems.

2.2.2 Hash-based Scalable Protocol

Henrici *et al.* [14] (HEN04), Lee *et al.* [24] (LACP05), and TD05 are scalable, but traceable during a valid session. For example, the response from T always contains hash value of ID which does not need brute force search in the back-end server; but, it brings about traceability problem during a valid session. HEN04 first proposed a way to recover message failure and message error using two rows a tag in database. LACP05 and Rhee *et al.* protocol [30] (RKKW05) adopt this way. On the other hands, HEN04 can be traceable by side channel attack due to careless use of counter values. In addition, the other problems in HEN04 was demonstrated in [3].

TD05 solves desynchronization problem entirely through using timestamp. The main idea of TD05 is using nonce, which is implemented with timestamp, and two keyed hash functions(MAC) to prevent cloning attacks [4]. In contrast, MAC is needed to two hash opeations to compute. The main idea of LACP05 is twofold: One is having two rows in the database like HEN04 to recover message failure or message error; the other is using a half of hash when sending a message to reduce communications cost, but it reduces security. Furthermore, spoofing the tag is possible in LACP05. The procedure of spoofing works as follows: first, an attacker sends a query with r, which can be empty, to the tag and then receives $h(ID)$, $h_L(ID, r)$ that is left half of $h(ID, r)$. Lastly, $h(ID) - h_L(ID, r)$, where "$-$" denotes string subtraction, is to let attacker know the $h_R(ID, r)$ which is the third message of LACP05

is the right half of $h(ID, r)$.

2.2.3 Hash-based Untraceable Protocol

Ohkubo *et al.* protocol [28] (OSK03) have strength, untraceability, but also weakness, scalability. The main idea of OSK03 is using two hash functions. One is used to refresh the secret in the tag. The other one is used to respond to a query. OSK03 guarantees untraceability and forward secrecy. However, OSK03 requires exhaustive search in the back-end server. The time complexity of this protocol is $2mn$ in terms of hash computations [2]. The time complexity $O(2mn)$ of OSK03 is less scalable in comparison with RHLS03 and the protocol in [30], $O(n)$.

2.2.4 Other Protocols

Wong *et al.* [44] and Tuyls *et al.* protocol [39] can be traceable since the response from T is fixed all the time; and also, pseudonym-based protocols [18, 13] can be traceable after an adversary collects all of the pseudonym. To prevent tracing, some work in [18, 44, 13] addresses that the right product owner can alter the scattering way or all of the pseudonym after a certain period. Deciding a certain period, however, is also another problem. Tree-based protocol [27] proposed by Molnar *et al.* supports scalability, ownership transfer, delegation and off-line operation. In contrast, it needs lots of computation time, communications cost, and memory storages; in addition, the other tags also can be compromised under a compromise by the other tags.

Yang *et al.* protocol [46] is unscalable and traceable during a valid session. This protocol needs ln computation to find a tag in back-end server where l denotes the number of readers. This protocol belongs to the worst case from the point of scalability to our understanding. For a traceable case, an adversary can send first message to T like an authorized reader, and then the response from T has same value during a valid session. More seriously, If

one reader is compromised, the adversary can succeed mutual authentication with all tags since there is no mechanism to authenticate reader in the side of T.

2.3 How to achieve security requirements?

When designing a RFID authentication protocol, the following properties should be considered to meet: forward secrecy, untraceablility, scalability, synchronization, cloning, preventing spoofing and item privacy. Each subsection describes how to design a protocol which satisfies each property using the five well-known protocols: HLS03, RHLS03, OSK03, TD05, and LACP05.

2.3.1 Forward Secrecy

OSK03 and TD05 are known to meet forward secrecy. OSK03 and TD05 utilize a hash function to update an identifier while HLS03 and RHLS03 did not refresh an identifier; that is, upon compromising an identifier, the adversary learns all the previous transactions in HLS03 and RHLS03. LACP05 uses XOR operation to update an identifier; consequently, LACP05 fails to guarantee forward secrecy. Hash function has a one-wayness property, while XOR operation does not. Pseudonym is used in [18, 13], but the adversary can collect all pseudonyms from the response of T in which case protocol based on pseudonym can not guarantee forward secrecy; more seriously, it can not guarantee untraceablility. In order to design a protocol that guarantees forward secrecy, a protocol designer has to use a hash function when updating secret key as long as there is no alternative. When updating ID_i, finding a lightweight function or scheme that guarantees forward secrecy is also a big open research problem.

2.3.2 Untraceability and Scalability

In Table 3.1, forward secrecy (FS), untraceability (UNT), and untraceability during a valid session (UNT-DVS) are classified as one category. FS and UNT-DVS are classified into UNT; Guaranteeing UNT means satisfying FS and UNT-DVS. OSK03 is successful in designing a UNT completely, but it causes the worst result in terms of scalability. The number of tags is going to increase sharply in the nearest future; furthermore, tag recognition rate is not perfect so far. It increases read operation times; that is, the complexity of OSK03 $O(4mn^2)$ definitely suffers from too much in multi-tag-reader environments since all of the tags within the operating range of reader are supposed to respond a query. That's why scalability also can not be overlooked. We introduce γ as the number of tags within a operating range since all tags, which are stored in back-end server, are not likely to be within a range of reader. After applying γ to complexity of OSK03, it becomes $O(4mn\gamma)$. In this thesis, we define scalability as that the computational complexity is quite suitable for multi-tag-reader environment in the back-end server.

2.3.3 Synchronization

HLS03 and RHLS03 don't need to synchronize the shared key because the shared secret is fixed, but TD05, OSK03 and LACP05 have to synchronize the secret information since they update a key only with authorized readers. OSK03 can lose synchronization due to resilience. If desynchronization occurs, database can not recognize the tag; tag is useless in this case.

2.3.4 Spoofing and Cloning

HLS03 and RHLS03 send message in unencrypted form; so, the adversary can learn the shared secret key by eavesdropping, and then can spoof the

reader and tag. OSK03 does not describe how to play a role in the side of R. In LACP05, reader and tag send message carelessly (See section 2.2.3.); and so, the adversary can spoof the tag. To prevent attacker from spoofing the tag and reader, we should not send message in unencrypted form and update the shared secret key, the same as part of message which is sent. Cloning is divided into two groups: cloning by eavesdropping and cloning by tampering. Cloning by eavesdropping has the same significance with spoofing the reader in terms of security; preventing the adversary from cloning by tampering is hard to prevent since it means the adversary learns all information of storage. However, Tuyls *et al.* protocol [39] shows how to prevent the adversary from cloning by tampering using PUF (Physical Unclonable Function), but it's too costful.

2.3.5 Item Privacy

Guaranteeing item privacy means that the adversary can not find out the contents and price of tagged items even though EPC is revealed by means of any attacks. Violation of item privacy gives an adversary the seduction to steal tagged items after he/she eavesdrops EPC. Although the adversary knows what kind of product is it after tampering tag, item privacy should be guaranteed. For example, there is imitation like tiny jewelry such that the general public can not distinguish genuine from a imitation. The adversary cannot decide to counterfeit or not since there is no way to distinguish them.

2.4 Universal Re-encryption

Universal re-encryption (UR) was proposed by Golle *et al.* [13]. C' is said to represent re-encryption of C provided that two ciphertexts are decrypted to the same plaintext.

UR has several security properties. First, re-encryption is permissible

without knowledge of PK. Second, the one time decryption is sufficient to get the plaintext no matter how many times the re-encryption has done. In most previous literatures, secret keys should be refreshed not to be traced on a regular interval. These secret keys have to be synchronized among parties who share the secret keys after updating while C based on UR doesn't need to synchronize because of the second property. Third, UR based on ElGamal encryption algorithm is semantically secure under the re-encryption if the adversary can not determine b for given two re-encryptions (C'_b, C'_{1-b}) with probability significantly greater than 1/2 (See more details in [13]).

2.4.1 Description

UR consists of four functional components like following description.

- **Key Generation** : TA (Trusted Authority) takes charge of generating keys. TA generates private key SK x and public key PK $y = g^x$, where $x \in_U Z_q$; $\in_U$ denotes uniform and random selection, and q denotes the order of $\mathcal{G}$ which denotes underlying group for the ElGamal cryptosystem.

- **Encryption** : R selects random encryption factor $r = (k_0, k_1) \in Z_q^2$, and then generates $C = [(\alpha_0,\beta_0); (\alpha_1,\beta_1)] = [(my^{k_0},g^{k_0}); (y^{k_1},g^{k_1})]$, where m denotes a plaintext.

- **Decryption** : R forwards C to S. S decrypts C to get the m. First, S checks $\alpha_0,\beta_0,\alpha_1,\beta_1 \in \mathcal{G}$. If this check fails, then decryption fails. Second, S computes $m_0 = (\alpha_0/\beta_0^x)$ and $m_1 = (\alpha_1/\beta_1^x)$ using the private key x of T. If $m_1 = 1$, then decrypted message is $m = m_0$. Otherwise, decryption fails.

- **Re-encryption** : Anyone can re-encrypt C to $C' = [(\alpha'_0,\beta'_0); (\alpha'_1,\beta'_1)] = [(\alpha_0\alpha_1^{k'_0},\beta_0\beta_1^{k'_0}); (\alpha_1^{k'_1},\beta_1^{k'_1})]$ with random re-encryption factor $r' = (k'_0, k'_1) \in Z_q^2$, where $k'_0,k'_1 \in_U Z_q$.

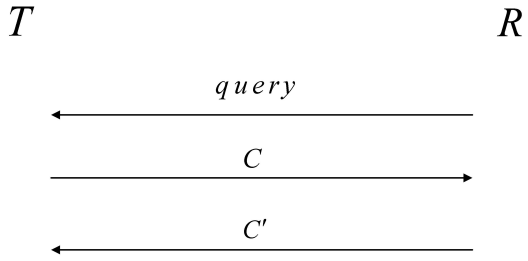


Figure 2.1: Protocol of RFID tags using universal re-encryption.

Figure 2.1 shows the protocol of RFID tags using universal re-encryption. Any-party let alone an authorized R can write C' into T in this protocol.

2.4.2 Security Properties of UR

Golle *et al.* [13] define the first vulnerability when re-encryption is used for RFID. Saito *et al.* [34] (SAITO04) point out the second and third vulnerabilities, and then find two solutions on the first and second vulnerabilities that mentioned in each paper. However, SAITO04 fails to find a solution on the third vulnerability.

The first attack suggested by Golle *et al.* is as follows: if $C = [(\alpha_0,\beta_0);(\alpha_1,\beta_1)]$ $= [(\alpha'_0,\beta'_0);(1,1)]$ is written into T, re-encrypted C' is same with C; that is, T is completely traceable since response C from T is never changed.

The second attack suggested by Saito *et al.* is as follows: if the adversary writes C like $C_A = [(\alpha_0,\beta_0);(\alpha_1,\beta_1)] = [(m_A y_A,g_A;\ (y_A,g_A)])$ encrypted with the adversary's PK $y_A = g_A^{x_A}$, and then the adversary can decrypt C'_A with the adversary's SK x no matter how many times authorized parties re-encrypt C. The second and third attacks exploit (α_1,β_1) although the plaintext is included in (α_0,β_0) where $m_0 = \alpha_0/\beta_0^x$.

The third attack suggested by Saito *et al.* is as follows: The adversary can change m with different one using (α_1,β_1) of C previously sent by T or an authorized R not to exploit the second and third attacks. For example, the

decrypted plaintext can be m_A if the adversary writes C like $C_A = [(\alpha_0,\beta_0); (\alpha_1,\beta_1)] = [(m_A,1); (\alpha_A,\beta_A)]$). In this case, T can not be used for the adversary and authorized ones. The third attack exhibits the same effect with just writing the meaningless garbage value into T, which is DoS; so, we will consider the garbage value not the third attack.

The first and second attacks are used for tracking, while the garbage value is for DoS; more seriously, UR has the vulnerability on a swapping attack against which PIN can be used for protecting. In this thesis, we use PIN to protect a swapping attack; Juels *et al.* also suggest protecting against a swapping attack in [21].

Chapter 3
Our Protocol without Proxy

In this section, we propose a scalable and untraceable RFID authentication protocol based on hash function.

3.1 Main idea

Our main idea is to use a shared secret key k which assumed to be writable and non-readable when R sends a query to T; k is written as a new value when enrolling tags in the system or doing ownership transfer while ID_i is updated as ID_{i+1} when successful mutual authentication happens with only authorized readers.

3.2 Our Proposed Protocol

Our protocol is shown in Figure 3.1. TD05 does not guarantee UNT-DVS; and so, we suggest a protocol to make up for the weakness of TD05. In addition, we propose how R communicates with T using timestamp to prevent replay attack without implementing time clock in T unlike TD05.

3.2.1 Initialization and Assumption

Any T has four non-volatile memories ID_0, k, access PIN and TS_{last} which are initialized into the memory of T during manufacturing process; ID_0, pseudo-EPC, which is produced by hash function or the other encoding schemes,

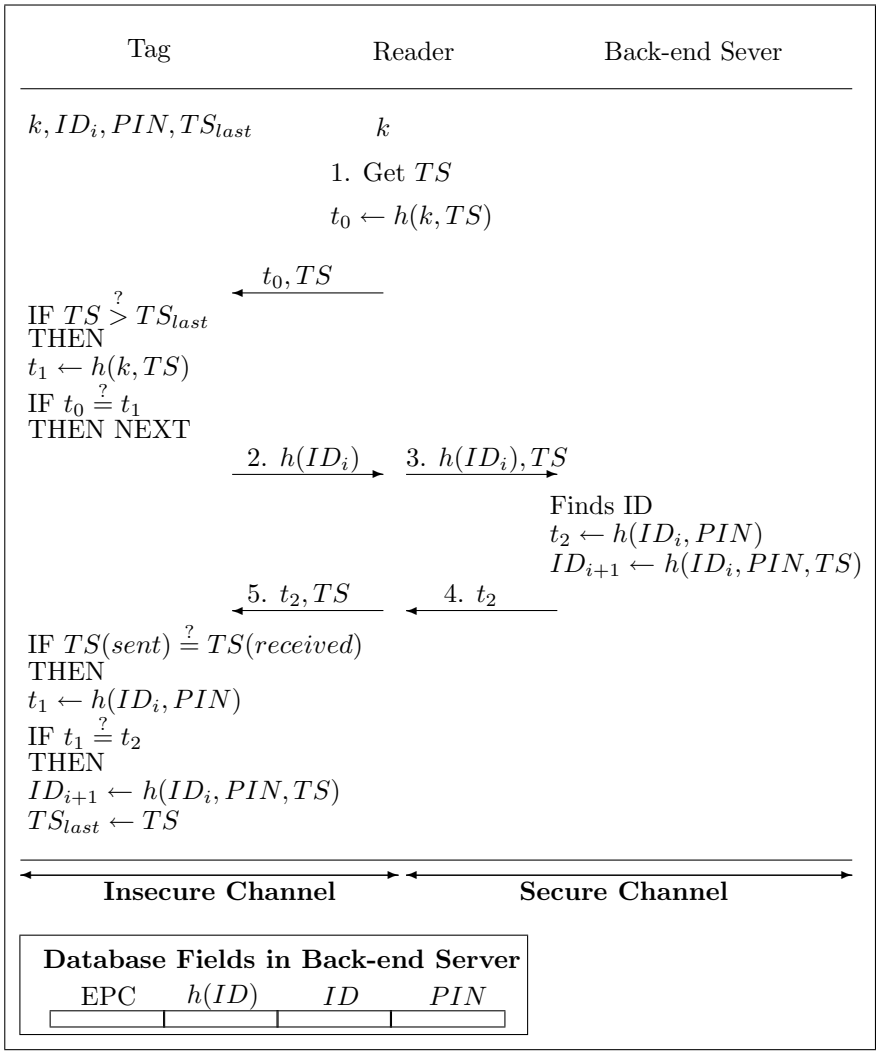


Figure 3.1: Our protocol without proxy

is written into the memory of T; access PIN is written into the reserved memory of T; k is written into the memory of T; TS_{last} is set to 0 during the initialization process. TS_{last} is updated with TS sent by an authorized R to prevent replay attack after successful mutual authentication. R only has k which is stored during manufacturing process or ownership transfer. S keeps four fields: EPC, $h(ID_i)$, ID_i, and access PIN; ID_i and access PIN are

shared between T and S, while EPC and $h(ID_i)$ are not.

In our protocol, we assume that S can tell an authorized R from an unauthorized one; the clock which is built in R is tightly synchronized like the mobile phone in multi-tag-reader environment.

3.2.2 Protocol Description

Step 1 R gets TS from its timestamp information. R computes $h(k, TS)$, and then transmits $h(k, TS), TS$ to T. T compares TS and TS_{last}. If TS is greater than TS_{last}, then T generates $h(k, TS)$ using TS and k. Otherwise, T considers it as an unauthorized request. If the value received is the same as the value computed, they authenticate R as an authorized one. The step 1 is quite different from the other protocols: the other protocols authenticate R at the last steps(4 - 5) while our protocol authenticates R at the step 1. In other words, $\mathcal{T}_k$ responds to R while $\mathcal{T}_k'$ does not respond.

Step 2 T sends $h(ID_i)$ to R, which reduces time complexity to $O(\beta)$ in multi-tag-reader environment because all of the tags respond to a query of R in the previous protocols at all times while only $\mathcal{T}_k$ responds in our protocol.

Step 3 R forwards $h(ID_i)$ and TS to S. S finds ID_i; S computes $h(ID_i, PIN)$ using ID_i and PIN; S updates ID_i to ID_{i+1} where $ID_{i+1} = h(ID_i, PIN, TS)$. Otherwise, S stops the procedure.

Step 4 S sends $h(ID_i, PIN)$ to R.

Step 5 R forwards $h(ID_i, PIN)$ and TS to T. T compares received and sent TS. If two values equal, T also computes $h(ID_i, PIN)$ and compare the received and value with the computed one. If all comparisons are successful, T updates ID_i to ID_{i+1} like S does; T also updates TS_{last}. Otherwise, T stops the procedure.

3.3 Security and Performance Analysis

In this section, we analyze the security of our protocol including the protocols in Table 3.1.

Table 3.1: Comparison with other protocols.

Protocol	HLS03 [41]	RHLS03 [41]	OSK03 [28]	TD05 [8]	LACP05 [24]	Our Protocol
Forward Secrecy	×	×	◯	◯	×	◯
Untraceability during valid session	×	◯	◯	×	×	$*$
Untraceability	×	△	◯	△	△	$\star$
Scalability	$O(1)$	$O(n)$	$O(2mn)$	$O(1)$	$O(1)$	$O(1)$
Scalability in multi-tag-reader environment	$O(\gamma)$	$O(n\gamma)$	$O(4mn\gamma)$	$O(\gamma)$	$O(\gamma)$	$O(\beta)$
Hash operations	0	1	2	$5+\alpha$	2	4
Prevent Spoofing R	×	×	◯	◯	◯	◯
Prevent Spoofing T	×	×	×	◯	×	◯
Synchronization	NA	NA	△	◯	◯	◯

†† Notations

◯	satisfied	△	partially satisfied
×	not satisfied	$*$	if k is revealed, ×. Otherwise, ◯
		$\star$	if k is revealed, △. Otherwise, ◯

- **Synchronization.** The simplified TD05 protocol happens to desynchronization problem. TD05 protects desynchronization between S and tags at the last step in enhanced TD05 protocol. We, however, don't need the last step to avoid desynchronization since our protocol emits a query with shared secret k which is used to authenticate R. On the

other hand, although the memory channel is read by $\mathcal{A}$ once; we guarantees synchronization between tags and S even though $\mathcal{A}$ knows k and $h(ID_i)$. The reason why we should use TS is discussed in [8].

- **Forward Secrecy.** Our protocol updates ID_i to ID_{i+1} using a one-way function $h()$ like OSK03 and TD05. As long as there is no alternative, we have to use one-way function to guarantee forward secrecy.

- **Untraceability during a valid session.** Tags authenticate R after receiving the first message, and then tags respond to only an authorized query of R. Therefore, tags do not respond to R with different k. As a result, tags are untraceable during a valid session since $\mathcal{A}$ doesn't impersonate even in the step 1.

- **Untraceability.** Tags authenticate R after receiving the first message; R authenticates the tags after receiving the second message. In each step, tags and R authenticate counterpart to remove traceability. In addition, although $\mathcal{A}$ knows k, $\mathcal{A}$ can not trace a particular tag since tag responses to query is always different at the valid session.

- **Scalability.** This is most big contribution in our work. S has time complexity $O(\beta)$ to find a tag in multi-tag-reader environment. This result is the best complexity in comparison with the previous protocols. Time complexity of each protocol changes in multi-tag-reader environment (See Table 3.1); from $O(1)$ to $O(\gamma)$ in most cases, from $O(1)$ to $O(\beta)$ in ours where $\beta < \gamma < n$.

- **Spoofing the tag.** As long as $\mathcal{A}$ doesn't know the value of k, $\mathcal{A}$ can not spoof the tags in our protocol. If $\mathcal{A}$ tampers with a tag, then $\mathcal{A}$ can spoof the tags at the step 1. However, S finds out that $\mathcal{A}$ is not an authorized R in the end. There is no way to spoof the a tag unless $\mathcal{A}$ knows k and ID_i.

- **Spoofing the reader.** As long as $\mathcal{A}$ doesn't know the ID_i, $\mathcal{A}$ can not spoof R since tag response to a query by R will be different at all times.

- **Item Privacy.** The party who has EPC is only S in our protocol; that is, we guarantee item privacy as long as S is not compromised. The other previous protocols are also designed to meet item privacy if HLS03, RHLS03, OSK03, TD05, and LACP05 hold following three conditions: only S has EPC, T doesn't have EPC, ID is not a EPC itself.

- **Performance Analysis.** Our protocol is more secure than TD05 in terms of traceability aspects even though ours reduces hash operations from five and more to four. In our protocol, tag needs four hash operations to communicate with R with quite good security performance. Under the assumption that tags can not be tampered, we don't need to send last message.

- **Ownership Transfer.** We supports ownership transfer using k. As far as we know, ownership transfer issue is dealt with only in [27] so far. For example, Alice has R that has k which is also stored in tagged items of Alice. When Alice gets some tagged items from Bob, Alice can write her own k.

Chapter 4
Our Protocol with Proxy

4.1 Overview and Main idea

In this chapter, we propose an off-tag access control mechanism[1] using an external device which has scalability and untraceability. Off-tag access control provides a chance to be widespread with low-cost tags since the external device takes care of almost high-cost computations instead of T.

Table 4.1 shows various countermeasures and their examples that have been proposed. Deactivation by permanent and temporary tags is analogous to power-off of personal computers due to the fear of being cracked. In other words, these can not be an eventual solution. On-tag cryptographic primitive and on-tag access control require high-end RFID tags; that is, they are not reasonable to implement RFID tags so far. Low-cost is the most important factor to proliferate RFID technology into the billions of items.

Our proposed protocol has three properties (RFID tags with a pseudorandom number generator, exploiting universal re-encryption and a proxy) with main research goals (lightweight RFID tags and ownership transfer).

4.1.1 Initialization and Assumption

We assume the followings.

1. PKI (Public Key Infrastructure) is established,

[1] On-tag means that mechanisms are located on the RFID tags themselves; in contrast, off-tag is taken care of by the external device.

Countermeasure	Example
Permanent tag deactivation	kill command[47], tag destruction
Temporary tag deactivation	Faraday cages, sleep/wake command
On-tag cryptographic primitive	stream ciphers, asymmetric or symmetric cryptographic algorithm[10]
On-tag access control	hash-based[41], pseudonym-based[18, 13], tree-based schemes[27, 9]
Off-tag access control	blocker[20, 17], noisy tag[6], proxy-based schemes[21, 32]

Table 4.1: Countermeasures for preventing attacks in RFID systems

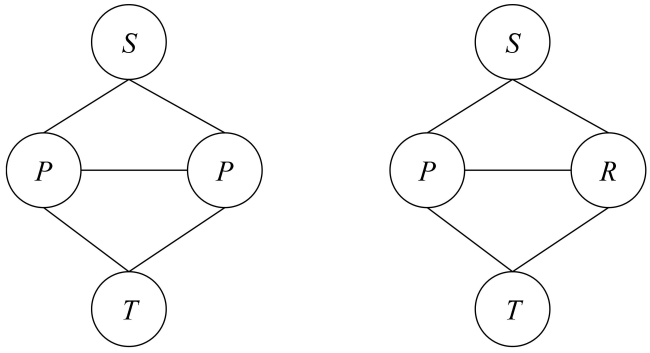


Figure 4.1: All possible channels in our protocol with a proxy

2. One proxy manages only one tag.

3. Proxy is within backward channel which is the operating range of T.

4. All channels are insecure. The possible channels are depicted in Figure 4.1 in which the solid line represents an insecure channel.

P has four database fields: Private key, Tag identifier[2], PIN, Server Location for each tag; Server Location field for each tag can contribute to reducing the work of S. In our protocol, the back-end server has to find a server location if

[2] ID, pseudo-EPC, tag identifier, and m are the same meaning in our protocol.

Action	Source
Pass level **A**	List of readers which have authorization level **A** for some tags
Pass level **B**	List of readers which have authorization level **B** for some tags
…	…
No answer	The others

Table 4.2: Access control list.

SL is a NULL value where SL denotes a server location for T. P also has an access control list. An example of an access control list is described in Table 4.2. **A** and **B** are used to represent the data access authorization level of R. S can transfers fine granular information of T based on granular data access authorization level; the degree of level depends on the system designer.

T has a pseudorandom number generator and memory storages to store PIN and C; C is based on ElGamal encryption algorithm. Any other cryptographic primitives like hash or symmetric or asymmetric algorithm do not need.

The owner of T is a person who carries and owns a proxy and all tags which is managed by the proxy.

S has six database fields: Private key, Public key, EPC, Tag identifier, Tag owner and Data; SK and PK can be generated and managed by S or the other trusted entities since R does not send messages included SK or PK, Tag owner field is used for ownership transfer, Data field supports fine granular data access authorization level.

4.1.2 How the proxy works

A proxy, P is used for *personal usage* like RFID Guardian (GUARDIAN05) [32]. P is a reader which can be integrated into cellular phones, PDAs (per-

sonal Digital Assistants) or tiny portable device which manages owner's tags; P also enforces privacy policy desired by its owner using a access control list. In our proposed protocol, P should exist around his own tags; so, the operating range of P works around 1 or 2 meters which is approximately from head to toe of an individual.

Juels (REP05) [21]'s proxy and Rieback's GUARDIAN05 meet four different security properties; REP05 has tag acquisition, tag relabelling, tag simulation and tag release; GUARDIAN05 has auditing, key management, access control and authentication. P has six functional security properties which are depicted in Figure 4.2 in which an arrow represents a state transition; these properties in our protocol are a little different with REP05 and GUARDIAN05. The description of each component is as follows:

- *Tag acquisition* : P gets a new SK corresponding to PK and ID of T from S; P also gets PIN from the previous tag owner's P. P generates C, and then writes C and PIN' into the acquired memory of T when P acquires T.

- *Information management* : P manages ID of T, SK, PIN and a server location for each T. P inserts the record in a database when it acquires T; P deletes the record about T when it releases T.

- *Relabeling* : P relabels T contents whenever the other devices try to write data into T managed by P, which means that P writes C' into T.

- *Authentication* : P checks whether the queried R is an authorized party or an unauthorized one.

- *Access control* : If an authorized party has sent a query, then P checks a data access authorization level and passes the proper message for level. Access control can considers three cases with P: which R, which T, which circumstances like GUARDIAN05 (See more details in [21]).

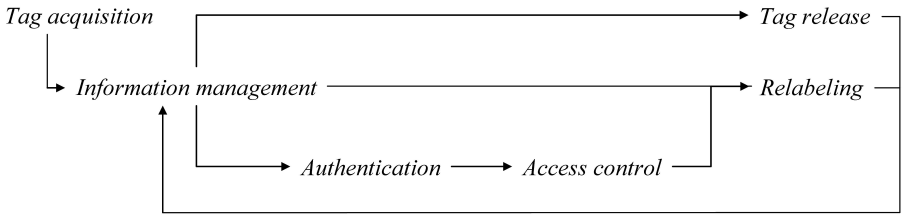


Figure 4.2: Six functional security properties of P.

- *Tag release* : An owner of T releases T when the owner of T does not want to keep his T any more; that is, ownership transfer happens.

4.1.3 Proposed Protocol

In SAITO04 [34] which is one of on-tag access control scheme, T checks the first and second attacks by itself. Exponential computation is needed to check the second attack; however, it is big overhead on T. SAITO04 checks only the contents written in T not to authenticate R; that is, anybody can get information of T upon receiving C from T while we authenticate R exploiting the external device on behalf of T.

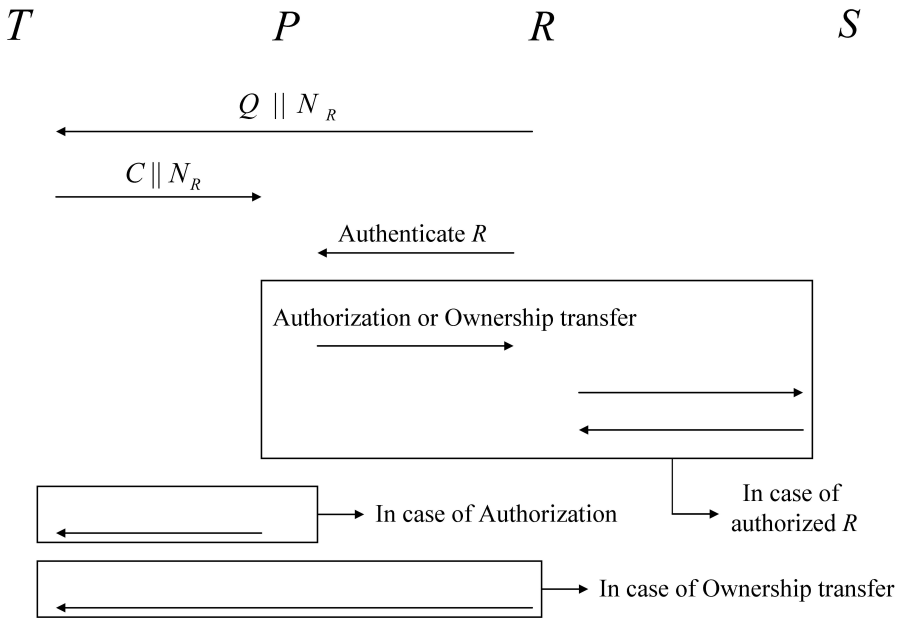
Our protocol is shown in Figures 4.3, 4.4 and 4.5; Figure 4.3 shows our protocol, Figure 4.4 shows our protocol for authorization, Figure 4.5 shows our protocol for ownership transfer.

Our overall protocol works as follows:

Step 1 R sends Q query and random nonce N_R generated by R to T.

Step 2 T sends C and N_R to P. P decrypts C with private key SK x.

Step 3 The way to communicate between R and P is using a variety of out-of-band or in-band means preferably over the secure channel (See more details in [21]. In our protocol, R sends its information like $Sig_R(N_R)||Cert_R$ to P.



Database Fields in Proxy

Private key	Tag identifier(m)	PIN	Sever Location

Database Fields in Back-end Server

Private Key	Public Key	EPC	Tag identifier(m)	Tag owner	Data
					DataA||DataB|| …

Figure 4.3: Our protocol with a proxy

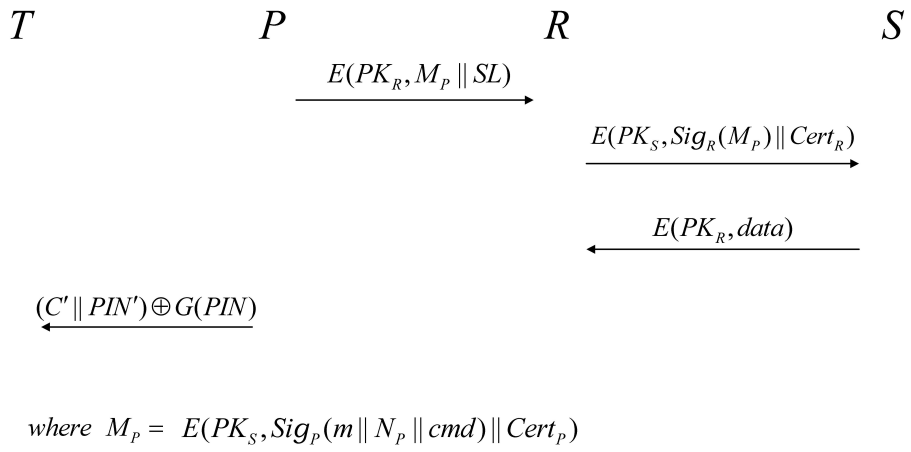


Figure 4.4: Our protocol with a proxy for authorization

Step 4 P checks whether R is authorized or not using an access control list, and checks data access authorization level in case of authorized R. As another case, ownership transfer happens in Step 4; ownership transfer is unusual case, so it require human interaction to do ownership transfer.

Step 5 Protocol descriptions for authorization and ownership transfer is handled with in each protocol. For an unauthorized R, P sends random value to R, which can not give a chance for the adversary to distinguish the tag from the other tags. In case of authorization protocol, P relabels the contents of T while R relabels the contents of T in case of ownership transfer protocol; the detail description is described in each protocol.

Nonce (N_R and N_P) in our protocol is to ensure that old communications cannot be reused in replay attacks. Nonce can be time-variant or generated with enough random bits which ensure a probabilistically insignificant chance of repeating a previously generated value.

Our protocol in case of authorization works as follows:

Step 1 P sends $E(PK_R, M_P||SL)$ to R where M_P denotes $E(PK_S, Sig_P(m||N_P||cmd)||Cert_P)$; SL denotes a server location for T, N_P denotes a random nonce generated by P, cmd represents an authorization level, and m denotes a pseudo-EPC (ID of T) in our protocol. We recommend to use the pseudo-EPC rather than EPC ([37] states the reason for that)

Step 2 R decrypts $E(PK_R, M_P||SL)$ with the private key SK_R of R. R gets a server location, and sends $E(PK_S, Sig_R(M_P)||Cert_R)$ to S which is same with the server location.

Step 3 S decrypts $E(PK_S, Sig_R(M_P)||Cert_R)$, $Cert_R$, M_P, $Cert_P$ with the private key of S. S finds out the identities of P and R, ID of T, and an authorization level. S checks whether P is the owner of T or not. If P is the owner of T, then S checks the authorization level of R for T. For example, In case that an authorization level is **A**, S sends $E(PK_R, DataA)$ to R; In case that an authorization level is **B**, S sends $E(PK_R, DataA||DataB)$. The degree of an authorization level is decided by the system designer. If P is not the owner of T, S sends a random value to R to provide indistinguishability.

Step 4 P computes $G(PIN)$ and generates PIN' where G is a pseudorandom number generator and PIN is used for a seed; G is used for matching the bit size of $G(PIN)$ and $(C'||PIN')$. P selects a random encryption factor $r' = (k'_0, k'_1) \in Z_q^2$, re-encrypts C to $C' = [(\alpha'_0,\beta'_0); (\alpha'_1,\beta'_1)] = [(\alpha_0\alpha_1^{k'_0},\beta_0\beta_1^{k'_0}); (\alpha_1^{k'_1},\beta_1^{k'_1})]$, and sends $(C'||PIN') \oplus G(PIN)$ to T; lastly, updates PIN with PIN'.

Step 5 T computes $G(PIN)$ with PIN which is in the memory of T, performs $\oplus$ operation ($G(PIN)$ generated by T with $(C'||PIN') \oplus G(PIN)$ received from P), and can get C' and PIN'; lastly, T updates PIN with PIN' and C with C'.

Our protocol in case of ownership transfer works as follows:

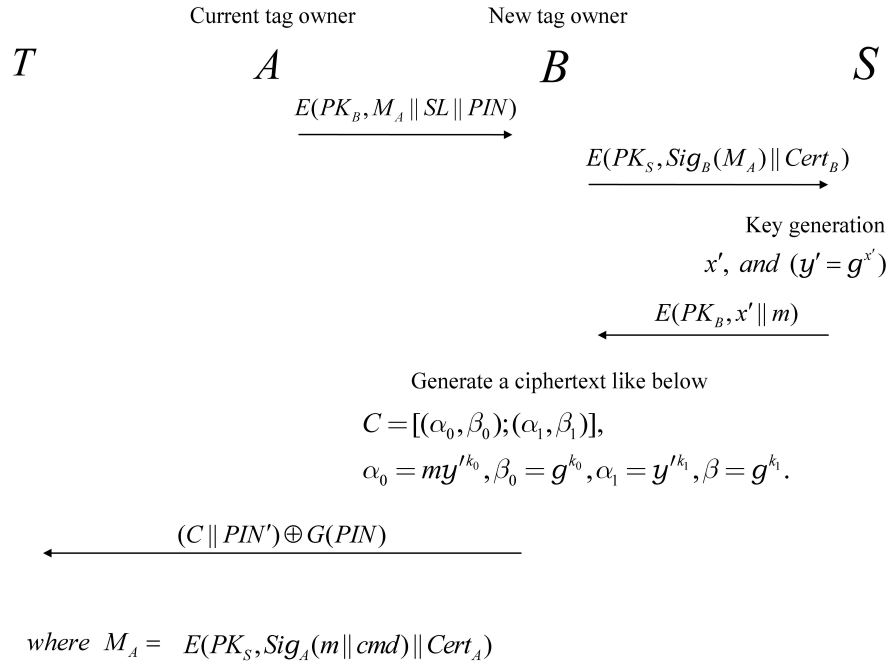


Figure 4.5: Our protocol with a proxy for ownership transfer

Step 1 A sends $E(PK_B, M_A||SL||PIN)$ to B where M_A denotes $E(PK_S, Sig_A(m||cmd)||Cert_A)$, A denotes the current tag owner, B denotes the new tag owner, and cmd represents `ownership transfer command`.

Step 2 B decrypts $E(PK_B, M_A||SL||PIN)$ with the private key of B. B gets a server location and PIN, and sends $E(PK_S, Sig_B(M_A)||Cert_B)$ to S.

Step 3 S decrypts $E(PK_S, Sig_B(M_A)||Cert_B)$, $Cert_B$, M_A, $Cert_A$ with the private key of S. S finds out the identities of and A and B, ID of T, and `ownership transfer command`. S checks where A is the owner of T or not. If P is the owner of T, then S generates SK and PK corresponding to SK. S updates previous key pairs with new key pairs for the tag and the previous tag owner with the new tag owner in the database. And then, S sends $E(PK_B, x||m)$ to B. If A is not the owner of T, S sends a random value to B. Lastly, B generate a new ciphertext.

Step 4 B computes $G(PIN)$ and generates PIN', selects a random encryption factor $r = (k_0, k_1) \in Z_q^2$, generates $C = [(\alpha_0,\beta_0); (\alpha_1,\beta_1)] = [(my'^{k_0},g^{k_0}); (y'^{k_1},g^{k_1})]$, and sends $(C||PIN') \oplus G(PIN)$ to T; lastly, B updates PIN with PIN'.

Step 5 T computes $G(PIN)$ with PIN which is in the memory of T, performs a $\oplus$ operation ($G(PIN)$ generated by T with $(C'||PIN')\oplus G(PIN)$ received from P), and can get C' and PIN'; lastly, T updates PIN with PIN' and C with C'.

After the ownership transfer protocol, B should perform operation over the secure channel so that PIN' is not eavesdropped by A when writing a new ciphertext. Nevertheless, it can be easily performed with secure way since P can control its operation range. For example, P writes PIN' and C with less than one centimeter operating range by physical contact.

4.2 Security and Performance Analysis

In this section, we check whether our protocol guarantees the following security requirements.

- ***Protection against tracing.*** T sends different message at any time R sends a query. C and C' is indistinguishable (See [13]), and `write command` of P is secure because the adversary doesn't have a way to know PIN. Even if the adversary gets PIN under tampering T, the adversary have to be within 1-2m to trace T at all time while the other almost all the previous protocols in the literature easily can be traced under tampering T. In addition, `write command` by physical contact guarantees updating PIN securely.

- ***Protection against cloning and spoofing.***

  Cloning T and spoofing R are meaningless since P maintains a private key and an access control list for each tag.

  Spoofing T is also meaningless because T doesn't have a way to check whether `write command` sent by some devices is authorized or not; since the adversary doesn't have any gains, the adversary does not try to spoof T. The adversary's `write command` make T replace PIN with PIN_A where PIN_A is the generated by the adversary; but, P also checks PIN_A and can writes re-encrypted ciphertext generated by P with the PIN_A.

- ***Privacy.*** We provide privacy since C emitted is provably secure since it is based on UR [13]. As another way to provide privacy, pseudo-EPC as ID of T should be used (See the more details in [37]); S has EPC and Tag identifier field to use pseudo-EPC. We support data access authorization level-based service, which enhances privacy for individual.

- ***Protection against DoS.*** DoS attack can cause battery consumption of P, which is one big problem when using the battery-powered device to protect T.

- ***Ownership transfer.*** We described the protocol for ownership transfer. Ownership transfer is one of the advanced security requirements; but, Molnar *et al.* [27] supports sophisticated ownership transfer to the best of our knowledge.

- ***Protection against swapping.*** Swapping attack is one of the vulnerabilities on UR. In our protocol, we prevent from swapping attack using PIN.

- ***Protection against two attacks and the garbage value in UR.*** P writes new C into T whenever the other devices try to write C, which means that T has always C generated by an authorized P unless the battery of an authorized P is totally consumed. Sleep / wake command can defend against two attacks and the garbage value even in case that the battery of P is totally consumed.

- ***Scalability.*** Since P sends m with encrypted form to authorized R which forwards received message to S, the complexity of tag identification on S is $O(1)$. In other words, S does not need computations related to non-relevant T, which means our protocol is *completely scalable*.

- ***Cost.*** T requires only one lightweight cryptographic primitive, a pseudorandom number generator, and re-writable memory to store C and PIN. Consequently, our protocol can be implemented with reasonable low-cost.

4.3 Comparison with Related Work

Selective RFID jamming [31] makes a signal jam up the airwaves under lots of queries from an unauthorized R while an external device just re-encrypts a new valid C in our protocol. In addition, the use of jamming signal is legally questionable.

REP and GUARDIAN send the secret value of T in unencrypted form, which is insecure since REP and GUARDIAN give the adversary a chance to eavesdrop secret values while our P does not reveal the secret information of T.

SAITO04 has several weaknesses: 1) big overhead on T, 2) tracking with only eavesdropping within forward channel, 3) no R authentication mechanism, 4) allowing swapping attack which is the venerability of UR. Unlike SAITO04, we resolve all the problems of SAITO04 using P and PIN. In Table 4.3, we show the comparison table with the previous schemes.

Table 4.3: Comparison to the previous schemes.

Scheme	Golle's [13]	SAITO04 [34]	Selective RFID Jamming [31]	Our Protocol
External device	No	No	Yes	Yes
Jamming signal	No	No	Yes	No
Prevent two attacks	×	◯	NA	◯
Prevent swapping attack and garbage value	×	×	NA	◯
R authentication mechanism	×	×	◯	◯
Secure over forward channel	×	×	×	◯
Untraceability	×	×	×	◯

†† Notations

◯ satisfied
× not satisfied

Chapter 5
Conclusion

There is a trade-off between scalability and untraceablility in RFID authentication protocol; therefore, many literatures did not suggest a protocol which guarantees scalability and untraceability together. However, in this thesis, we propose two scalable and untraceable protocols enabling ownership transfer.

Our protocol without proxies has several security properties as followings:

- supports ownership transfer
- considers multi-tag-reader environment
- receives messages from the tags what a reader wants.

Our protocol with a proxy Supports several security properties as followings:

- ownership transfer
- granular data access
- scalability
- untraceability
- privacy
- protection against several attacks which are spoofing, cloning, and swapping
- the untraceable way even under compromising a tag
- the more fast way to find a server location

As extra contributions, we deal with what item privacy is, why item privacy is important and how the way guaranteeing item privacy can be applied to our protocol. Consequently, we make sure that our proposed schemes can contribute to make RFID deployment widespread.

확장 가능하고 추적 불가능한 RFID 인증 프로토콜에 관한 연구

서영준

RFID (Radio Frequency IDentification)은 편리성과 경제성으로 인해 최근에 많은 주목을 받고 있으며 유비쿼터스 사회로 진화하기 위한 핵심적인 역할을 할 것으로 기대된다. 또한, RFID는 바코드 시스템의 여러가지 문제점을 해결할 수 있기 때문에 바코드를 대체하기 위한 하나의 새로운 기술로 보인다. RFID는 무한한 경제적인 잠재력을 가지고 있다. 예를 들어, 10억개의 태그가 부착된 물품이 있다고 하면, 바코드와 태그의 가격 차이가 1센트일시에 100만달러의 경제적 이득을 가져다 줄 것이다. 이것은 태그가 모든 종류의 물품에 부착될 수 있는 특성에 기인한다.

반면에, RFID 기술은 다양한 공격으로부터 침해당할 수 있으며, 이러한 문제점들은 RFID가 널리 전개되는 것을 방해하는 요소이다. 특히 이러한 문제점들 중 이 논문에서는 불추적성과 확장성에 초첨을 맞추며, 왜 불추적성과 확장성을 동시에 만족하는 것이 RFID 기술을 널리 전개하는데 중요한가를 다룰 것이다. 공격자가 태그가 부착된 물품을 추적하게 되면 개인의 사생활을 침해할 소지가 크다. 이러한 특성으로 인해 베네톤을 조직한 Albrecht는 RFID 태그를 스파이 칩으로 부르고 불매운동을 하였다. 게다가, 고유의 ID를 가진 태그는 개인의 정체성과 연결될 수도 있으며, Garfinkel등은 [12]에서 RFID에 대한 여러가지 개인의 사생활 침해에 대한 위협들을 다루었다.

불추적성과 확장성은 상보관계에 있으나, 우리는 추적이 불가능한 프로토콜을 고안하면서 동시에 확장성 (RFID 시스템상에서 얼마만큼의 태그의 수를 수용할 있는가를 뜻하며, 이것을 달성하기 위해서는 하나의 태그 식별자를 찾을 때 후방서버상에서의 연산이 다른 태그와의 영향을 받지 않아야한다.)을 만족하기위한 프로토콜을 고안하였다. 예를 들어, 태그의 응

답이 태그의 식별자에 대한 정보를 포함하지 않는 프로토콜에서는 태그의 식별자를 찾기 위해서 전체탐색이 요구되므로 데이타베이스에서 확장성이 좋지 못하다. 반면에, 태그의 응답이 태그의 식별자에 대한 정보를 포함한다면 적 또한 허가받은 사용자와 같은 입장에 놓이기 되므로 태그가 부착된 물품을 추적할 수 있다.

이전의 프로토콜들 [14, 44, 24, 8, 39]과 해쉬 락 안 [41]은 확장성이 좋았으나 추적성에 문제가 있었다. Rhee등의 논문 [30], Ohkubo등의 논문 [28], 임의의 해쉬 락 안 [41]은 추적이 불가능하였으나 확장성이 좋지 못하였다. 그리하여 이 논문에서는 확장성과 불추적성을 동시에 만족시키는 프로토콜을 고안하고자 노력하였고, 그러한 RFID 인증 프로토콜 두개를 제시한다. 그 하나는 태그 대리자를 이용한 것이며, 다른 하나는 대리자를 이용하지 않은 것이다.

태그 대리자를 이용하지 않은 프로토콜은 다음과 같은 장점을 가진다. (1) 소유권 이전을 지원한다. (2) 멀티 태그-리더 환경을 고려하였다. (3) 리더가 원하는 메세지만을 태그들로부터 받을 수 있다.

태그 대리자를 이용한 프로토콜은 다음과 같은 장점을 가진다. (1) 소유권 이전을 지원한다. (2) 모든 채널이 안전하지 않은 채널이라고 가정한다. (3) 데이터 접근 인가 레벨에 근거한 서비스를 지원한다.

또한, 이 논문에서는 아이템 프라이버시라는 용어와 그 중요성에 대해서 다룬다.

References

1. Gildas Avoine. Security and privacy in rfid systems, http://lasecwww.epfl.ch/ gavoine/rfid/

2. Gildas Avoine and Philippe Oechslin, "A Scalable and Provably Secure Hash based RFID Protocol", *In International Workshop on Pervasive Computing and Communication Security – PerSec 2005*, pp.110-114, Mar. 2005, IEEE Computer Society Press, Kauai Island, Hawaii, USA.

3. Gildas Avoine and Philippe Oechslin, "RFID traceability: A multilayer problem" In Andrew Patrick and Moti Yung, editors, *Financial Cryptography – FC'05, LNCS 3570*, pp.125-140, Feb.-Mar. 2005, Springer-Verlag, Roseau, The Commonwealth Of Dominica.

4. Mihir Bellare, Ran Canetti and Hugo Krawczyk, "Keying hash functions for message authentication" *Advances in Cryptology – Crypto 96, LNCS 1109*, pp.1-15, Aug. 1996, Springer-Verlag, California, USA.

5. Steve Bono, Matthew Green, Adam Stubblefield, Ari Juels, Avi Rubin, and Michael Szydlo, "Security analysis of a cryptographically-enabled RFID device", *14th USENIX Security Symposium*, Jul.-Aug., 2005, Baltimore, Maryland, USA.

6. Claude Castelluccia and Gildas Avoine, "Noisy Tags: A Pretty Good Key Exchange Protocol for RFID Tags", *International Conference on Smart Card Research and Advanced Applications - Cardis, LNCS 3928*, Apr. 2006, Tarragona, Spain.

7. Eun Young Choi, Su Mi Lee and Dong Hoon Lee, "Efficient RFID Authentication protocol for Ubiquitous Computing Environment", *Interna-*

tional Workshop on Security in Ubiquitous Computing Systems – secubiq 2005, LNCS 3823, pp.945-954, Dec. 2005, Springer-Verlag, Nagasaki, Japan.

8. Tassos Dimitriou, "A Lightweight RFID Protocol to protect against Traceability and Cloning attacks", *Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm'05*, pp.59-66, Sep. 2005, Athens, Greece.

9. Tassos Dimitriou, "A Secure and Efficient RFID Protocol that could make Big Brother (partially) Obsolete", *International Conference on Pervasive Computing and Communications – PerCom 2006*, Mar. 2006, Pisa, Italy.

10. Martin Feldhofer, Sandra Dominikus, and Johannes Wolkerstorfer, "Strong authentication for RFID systems using the AES algorithm". *Workshop on Cryptographic Hardware and Embedded Systems – CHES' 04, LNCS 3156*, Jul. 2004, Boston, Massachusetts, USA.

11. Xingxin (Grace) Gao, Zhe (Alex) Xiang, Hao Wang, Jun Shen, Jian Huang and Song Song, "An Approach to Security and Privacy of RFID System for Supply Chain", *Conference on E-Commerce Technology for Dynamic E-Business*, pp.164-168, Sep. 2005, IEEE Computer Society, Beijing, China.

12. Simson L. Garfinkel, Ari Juels and Ravi Pappu, "RFID Privacy: An Overview of Problems and Proposed Solutions", *IEEE SECURITY and Privacy*, pp.34-43, May-Jun. 2005.

13. Philippe Golle, Markus Jakobsson, Ari Juels and Paul Syverson. "Universal Re-encryption for Mixnets", *The Cryptographers' Track at the RSA Conference – CT-RSA, LNCS 2964*, Feb. 2004, San Francisco, California, USA.

14. Dirk Henrici and Paul Müller, "Hash-based Enhancement of Location Privacy for Radio-Frequency Identification Devices using Varying Identifiers", *International Workshop on Pervasive Computing and Communication Security – PerSec 2004*, pp.149-153, Mar. 2004, IEEE Computer Society, Orlando, Florida, USA.

15. Ari Juels, "RFID Security and Privacy: A Research Survey", *IEEE Journal on Selected Areas in Communication*, Vol. 24, No. 2, pp. 381-394, Feb. 2006.

16. Ari Juels and Stephen Weis, "Defining strong privacy for RFID", *Cryptology ePrint Archive, Report 2006/137*, 2006.

17. Ari Juels, Ronald Rivest and Michael Szydlo, "The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy", *Conference on Computer and Communications Security - ACM CCS*, Oct. 2003, Washington, DC, USA.

18. Ari Juels, "Minimalist Cryptography for Low-cost RFID Tags", In C. Blundo and S. Cimato, editors, *The Fourth International Conference on Security in Communication Networks – SCN 2004, LNCS 3352*, pp.149-164, Sep. 2004, Springer-Verlag, Amalfi, Italia.

19. Ari Juels, "Yoking Proofs for RFID Tags", *The First International Workshop on Pervasive Computing and Communication Security*, pp.138-143, Mar. 2004. IEEE Press, Orlando, Florida, USA.

20. Ari Juels and John Brainard, "Soft Blocking: Flexible Blocker Tags on the Cheap", *Workshop on Privacy in the Electronic Society - WPES*, Oct. 2004, Washington, DC, USA.

21. Ari Juels, Paul Syverson and Dan Bailey, "High-power Proxies for Enhancing RFID Privacy and Utility", *Workshop on Privacy Enhancing Technologies – PET 2005*, May-Jun. 2005, Dubrovnik, Croatia.

22. Soo-Cheol Kim, Sang-Soo Yeo, Sung Kwon Kim, "MARP: Mobile Agent for RFID Privacy Protection", *International Conference on Smart Card Research and Advanced Applications - Cardis, LNCS 3928*, pp.289-299, Apr. 2006, Springer-Verlag, Tarragona, Spain.

23. Ziv Kfir and Avishai Wool, "Picking virtual pockets using relay attacks on contactless smartcard systems", *1st International Conference on Security and Privacy for Emerging Areas in Communication Networks – SecureComm*, Sep. 2005, Athens, Greece.

24. Su-Mi Lee, Young Ju Hwang, Dong Hoon Lee and Jong In Lim, "Efficient Authentication for Low-Cost RFID Systems", *International Conference on Computational Science and its Applications - ICCSA 2005, LNCS 3480*, pp.619-627, May 2005, Springer-Verlag, Singapore.

25. Mikko Lehtonen, Thorsten Staake, Florian Michahelles, and Elgar Fleisch, "From Identification to Authentication A Review of RFID Product Authentication Techniques", *Ecrypt Workshop on RFID and Lightweight Crypto – RFIDSec 2006*, Jul. 2006, Graz, Austria.

26. Zhaoyu Liu and Dichao Peng, "True random number generator in RFID systems against traceability", *IEEE Consumer Communications and Networking Conference – CCNS*, IEEE, pp.620-624, Jan. 2006, Las Vegas, Nevada, USA.

27. David Molnar, Andrea Soppera and David Wagner, "A Scalable, Delegatable Pseudonym Protocol Enabling Ownership Transfer of RFID Tags", *Selected Areas in Cryptography – SAC 2005, LNCS 3897*, pp.276-290, Aug. 2005, Springer-Verlag, Kingston, Canada.

28. Miyako Ohkubo, Koutarou Suzuki and Shingo Kinoshita, "Cryptographic Approach to Privacy-friendly Tags", *In RFID Privacy Workshop*, 2003, MIT, USA.

29. Selwyn Piramuthu, "On Existence Proofs for Multiple RFID Tags", *IEEE International Conference on Pervasive Services, Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing – SecPerU 2006*, Jun. 2006, IEEE Computer Society Press, Lyon, France.

30. Keunwoo Rhee, Jin Kwak, Seungjoo Kim and Dongho Won, "Challenge-Response based RFID Authentication Protocol for Distributed Database Environment", *International Conference on Security in Pervasive Computing – SPC 2005, LNCS 3450*, pp.70-84, Apr. 2005, Springer-Verlag, Boppard, Germany.

31. Melanie R. Rieback, Bruno Crispo and Andrew S. Tanenbaum, "Keep on Blockin' in the Free World: Personal Access Control for Low-Cost RFID Tags", *International Workshop on Security Protocols – IWSP'05*, Apr. 2006, Cambridge, England.

32. Melanie R. Rieback, Bruno Crispo, and Andrew S. Tanenbaum, "RFID Guardian: A Battery-Powered Mobile Device for RFID Privacy Management", *Australasian Conference on Information Security and Privacy – ACISP'05, LNCS 3574*, Jul. 2005, Springer-Verlag, Brisbane, Australia.

33. Melanie R. Rieback, Bruno Crispo, and Andrew S. Tanenbaum, "The Evolution of RFID Security", *IEEE Pervasive Computing*, pp. 62-69, 2006.

34. Junichiro Saito, Jae-Cheol Ryou and Kouichi Sakurai, "Enhancing Privacy of Universal Re-encryption Scheme for RFID Tags", *Embedded and Ubiquitous Computing – EUC 2004, LNCS 3207*, Aug. 2004, Aizu-Wakamatsu City, Japan.

35. Junichiro Saito and Sakurai Kouichi, "Grouping Proof for RFID Tags", *The 19th International Conference on Advanced Information Networking and Applications – AINA'05*, pp. 621-624, Mar. 2005, IEEE, Taiwan.

36. Sanjay Sarma, Stephen Weis, and Daniel Engels, "RFID systems, security and privacy implications", *4th International Workshop on Cryptographic Hardware and Embedded Systems – – CHES 2002, LNCS 2523*, pp. 454-469, Aug. 2002, Springer-Verlag, Redwood Shores, CA, USA.

37. Youngjoon Seo, Hyunrok Lee and Kwangjo Kim, "A Scalable and Untraceable Authentication Protocol for RFID", *The Second International Workshop on Security Ubiquitous Computing Systems – secubiq 2006, LNCS 4097*, pp.252-261, Aug. 2006, Springer-Verlag, Seoul, Korea.

38. Gene Tsudik, "YA-TRAP: Yet Another Trivial RFID Authentication Protocol", *International Conference on Pervasive Computing and Communications – PerCom 2006*, Mar. 2006, IEEE Computer Society Press, Pisa, Italy.

39. Pim Tuyls and Lejla Batina, Lejla, "RFID-Tags for Anti-Counterfeiting", *Topics in Cryptology – CT-RSA 2006, LNCS 3860*, pp.115-131, Feb. 2006, Springer-Verlag, San Jose, CA, USA.

40. Xiaoyun Wang, Yiqun Lisa Yin and Hongbo Yu, "Finding Collisions in the Full SHA-1", *Advances in Cryptology – Crypto 2005, LNCS 3621*, Aug. 2005, Santa Barbara, California, USA.

41. Stephen Weis, Sanjay Sarma, Ronald Rivest and Daniel Engels, "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems", *Conference on Security in Pervasive Computing – SPC 2003, LNCS 2802*, pp.454-469, Mar. 2003, Springer-Verlag, Boppard, Germany.

42. Stephen Weis, "Radio-frequency identification security and privacy", Master thesis, M.I.T., Jun. 2003.

43. Stephen Weis, "Security parallels between people and pervasive devices", *In International Workshop on Pervasive Computing and Communication Security – PerSec 2005*, Mar. 2005, Kauai Island, Hawaii, USA.

44. Kirk Wong, Patrick Hui and Allan Chan, "Cryptography and Authentication on RFID Passive Tags for Apparel Products", *Computers in Industry*, Nov. 2006, Elsevier Science, Article In press.

45. Nigel Wood, "Global Supply Chain GTIN & RFID Standards II", *EPC Global Standards Development*, EPCglobal Canada, Oct. 14, 2004.

46. Jeongkyu Yang, Jaemin Park, Hyunrok Lee, Kui Ren and Kwangjo Kim, "Mutual Authentication Protocol for Low-cost RFID", *Ecrypt Workshop on RFID and Lightweight Crypto*, pp.17-24, Jul. 2005, Graz, Austria.

47. EPCglobal Inc. Class 1 generation 2 UHF air interface protocol standard version 1.0.9. Referenced 2005 at http://www.epcglobalinc.com/standards technology/EPCglobalClass-1Generation- 2UHFRFIDProtocolV109.pdf.

48. "Navigating the New Era of RFID", Article in EPCglobal Canada Inc.

49. http://www.spychips.com/what-is-rfid.html.

Acknowledgements

First of all, I would like to express my sincere gratitude to Prof. Kwangjo Kim, my academic advisor, 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. Cha and Prof. Lee for their generosity and agreeing to serve as advisory committee members.

I would also like to thank all members of Cryptology and Information Security Laboratory: Hyunrok Lee, Zeen Kim, Kyusuk Han, SungChul Heo, Sungjune Yoon, Jangseong Kim, Sungbae Ji, Vo Duc Lim, Dang Nguyen Duc, Konidala Munirathnam Divyan and Le Quy Loc, for giving me lots of interests and good advices during the course of my study. I also thank Hyunkyung Park for helpful support as a staff member. I also appreciate to the graduates: Sangshin Lee, Jaemin Park for their everlasting guidance in life and study of ICU. I also give my special gratitude for the aids and advices on every aspect of my research to Tomoyuki Asano who is in Sony, Japan.

Most of all, I would like to express my heartfelt thanks to my parents for their endless concerns and devotional affection. Without their prayers, faiths, and supports to me, I could never complete my study and have a good time in ICU.

Finally, I'll never forget the time in ICU. I could study my interesting field, information security, and experience many research projects.

Curriculum Vitae

Name : Youngjoon Seo

Date of Birth : Oct. 20. 1981

Sex : Male

Nationality : Korean

Education

2000.3–2005.2 Electronic and Electric Information Computer Engineering
Pusan National University (B.S.)

2005.2–2007.2 Cryptography and Information Security, Engineering
Information and Communications University (M.S.)

Career

2005.03–2006.02 Graduate Research Assistant
A Study on the Security for Special Digital Signature
Security Research Center(SERC), Hannam University

2005.03–2006.02 Graduate Research Assistant
A Study on the Security of RFID
Security Research Center(SERC), Hannam University

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

2006.03–2006.12 Graduate Research Assistant
Research on Link Layer Security
Electronics and Telecommunications Research Institute(ETRI)

2006.03–2006.12 Graduate Research Assistant
Development of Sensor Tag and Sensor Node Technologies for RFID/USN
Electronics and Telecommunications Research Institute(ETRI)

2006.06–2006.12 Graduate Research Assistant
Research on Security Standardization in RFID/USN
Electronics and Telecommunications Research Institute(ETRI)

2006.08–2006.12 Graduate Research Assistant
Research on Platform Security
Electronics and Telecommunications Research Institute(ETRI)

2006.07–2006.12 Graduate Research Assistant
Samsung-ICU Research Center
Samsung Electronics

2006.06–2006.08 Apprentice Researcher
Information Technologies Laboratories, Secure System Group,
Sony, Japan.

2006.03–2006.12 Teaching Assistant
Institute for IT-gifted Youth

Publications

(1) 2005.10 박재민, Dang Nguyen Duc, Vo Duc Liem, 서영준, 김광조, "2 세대 EPCglobal RFID 규격의 보안 취약성 검토 및 개선 방안 연구", 2005년도 한국정보보호학회 충청지부학술대회, 나사렛대학교, 천안.

(2) 2006.08 Youngjoon Seo, Hyunrok Lee and Kwangjo Kim, "A Scalable and Untraceable Authentication Protocol for RFID", *The Second International Workshop on Security Ubiquitous Computing Systems – secubiq 2006, LNCS 4097*, pp.252-261, Aug. 2006, Springer-Verlag, Seoul, Korea.

(3) 2006.09 한규석, 서영준, 윤성준, 김광조, "Enhancing Security for Vertical Handoff in SARAH under the Heterogeneous Networks ",2006년도 정보보호학술발표회논문집, pp. 159-166, 2006.9.29-30, 목원대학교, 대전.

(4) 2006.12 서영준, 이현록, 김광조, "A Lightweight Authentication Protocol Based on Universal Re-encryption of RFID tags", 2006년도 한국정보보호학회 동계학술대회, pp. 207-212, 2006.12.09, 세종대학교, 서울. Best Paper Award.

(5) 2006.12 이현록, 서영준, 김광조, "SLRRP 기반의 안전한 RFID 리더 프로토콜 연구", 2006년도 한국정보보호학회 동계학술대회, pp. 301-305, 2006.12.09, 세종대학교, 서울.

(6) 2007.01 Youngjoon Seo, Tomoyuki Asano, Hyunrok Lee and Kwangjo Kim, "A Lightweight Protocol Enabling Ownership Transfer and Granular Data Access of RFID Tags", *Symposium on Cryptography and Information Security – SCIS'07*, To be appeared, Huistenbosch, Japan.