저가형 태그를 위한 경량 및 초경량 RFID 인증 프로토콜

2011 이승광

석 사 학 위 논 문

저가형 태그를 위한 경량 및 초경량 RFID 인증 프로토콜

이 승 광 (李 承 桄)

전자컴퓨터공학부 (컴퓨터공학)

포항공과대학교 대학원

저가형 태그를 위한 경량 및 초경량 RFID 인증 프로토콜

Lightweight and Ultralightweight RFID Authentication Protocols for Low-cost Tags

Lightweight and Ultralightweight RFID Authentication Protocols for Low-cost Tags

by

Seungkwang Lee
Division of Electrical and Computer Engineering
Pohang University of Science & Technology

A thesis submitted to the faculty of Pohang University of Science & Technology in partial fulfillment of the requirements for the degree of Master in the Division of Electrical and Computer Engineering.

Pohang, Korea
December 2, 2010
Approved by

Major Advisor : Sung Je Hong

저가형 태그를 위한 경량 및 초경량 RFID 인증 프로토콜

이 승 광

위 논문은 포항공과대학교 대학원 석사 학위논문으로 학위논문 심사위원회를 통과하였음을 인정합니다.

2010년 12월 2일

학위논문심사 위원회 위원장 홍 성 제 (인)

위 원 김 종 (인)

위 원 김 장 우 (인)

MECE 20090290 이 승 광 Seungkwang Lee, Lightweight and Ultralightweight RFID Authentication Protocols for Low-cost Tags. 저가형 태그를 위한 경량 및 초경량 RFID 인증 프로토콜, Division of Eletrical and Computer Engineering, 2010, 44P, Advisor: Prof. Sung Je Hong. Text in English.

ABSTRACT

Counterfeiting is emerging as a serious threat to low-cost Radio Frequency Identification (RFID) tags. In addition, these RFID tags have engendered controversies on privacy due to their capability to provide unique identification. To solve theses problems, sophisticated tags can engage in authentication protocols using standard cryptographic algorithms. However, low-cost tags lack resources to implement these standard algorithms. So far, many studies have focused on implementing secure authentication protocols for low-cost tags.

The protocols for low-cost tags can be classified into two classes: lightweight and ultralightweight. The lightweight protocols require a random number generator and simple functions such as Cyclic Redundancy Checksum code but not a hash function while the ultralightweight protocols involve only bitwise operations on the tag-side. In the lightweight protocol class, HB^+ is computationally efficient but is vulnerable to a simple man-in-the-middle (MITM) attack, called the GRS attack. Later, $HB^\#$ improves HB^+ over the GRS attack. While HB^+ is a multi-round protocol, where each round consists of three passes, $HB^\#$ is a single-round protocol consisting of three passes, and thus reduces communication costs between the tag and the reader. But $HB^\#$ requires relatively large size of memory in hundreds of thousands bits for two shared secret matrices. More importantly, this protocol is also known to be vulnerable to a new type of MITM attack, called the OOV

attack. In the ultralightweight protocol class, the Gossamer protocol, the most recently published protocol, involves too heavyweight operations including modular additions and modulo operations with modulus 96.

In this thesis, we propose HB-SK, which is an improved version of HB^+. HB-SK is shown to be resistant to the GRS attack and also more lightweight than HB^+. Next, we propose $HB^\#$-SK, which is an improved version of $HB^\#$. $HB^\#$-SK is shown to be resistant to the OOV attack. Finally, we also propose an ultralightweight RFID authentication protocol, called UFO. UFO is shown to be more lightweight than the conventional Gossamer protocol.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Radio Frequency Identification

Radio Frequency Identification (RFID) is the use of an RFID tag incorporated into an object for identification without physical contact using radio signals. Typically, an RFID system consists of three main components: tag, reader, and back-end database. The tag carries object-identifying data and uses a radio signal to communicate with the reader. The reader queries the tag and can read the data supplied by the tag. Upon receiving a response from the tag, the reader forwards that response to the back-end database to retrieve detailed information that matches with the response from the tag.

The RFID system has distinct benefits over other automatic identification technologies. First, RFID does not require line-of-sight to operate, and thus the tags are readable without precise positioning. Second, each tag has unique identifying information that distinguishes it from many millions of other tags. This identifying information can act as a pointer to the back-end database containing detailed in-

formation for particular item. Finally, the tags have memory read/write capability. This feature is tremendously useful for IT systems and for security purposes.

1.2 Applications

There are a number of RFID applications. For example, RFID can be used to authenticate the pass-holder before permitting access to ski areas, concerts, and amusement parks, where tagged tickets are used. Also, many automobile models already use RFID tags in their keys to authenticate the owner, and in highway RFID allows drivers to pass through tolls without having to stop to pay a toll.

Throughout every stages of the supply chain, RFID can be used to provide visibility on the flow of goods, and thus the retail industry can track the location of products as they make their way from manufacturer, to a warehouse or distribution centers, and finally to retailer. In addition, RFID enhances the accuracy of information about where products are in the supply chain, and enables retailers to make decisions on what they need and what they do not need in stock. Based on the fact that RFID does not require line-of-sight to operate, RFID tags can be read much faster thereby goods move faster through the supply chain.

Not only these applications, one emerging application of RFID is health care, where RFID devices can be used to track the medical equipment and people inside the hospital. In the similar context, RFID can also enhance the quality of elder care by monitoring the owner's daily habits.

1.3 Privacy and Security

RFID gives rise to privacy concerns for users: tracking and inventorying. Upon receiving reader's interrogation, tags emit fixed serial numbers or static answers without alerting the owners. Thus, collaborating readers can track the location of

the specific target tag, and as a result, the owner's location can be also traced.

In the case that the tag's serial number is combined with personal information, the threat to privacy becomes more problematic. For example, when a consumer pays by a credit card for some goods, a seller can make a link between the tag's serial number and the consumer's identity. Then networked RFID readers can identify and profile the consumer. Furthermore, certain tags, Electronic Product Code (EPC) tags in particular, contain a field for the manufacturer of the object and a product code. Thus, the owner of EPC tags is subject to inventorying; the reader can collect information about what objects the owner possesses.

In addition to privacy, counterfeiting is emerging as an equally significant problem of RFID. When a tag is compromised so that the internal information of the tag is revealed, unauthorized manufacturers can produce a counterfeit tag and attach the tag to forged items. This counterfeiting results in undermining the concept of brands and deterring producers of reputable products from investing within national economy.

Last but not least, to guarantee availability of the RFID system is also important. In other words, the RFID system should be accessible and usable upon demand by an honest tag. Therefore, it is imperative to design secure authentication protocols in order to solve all these challenges on RFID.

1.4 Authentication Protocols for Low-cost RFID Tags

RFID authentication protocols can be classified into four classes: full-fledged, simple, lightweight, and ultralightweight [1]. The "full-fledged" protocols support conventional cryptographic functions such as symmetric encryption, public key encryption or cryptographic one-way function on the tag-side. The "simple" protocols support a random number generator and one-way hashing function on the tag-side. The "lightweight" protocols require a random number generator and simple functions

such as Cyclic Redundancy Checksum code but not a hash function. The "ultra-lightweight" protocols require logical bitwise operations on the tag-side.

From the retailer's perspective, low-cost tags are inevitably preferred to reduce manufacturing costs. Low-cost tags possess at most a couple of thousand gates, and only on the order of hundreds gates remain for security functionality. Thus, among those four classes, the first two classes, "full-fledged" and "simple", require excessive resources for implementation. The limited resources in the low-cost tags has posed intriguing research challenges and many lightweight and ultralightweight RFID authentication protocols have been proposed. In comparison to the ultralightweight protocols, the lightweight protocols have two advantages. First, the tag can generate random numbers and use them in the protocol. This is helpful to disrupt an attacker interrogating the tag with non-random challenges to extract secret information. Second, the tag and the reader do not need to update shared secret information after successful sessions. In the case of ultralightweight protocols, the tag and the reader always update their shared information to prevent tracking; otherwise the tag can be tracked by querying with the same challenges and observing the responses. When the back-end database is distributed, updating their shared information can cause coherence and synchronization problems on the shared information. In contrast, the ultralightweight protocols require lower implementation cost than the lightweight protocols.

HB^+ [2] is a lightweight protocol. This protocol is computationally efficient and presents concrete security proof based on the hardness of Learning Parity with Noise (LPN) problem. However, this protocol is known to be vulnerable to a simple man-in-the-middle (MITM) attack, the GRS attack [3, 4]. Later, $HB^{\#}$ [5, 6] improves the HB^+ protocol over the GRS attack based on the hardness of a matrix-based extension of the LPN problem, which is known as the Matrix-based HB (MHB) puzzle. Compared to the HB^+ protocol, this protocol reduces communication costs between the reader and the tag, but requires relatively large size of memory for two

shared secret matrices. Unfortunately, this protocol is vulnerable to a new variant of the MITM attack, called the OOV attack [7].

The Gossamer protocol [8], the most recently published ultralightweight protocol, has weak points in the aspect of computational cost. The ultralightweight protocols are expected to perform only simple bitwise operations like AND, AND, or XOR. However, the Gossamer protocol requires too heavyweight operations such as modular additions and modulo operations with modulus 96, not powers of 2.

1.5 Research Motivation and Goal

It is imperative to build a secure RFID authentication protocol to guarantee security and to protect owner's privacy. From the retailer's perspective, the low-cost tags are preferred to reduce manufacturing costs. Because the low-cost tags lack the resources to perform standard cryptographic operations, we need to design RFID authentication protocols for low-cost tags in more lightweight ways. This thesis focuses on the lightweight and ultralightweight RFID authentication protocols for low-cost tags. Previously, HB^+ and $HB^\#$ protocols, which are lightweight protocols, have vulnerabilities to the GRS attack and the OOV attack, respectively. Also, Gossamer, which is an ultralightweight protocol, requires too heavy operations compared to other ultralightweight protocols.

Three main topics of this thesis are as follows.

- We propose a modified HB^+ protocol to prevent the GRS attack and analyze the security of the proposed scheme. Our modification also reduces the transmission cost of HB^+.
- We propose a modified $HB^\#$ protocol to prevent the OOV attack with two additional secret vectors and two more XOR operations. To prove the security of the proposed scheme, we show that our scheme is secure against the OOV

attack and the attacks what HB$^{\#}$ prevented.

- We propose a new ultralightweight RFID authentication protocol which requires lower communication costs, lower computational costs, fewer memory requirements than Gossamer does. We also analyze the security and performance of the proposed scheme.

1.6 Organization

This thesis is organized as follows. Chapter 2 explains the HB^{+} protocol and its vulnerability on the MITM attack, called the GRS attack, and describes an improved version of HB^{+} which prevents the GRS attack. In Chapter 3, we review the HB$^{\#}$ protocol and explains the MITM attack, called the OOV attack, against the the HB$^{\#}$ protocol. Then we propose a modified HB$^{\#}$ protocol which prevents the OOV attack. In Chapter 4, we propose a new ultralightweight RFID authentication protocol involving only bitwise operations. In Chapter 5, we conclude this thesis and provide future work.

Chapter 2

HB-SK: Enhancing the Security and Efficiency of HB$^+$

2.1 Review of HB and HB$^+$

We begin by reviewing HB [9]. The tag and the reader share a k-bit secret x, and the tag would like to authenticate itself to the reader. The reader first selects a random challenge $a \in \{0,1\}^k$ and sends it to the tag. Upon receiving a, the tag computes the binary inner product $a \cdot x$. Since an eavesdropper capturing $O(k)$ valid challenge-response pairs between the tag and the reader can calculate x through Gaussian elimination, the tag intentionally sends the wrong response to the reader by injecting noise into its response $a \cdot x$ with constant probability $\eta \in (0, \frac{1}{2})$ [2]. HB is a multi-round protocol; by repeating for r rounds, the reader authenticates the tag if fewer than ηr of the tag's responses are incorrect. Figure 2.1 shows one round of HB. The security of this protocol is based on the hardness of the Learning Parity with Noise (LPN) problem [2]. This problem involves finding a vector x' such that

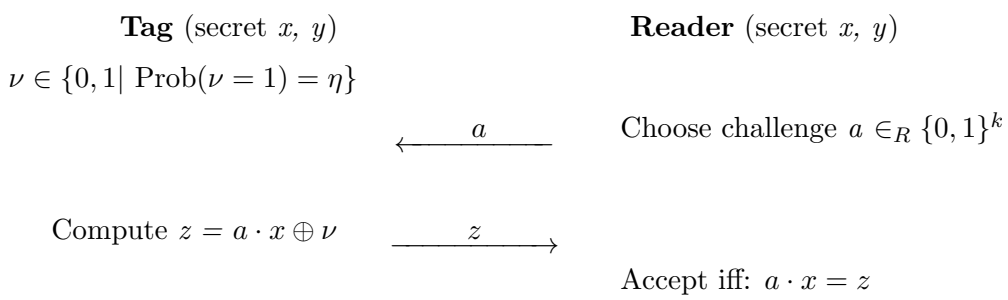


Figure 2.1: One round of HB.

$\mathbf{wt}(A \cdot x') \leq \eta q$. Formally, it is as follows:

Definition 1 (LPN problem) *Let A be a random $(q \times k)$- binary matrix, let x be a random k-bit vector, let $\eta \in (0, \frac{1}{2})$ be a noise parameter, and let ν be a random q-bit vector such that $\boldsymbol{wt}(\nu) \leq \eta q$, where $\boldsymbol{wt}$ indicates the Hamming weight. Given A, η,* and $z = A \cdot x \oplus \nu$, *find a k-bit vector x' such that $\boldsymbol{wt}(A \cdot x' \oplus z) \leq \eta q$.*

However, the HB protocol is only secure against a passive eavesdropper; it is not secure against an active attacker who has an ability to query the tags. If the same challenge a is repeatedly sent to the tag for $\Omega((1-2\eta)^{-2})$ times, the attacker can learn the error-free value of $a \cdot x$ with a overwhelming probability. Provided that $\Omega(k)$ error-free values are given, the attacker can compute x through Gaussian elimination [2].

Juels and Weis modified HB for the active attacks, and refer to it as HB$^+$ [2]. HB$^+$ is also based on the hardness of LPN problem. This modification introduces an additional secret y shared between the tag and the reader. HB$^+$ is also a multi-round authentication protocol, and each round described in Figure 2.2 is repeated r times. The reader accepts the round if $z = (a \cdot x) \oplus (b \cdot y)$. After r rounds, the reader accepts the tag if the tag's response is incorrect in less than ηr rounds.

The active attacker who defeated HB cannot extract x or y with non-random a challenges because the value $(b \cdot y) \oplus \nu$ effectively blinds the value $a \cdot x$.

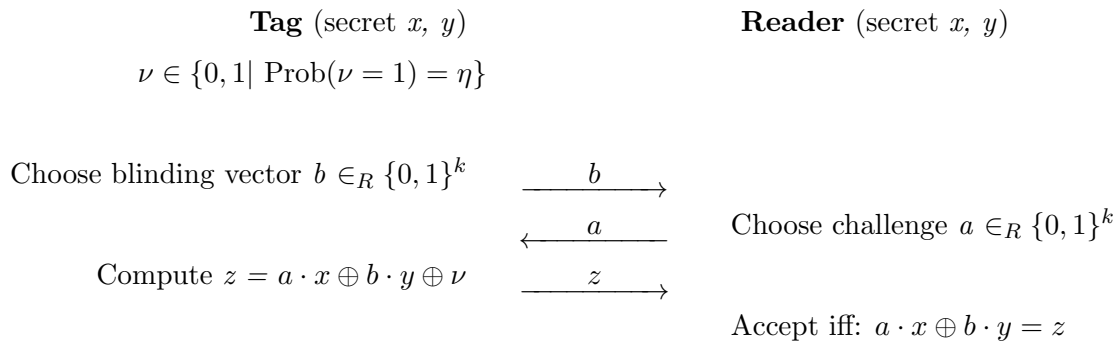


Figure 2.2: One round of HB$^+$.

2.2 The GRS attack to HB$^+$

Although Juels and Weis insisted that HB$^+$ is secure against the active attacks, Gilbert *et al.* introduced a MITM attack to the protocol, called the GRS attack. Figure 2.3 [3] describes the GRS attack on one round of HB$^+$. The attacker first chooses a constant k-bit vector δ and uses it to perturb the challenge a which is sent to the tag; the same δ is XORed to each authentication challenge for all r rounds of the protocol. If the reader finally accepts the tag, the attacker concludes that $\delta \cdot x = 0$ with overwhelming probability; otherwise $\delta \cdot x = 1$ with overwhelming probability. For example, if $\delta = 000...01$, and the reader accepts the tag, it means that $\delta \cdot x = 0$ with overwhelming probability and the LSB of x is 0. In this way, the attacker retrieves the k-bit secret x by repeating the full protocol k times for linearly independent δ's. After deriving x, the attacker can apply a similar process on b to recover the secret y.

2.3 HB-SK: Proposed Protocol

In this section, we propose HB-SK which is a modified HB$^+$ for the MITM attacks including the GRS attack. In HB-SK, the reader first sends a random challenge to

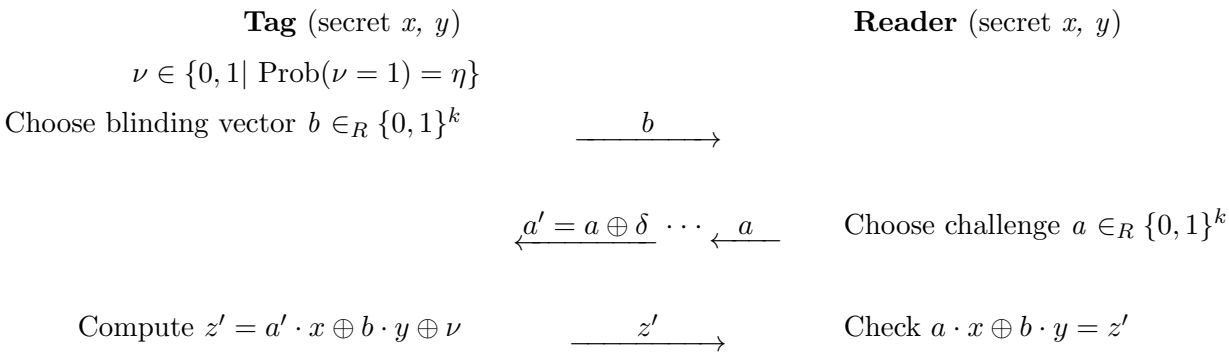


Figure 2.3: The GRS attack on one round of HB$^+$.

the tag which then responses with the final message like HB. As a result, HB-SK reduces the transmission and computational cost of HB$^+$. HB-SK is also a multi-round symmetric authentication protocol in which the tag and the reader still share two k-bit secret vectors x and y.

The reader first sends a random k-bit challenge a to the tag. Then the tag computes $z = (a \cdot x) \oplus (\bar{a} \cdot y) \oplus \nu$, and sends the response z to the reader. The reader accepts the round if $z = (a \cdot x) \oplus (\bar{a} \cdot y)$. Like the case in HB$^+$, the reader authenticates the tag after r rounds if the tag's response is incorrect in less than ηr rounds. Figure 2.4 shows one round of HB-SK.

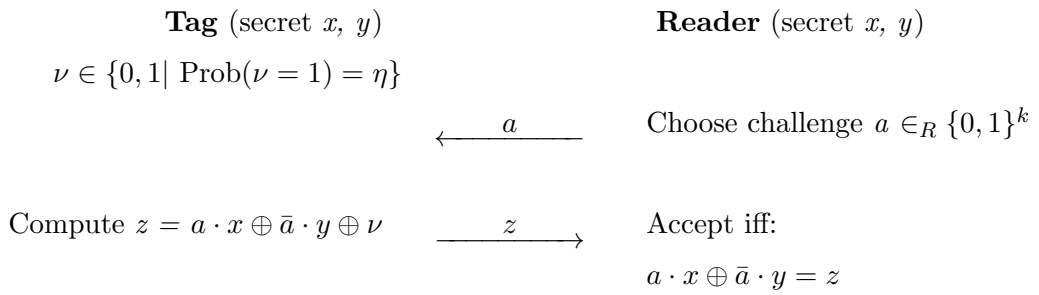


Figure 2.4: One round of HB-SK.

Compared to HB$^+$, logical negation is additionally used, but the tag does not

generate k-bit blinding vectors. Thus, the computational cost on the tag-side are lower than that of HB$^+$. Also, HB-SK requires only $\frac{k+1}{2k+1}$ of the transmission cost of HB$^+$ because the tag does not send the blinding vectors.

2.4 Security Analysis

Let's suppose the attacker launches the GRS attack on HB-SK by XORing δ to a. Because $a' = a \oplus \delta$, upon receiving a', the tag computes

$$\begin{aligned} z' &= (a' \cdot x) \oplus (\overline{a'} \cdot y) \oplus \nu \\ &= [(a \oplus \delta) \cdot x] \oplus [(\overline{a \oplus \delta}) \cdot y] \oplus \nu. \end{aligned}$$

If the the authentication process is successful, it means that $(\delta \cdot x) \oplus [(\overline{a \oplus \delta}) \cdot y] \oplus (\bar{a} \cdot y) = 0$ with overwhelming probability; otherwise $(\delta \cdot x) \oplus [(\overline{a \oplus \delta}) \cdot y] \oplus (\bar{a} \cdot y) = 1$. Let $\odot$ denote the bitwise XNOR operator under which 1 is the identity element. Then we have:

$$\begin{aligned} (\delta \cdot x) \oplus [(\overline{a \oplus \delta}) \cdot y] \oplus (\bar{a} \cdot y) &= (\delta \cdot x) \oplus (\overline{a \oplus \delta} \oplus \overline{a}) \cdot y \\ &= (\delta \cdot x) \oplus (a \odot \delta \oplus \overline{a}) \cdot y \\ &= (\delta \cdot x) \oplus (\delta \cdot y) \\ &= \delta \cdot (x \oplus y). \end{aligned}$$

Let $g(\delta) = \delta \cdot (x \oplus y)$. Then, if the authentication process is successful, this finally means that $g(\delta) = 0$. For $i \leq k$, let δ_i denote a constant vector in which only the i^{th} bit is 1; for example, $\delta_1 = 000 \cdots 001$. If the attacker mounts the attack with δ_i, and the authentication process is successful, this only implies $g(\delta_i) = 0$; in other words, the i^{th} bits of x and y are the same. However, this fact does not help the attacker to recover x and y because there are two possible cases to satisfy $\delta_i \cdot (x \oplus y) = 0$; 1) the i^{th} bits of x and y are all 0, 2) the i^{th} bits of x and y are all 1. In contrast, if the authentication process is unsuccessful for the attack with δ_i, this just implies

the i^{th} bits of x and y are different. This fact also does not reveal whether the i^{th} bits of x and y are 0 or 1.

When δ has the Hamming weight d, where $d > 1$, δ can be expressed as follows. For d integers $(\overbrace{i, j, \cdots, l}^{d}) \leq k$, where each integer indicates the position of 1,

$$\delta = \delta_i \oplus \delta_j \oplus \cdots \oplus \delta_l.$$

Plugging this into g gives:

$$\begin{aligned} g(\delta) &= g(\delta_i \oplus \delta_j \oplus \cdots \oplus \delta_l) \\ &= (\delta_i \oplus \delta_j \oplus \cdots \oplus \delta_l) \cdot (x \oplus y) \\ &= [\delta_i \cdot (x \oplus y)] \oplus [\delta_j \cdot (x \oplus y)] \oplus \cdots \oplus [\delta_l \cdot (x \oplus y)] \\ &= g(\delta_i) \oplus g(\delta_j) \oplus \cdots \oplus g(\delta_l). \end{aligned}$$

Thus, the attacker can know if the authentication process will be successful with δ without having to mount the attack, provided that $g(\delta_i)$, $g(\delta_j), \cdots, g(\delta_l)$ are known. However, this is only a linear combination of knowledge of $g(\delta_i)$, $g(\delta_j), \cdots, g(\delta_l)$, but does not provide any useful information on x and y. Therefore, the attacker cannot obtain any useful information on x and y, and as a result, HB-SK can simply prevent the GRS attack.

Notice that the OOV attack [7], which is a new type of the MITM attack against the HB$^{\#}$ protocol, can not be applied to HB-SK because the algorithm approximating the Hamming weight of noise vector distributed in the tag's final response does not work in HB-SK.

Chapter 3

HB$^{\#}$-SK: A Modified HB$^{\#}$ Protocol against the OOV Attack

3.1 Review of HB$^{\#}$

To increase the security of HB^{+}, Gilbert *et al.* [5, 6] proposed a new variant of HB^{+} named RANDOM-HB$^{\#}$ and its optimized version HB$^{\#}$, based on a matrix-based extension of the Learning Parity with Noise (LPN) problem, which is known as the *Matrix-based HB (MHB) puzzle* [5, 6]. These protocols reduced the complexity of the tag-reader communication in HB^{+}, but required relatively large size of memory, even hundreds of thousands bits, to provide 80-bit security.

Definition 2 (MHB puzzle) *Given q noisy samples $(a_i, a_i \cdot X \oplus \nu_i)$, where X is a secret $(k \times m)$-matrix and the a_i is a random k-bit vector, and a random challenge a, guess $a \cdot X$.*

Definition 3 *(DET-model). In the DET-model [2, 5], the attack is carried out in the following two phases:*

- *Phase 1: the attacker first interacts q times with the genuine tag, where q is the parameter used in the LPN problem (Definition 1).*
- *Phase 2: the attacker interacts with the reader and tries to impersonate the valid tag.*

Definition 4 *(GRS-MIM-model). In the GRS-MIM-model [5], the attack is carried out in the following two phases:*

- *Phase 1: the attacker can eavesdrop on all messages between the legitimate reader and the genuine tag including the reader's decision of whether to accept or not. In addition, the attacker can manipulate messages from the reader to the tag for q executions of the protocol, where q is the parameter used in the LPN problem (Definition 1).*
- *Phase 2: the attacker interacts with the reader and tries to impersonate the valid tag.*

RANDOM-HB$^{\#}$ is known to be secure in the DET-model (*Definition 3*) and the GRS-MIM-model (*Definition 4*). Also, HB$^{\#}$ is conjectured to be secure in the same models under the conjecture that the Toeplitz variant of the *MHB puzzle* is hard.

In the HB$^{\#}$ protocols, the tag and the reader share $(k_x \times m)$- and $(k_y \times m)$-binary matrices X and Y. The difference between RANDOM-HB$^{\#}$ and HB$^{\#}$ is the structure of the secret matrices, X and Y: while RANDOM-HB$^{\#}$ requires $(k_x + k_y)m$ bits of storage, they can be replaced with Toeplitz matrices in HB$^{\#}$, thereby reducing the storage for the secret matrices to $k_x + k_y + 2m - 2$.

During the authentication session, the tag first sends a k_y-bit random challenge b to the reader, which then sends a k_x-bit random challenge a. Upon receiving a, the

Table 3.1: Practical parameters for the HB$^{\#}$ protocol.

Set	k_x	k_y	m	η	t	P_{FR}	P_{FA}
I	80	512	1164	0.25	405	2^{-45}	2^{-83}
II	80	512	441	0.125	113	2^{-45}	2^{-83}
III	80	512	256	0.125	48	0	2^{-81}

tag XORs an extra noise vector ν to $(aX \oplus bY)$, and then sends the final message $z = aX \oplus bY \oplus \nu$; each bit in ν can become 1 with a probability $\eta \in (0, \frac{1}{2})$. Finally, the reader accepts the tag if the Hamming weight of $(aX \oplus bY \oplus z) \leq t$, where $t \in [\eta m, \frac{1}{2}m]$. Besides generating the noise vector ν and the random challenge b, the operations needed on the tag-side are two inner products using bitwise AND and XOR operations and two extra XOR operations. Figure 3.1 shows HB$^{\#}$. The proposed practical parameter sets are summarized in Table 3.1. P_{FR} and P_{FA} indicate the false rejection and false acceptance rates, respectively.

Tag (secret X, Y)		**Reader** (secret X, Y)
Choose $b \in_R \{0,1\}^{k_y}$	$\xrightarrow{\quad b \quad}$	
	$\xleftarrow{\quad a \quad}$	Choose $a \in_R \{0,1\}^{k_x}$
Choose $\nu \in_R \{0,1\}^m$		
s.t. $\Pr[\nu_i = 1] = \eta$		
Compute $z = aX \oplus bY \oplus \nu$	$\xrightarrow{\quad z \quad}$	Accept iff:
		$\mathbf{wt}(aX \oplus bY \oplus z) \leq t$

Figure 3.1: HB$^{\#}$.

3.2 The OOV Attack against HB$^{\#}$

HB$^{\#}$ is vulnerable to the OOV attack which is a new variant of MITM attack discovered by Ouafi, Overbeck, and Vaudenay [7]. During a successful authentication session, the attacker first looks at the values of a, b, and z, say $\bar{a}$, $\bar{b}$, and $\bar{z}$, respectively. The rest of the attack consists of two main steps. Algorithm 1 approximates the Hamming weight of the noise vector distributed in $\bar{z}$, and Algorithm 2 yields linear equations for X and Y. We briefly explain how the attack succeeds in extracting the secret matrices X and Y.

Given $\bar{a}$, $\bar{b}$ and $\bar{z}$, Algorithm 1 computes the Hamming weight of the noise vector $\bar{\nu} = \bar{a}X \oplus \bar{b}Y \oplus \bar{z}$ denoted $\bar{\omega} = \mathbf{wt}(\bar{\nu})$. At each iteration of Algorithm 1, the attacker intercepts on-going messages, a, b, and z, then sets $\hat{a} = a \oplus \bar{a}$, $\hat{b} = b \oplus \bar{b}$, and $\hat{z} = z \oplus \bar{z}$. Then we have:

$$aX \oplus \hat{b}Y \oplus \hat{z} = aX \oplus (b \oplus \bar{b})Y \oplus (z \oplus \bar{z}) = \nu \oplus \bar{\nu}.$$

Based on this fact, upon receiving the final message $\hat{z}$, the reader accepts the tag if $\mathbf{wt}(\nu \oplus \bar{\nu}) \leq t$. The probability that a bit of $(\nu \oplus \bar{\nu})$ becomes 1 is as follows:

$$\Pr[(\nu \oplus \bar{\nu}_i) = 1] = \begin{cases} \eta & \text{if } \bar{\nu}_i = 0 \\ 1 - \eta & \text{if } \bar{\nu}_i = 1. \end{cases}$$

Because of the independence of all bits, the expected value and variance of $\mathbf{wt}(\nu \oplus \bar{\nu})$ are given by $(m - \bar{\omega})\eta + \bar{\omega}(1 - \eta)$ and $m\eta(1 - \eta)$, respectively. Let $P(\bar{\omega})$ be the probability that the reader accepts the tag even though the attacker has altered the authentication messages. By the definition of the standard normal cumulative distribution function Φ, we have that:

$$\mathrm{P}(\bar{\omega}) = \Pr[\mathbf{wt}(\nu \oplus \bar{\nu}) \leq t] = \Phi\left(\frac{t - (m - \bar{\omega})\eta - \bar{\omega}(1 - \eta)}{\sqrt{m\eta(1 - \eta)}}\right).$$

$P(\bar{\omega})$ is a function of only one variable $\bar{\omega}$; the others are different at each invocation. This fact implies that $P(\bar{\omega})$ follows some specific distribution that depends on $\bar{\omega}$.

Therefore, if n is large enough, we can approximate $P(\bar{\omega})$ by $c \cdot n^{-1}$, where c is the number of times the reader accepts the tag, and get $\bar{\omega}$ by inverting $P(\bar{\omega})$.

Algorithm 1 Approximating $\bar{\omega}$

Input: $\bar{a}, \bar{b}, \bar{z}, n$

Output: $P^{-1}(\frac{c}{n})$, an approximation of $\bar{\omega} = \mathbf{wt}(\bar{a}X \oplus \bar{b}Y \oplus \bar{z})$,
where $P(\bar{\omega}) = \Pr[\mathbf{wt}(\nu \oplus \bar{\nu}) \leq t\ ] = \Phi\left(\frac{t-(m-\bar{\omega})\eta-\bar{\omega}(1-\eta)}{\sqrt{m\eta(1-\eta)}}\right)$

1. Initialize $c \leftarrow 0$
2. for $i = 1$ to n
3. During a protocol, set $\hat{a} \leftarrow a \oplus \bar{a}, \hat{b} \leftarrow b \oplus \bar{b}$, and $\hat{z} \leftarrow z \oplus \bar{z}$
4. if (reader accepts) then
5. $c = c + 1$
6. endif

Algorithm 2 outputs linear equations for X and Y. After initializing m-bit vector $\bar{c}$ to the original value of $\bar{z}$, the attacker gets $\bar{\omega}$ by using Algorithm 1. For $i \leq m$, the attacker flips the bit i of $\bar{z}$ to get $\bar{z}'$, then calls Algorithm 1 with $\bar{z}'$ as a part of arguments to get $\bar{\omega}'$. If $\bar{\omega}' = \bar{\omega}$ - 1, the attacker flips the bit i of $\bar{c}$ to make it noise-free. At the end, the attacker gets a linear equation $\bar{a}X \oplus \bar{b}Y = \bar{z} \oplus \bar{\nu} = \bar{c}$, where $\bar{c}$ is the noise-free vector. Given such linear equations, the attacker can solve the equations and finally recover the secret matrices.

3.3 HB$^{\#}$-SK: Proposed Protocol

In this section, we propose HB$^{\#}$-SK, which is an improved version of HB$^{\#}$, in such a way to prevent the OOV attack. Compared to HB$^{\#}$, our modification introduces an additional m-bit secret vector s and a k_x-bit secret vector e which satisfies $s =$

Algorithm 2 Getting linear equations for X and Y

Input: $\bar{a}, \bar{b}, \bar{z}$ and $\bar{\omega}_{est}$ the expected weight of $\bar{\nu} = \bar{a}X \oplus \bar{b}Y \oplus \bar{z}$

Output: A linear equation $\bar{a}X \oplus \bar{b}Y = \bar{c}$

1. Initialize *m*-bit vector $\bar{c} \leftarrow \bar{z}$
2. Call Algorithm 1 on input $(\bar{a}, \bar{b}, \bar{z}, n)$ to get $\bar{\omega}$
3. for $i = 1$ to m
4. Flip bit i of $\bar{z}$ to get $\bar{z}'$
5. Call Algorithm 1 on input $(\bar{a}, \bar{b}, \bar{z}', n)$ to get $\bar{\omega}'$
6. if $(\bar{\omega}' = \bar{\omega}$ - 1) then
7. $\bar{c}_i = \bar{c}_i \oplus 1$
8. endif

$e \cdot X$. Since the practical size of the secret matrix X is more than tens of thousands in bits (Table 3.1), the tag and the reader share s rather than computing $e \cdot X$ during the protocol to reduce the computational costs. Unlike HB$^{\#}$, the reader generates a k_x-bit random challenge a' and sends a, where $a = a' \oplus e$. Because a' is a random challenge, a is also treated as a random challenge. To extract a' from a, the tag simply computes $a \oplus e$. HB$^{\#}$-SK is as follows.

HB$^{\#}$-SK The tag and the reader share secret matrices X, Y, an m-bit secret vector s, and a k_x-bit secret vector e which satisfies $s = e \cdot X$. Given $\eta \in (0, \frac{1}{2})$, $t \in [m\eta, \frac{m}{2}]$.

1. The tag sends a k_y-bit random challenge b to the reader.
2. The reader chooses a k_x-bit random challenge a'.
3. The reader sends $a = a' \oplus e$ to the tag.
4. The tag extracts the random challenge a', $a' = a \oplus e$.

5. The tag computes $a' \cdot X$.
6. The tag generates an extra noise vector ν,
   computes the final message $z = a'X \oplus s \oplus bY \oplus \nu$ $(= a'X \oplus eX \oplus bY \oplus \nu$ $= aX \oplus bY \oplus \nu)$, and sends z to the reader.
7. The reader accepts the tag if the Hamming weight of
   $a'X \oplus s \oplus bY \oplus z \leq t$.

According to the definition of the LPN problem (*Definition 1*), coefficients in inner product should be given. It is important to notice that upon receiving a, the tag finally sends $z = aX \oplus bY \oplus \nu$ to the reader so that HB$^{\#}$-SK maintains consistency in the LPN problem.

Besides generating the noise vector ν and the random challenge b, the operations performed by the tag to authenticate itself to the reader are two inner products involving simple bitwise AND and XOR along with four XOR operations.

Tag (secret X, Y, s, e)		**Reader** (secret X, Y, s, e)
Choose $b \in_R \{0,1\}^{k_y}$	$\xrightarrow{\quad b \quad}$	
		Choose $a' \in_R \{0,1\}^{k_x}$
	$\xleftarrow{\quad a \quad}$	$a = a' \oplus e$
Extract a', $a' = a \oplus e$		
Choose $\nu \in_R \{0,1\}^m$ s.t. $\Pr[\nu_i = 1] = \eta$		
Compute $z = a'X \oplus s \oplus bY \oplus \nu$ $(= a'X \oplus eX \oplus bY \oplus \nu = aX \oplus bY \oplus \nu)$	$\xrightarrow{\quad z \quad}$	Accept iff: $\mathbf{wt}(a'X \oplus s \oplus bY \oplus z) \leq t$

Figure 3.2: HB$^{\#}$-SK.

3.4 Security Analysis

3.4.1 Security against the OOV attack

In HB$^{\#}$-SK, the tag extracts the random challenge a' from a sent by the reader. Suppose the attacker launches the OOV attack. Then, the reader accepts the tag if and only if

$$\begin{aligned} \mathbf{wt}(a'X \oplus s \oplus \hat{b}Y \oplus \hat{z}) &= \mathbf{wt}(a'X \oplus s \oplus bY \oplus \bar{b}Y \oplus z \oplus \bar{z}) \\ &= \mathbf{wt}(\nu \oplus \bar{\nu} \oplus s) \leq t. \end{aligned} \tag{3.1}$$

In this section, we show how the new secret vector s prevents the OOV attack in Equation (3.1) and recommend the Hamming weight of s to protect against the OOV attack.

The crucial observation in Equation (3.1) is that $\bar{\nu}$ and s are all fixed values. As defined previously, $\mathbf{wt}(\bar{\nu})$ is $\bar{\omega}$. Let γ be $\mathbf{wt}(s)$ and ρ be the expected value of $\mathbf{wt}(\bar{\nu} \oplus s)$. Let $i = \mathbf{wt}(\bar{\nu} \wedge s)$, then $\mathbf{wt}(\bar{\nu} \oplus s)$ is $\bar{\omega} + \gamma - 2i$. There are $\binom{\gamma}{i} \cdot \binom{m-\gamma}{\bar{\omega}-i}$ cases for $i = \mathbf{wt}(\bar{\nu} \wedge s)$. Thus, we have that:

$$\rho = \frac{\sum_{i=0}^{\bar{\omega}} \binom{\gamma}{i}\binom{m-\gamma}{\bar{\omega}-i}\{\bar{\omega} + \gamma - 2i\}}{\binom{m}{\bar{\omega}}}.$$

For $i > \gamma$, we assume $\binom{\gamma}{i} = 0$.

Theorem 1 *Given parameters in Table 3.1, HB$^{\#}$-SK can prevent the OOV attack when the Hamming weight of s is greater than $\frac{m}{2}$, where m is the bit-length of s.*

Proof : With the parameters in Table 3.1, we analyzed the tendency of ρ depending on γ and $\bar{\omega}$. Because $0 < \eta < \frac{1}{2}$, $\bar{\omega}$ was adjusted into four cases: $\bar{\omega}_{opt}$, $\frac{m}{4}$, $\frac{m}{3}$, and $\frac{m}{2}$.

Because of the independence of all bits, the expected value of $\mathbf{wt}(\nu \oplus \bar{\nu} \oplus s)$ is given by $\kappa = (m - \rho)\eta + \rho(1 - \eta)$. To achieve 2^{80}-security, the following condition must be satisfied [7].

$$1/P(\rho) = 1/\Phi\left(\frac{t - \kappa}{\sqrt{m\eta(1-\eta)}}\right) > 2^{80}.$$

Since Φ(-10.2) $\approx 2^{-80}$, ρ must satisfy that

$$\begin{aligned}
&\frac{t-\kappa}{\sqrt{m\eta(1-\eta)}} = \frac{t-(m-\rho)\eta-\rho(1-\eta)}{\sqrt{m\eta(1-\eta)}} < -10.2 \\
\iff\quad & (m-\rho)\eta+\rho(1-\eta) > 10.2\sqrt{m\eta(1-\eta)}+t \\
\iff\quad & -\rho\eta+\rho(1-\eta) > 10.2\sqrt{m\eta(1-\eta)}+t-m\eta \\
\iff\quad & \rho(1-2\eta) > 10.2\sqrt{m\eta(1-\eta)}+t-m\eta \\
\iff\quad & \rho > \tfrac{1}{1-2\eta}\left(10.2\sqrt{m\eta(1-\eta)}+t-m\eta\right).
\end{aligned}$$

From Table 3.1, $\frac{1}{1-2\eta}\left(10.2\sqrt{m\eta(1-\eta)}+t-m\eta\right)$ is approximately 529 for Set I, 172 for Set II, and 93 for Set III. Hence, if the Hamming weight of s is greater than $\frac{m}{2}$, HB$^{\#}$-SK can significantly disturb the OOV attack. Figure 3.3 shows ρ against γ with different $\bar{\omega}$. Regardless of the value of $\bar{\omega}$, in the case of Set I, $\rho > 529$ for $\gamma > 495$ ($\fallingdotseq 0.42m$), in the case of Set II, $\rho > 172$ for $\gamma > 147$ ($\fallingdotseq 0.34m$), and in the case of Set III in Table 3.1, $\rho > 93$ for $\gamma > 88$ ($\fallingdotseq 0.34m$). Therefore, with the new secret vector s with the Hamming weight greater than $\frac{m}{2}$, HB$^{\#}$-SK can prevent the OOV attack.

3.4.2 Security against the DET- and the GRS-MIM-model

In this section, we show that HB$^{\#}$-SK also provides the same level of security as HB$^{\#}$ against the DET-model (*Definition 3*) and the GRS-MIM-model (*Definition 4*). HB$^{\#}$-SK can be simplified as follows.

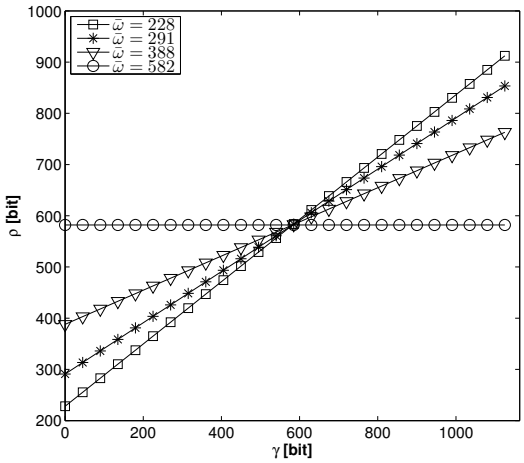
1. The tag and the reader send the blinding vector b and the challenge a, respectively. This is as follows :
   (a) The tag sends a k_y-bit blinding vector b to the reader.
   (b) The reader chooses a k_x-bit random challenge a' and sends $a = a' \oplus e$ to the tag.

(c) The tag extracts the random challenge a'.

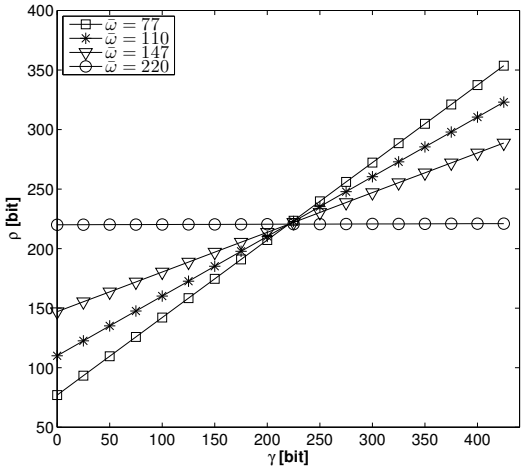
2. The tag computes $z = aX \oplus bY \oplus \nu$ $(= a'X \oplus s \oplus bY \oplus \nu)$, and sends z to the reader.

3. The reader accepts the tag if the Hamming weight of $aX \oplus bY \oplus z \leq t$.

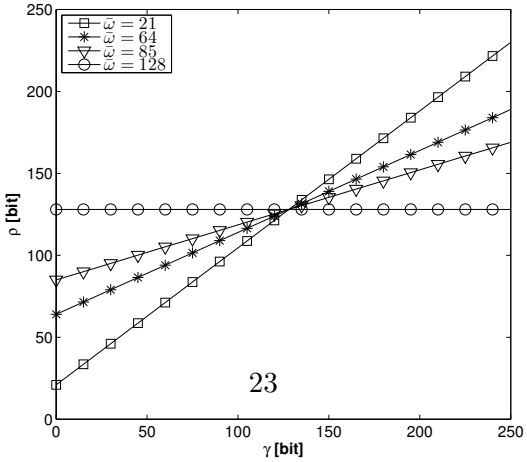
Thus, in HB$^{\#}$-SK, the tag and the reader send and receive the same type of messages in the same order as in HB$^{\#}$. This implies that the security proof of HB$^{\#}$ in the DET-model and the GRS-MIM-model can also be applied to HB$^{\#}$-SK. Therefore, HB$^{\#}$-SK still provides the same level of resistance in the DET-model and the GRS-MIM-model same as HB$^{\#}$.



(a) With Set I ($m = 1164$)



(b) With Set II ($m = 441$)



(c) With Set III ($m = 256$)

Figure 3.3: ρ against γ with different $\bar{\omega}$ ($\bar{\omega}_{opt}$, $m/4$, $m/3$, and $m/2$).

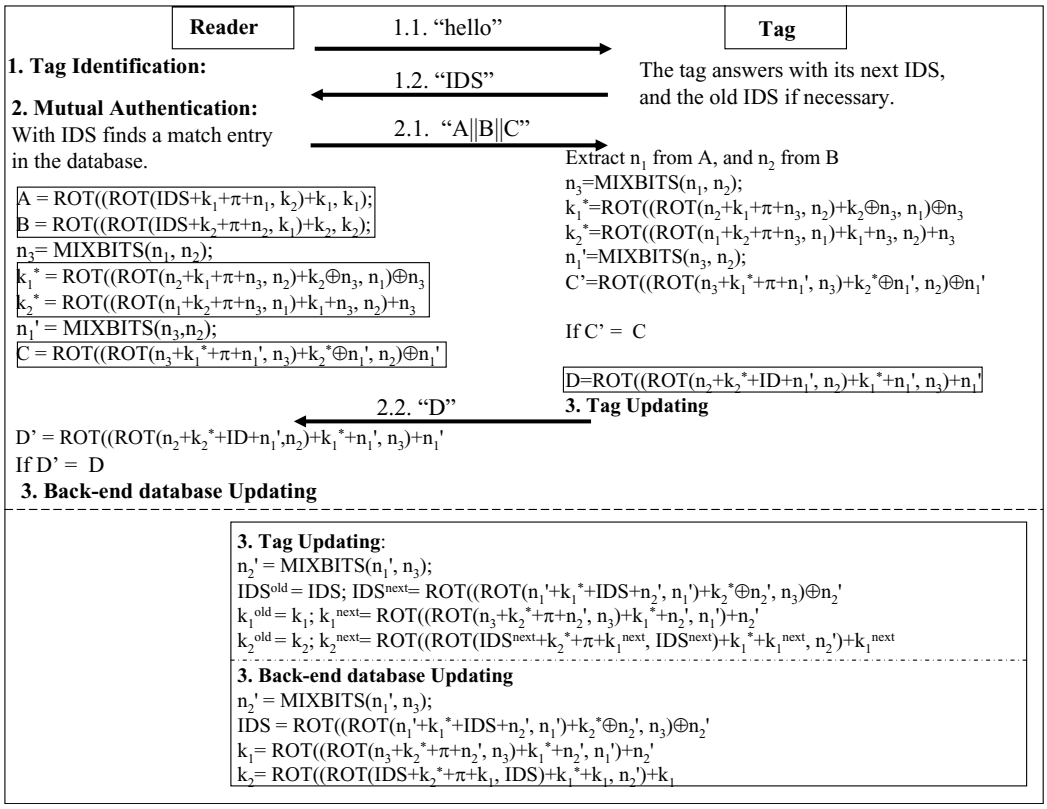
Chapter 4

UFO : A Secure and Ultralightweight RFID Authentication Protocol

4.1 The Gossamer Protocol and Its Weakness

Any security component is expected to produce different outputs for each different input. More importantly, even if two inputs are little different, the security component should produce totally different outputs for the two inputs; this property is called diffusion effect. The Gossamer protocol shown in Figure 4.1 [8], which is an ultralightweight RFID authentication protocol, mainly focuses on providing the diffusion effect because all its previous work's vulnerabilities were largely due to the absence of the diffusion effect [1, 10, 11, 12, 13, 14, 15, 16]. Notice that the modified Gossamer protocol [17] involves the same kinds of operations in the same order; the only difference is the order of applied variables to prevent the replay attacks.

However, these two protocols strongly depend on modular additions for the diffu-



† π = 0x3243F6A8885A308D313198A2 (L = 96 bits).

(a) The Gossamer protocol

```
Z = MixBits(X,Y)
----------------------------
Z = X;
for(i=0; i<32; i++) {
Z = (Z>>1) + Z + Z + Y  ;}
----------------------------
```

(b) MIXBITS

Figure 4.1: The Gossamer protocol

sion effect. Based on the fact that a full adder is composed of two XOR gates, two AND gates, and an OR gate, modular additions require not only more hardware resources but also more computational costs than bitwise operations, including XOR, AND, and OR, do. Even worse, for every left rotate operations, these protocols need modulo operations with modulus 96 to decide the rotate amount. In other words, in the Gossamer protocol, every $ROT(x,y)$ is defined to *left rotate* x with (y mod 96) bits. Because 96 is not powers of 2, those modulo operations inevitably involve multiplications, and thus could not be treated as a simple kind operations.

4.2 UFO: Proposed Protocol

In this section, we propose a new ultralightweight RFID authentication protocol involving only bitwise operations, named *UFO, Ultralightweight rFid authentication with Only bitwise*. Figure 4.2 shows UFO. Each tag shares a static identifier *ID* with the reader in advance. Also, the tag has a pseudonym *IDS* and a key K, which are updated every successful authentication session. The length of each *ID, IDS,* K is 96 bits. To prevent desynchronization, the tag is always keeping two entries of (*IDS,* K): one is for *old* values before update, the other is for *potential next* values after update.

A subscript indicates a 4-bit substring of a given vector so that each 96-bit vector is divided into 24 substrings each of 4-bit length. Thus, for any vector x,

$$x = x_{23}||\cdots||x_1||x_0.$$

$ROT(x,y)$ is defined to *left rotate* x with y bits.

UFO consists of three phases: the tag identification phase, the mutual authentication phase, and the pseudonym updating and key updating phase. At the mutual authentication, and the pseudonym updating and key updating phase, Algorithm $\boldsymbol{f}$ is used to provide the diffusion effect.

Given a 96-bit random operand x and a 4-bit operand y, ROT(x, y) is, as defined previously, to left-rotate x with y bits. Suppose that the attacker first chooses a constant 4-bit vector δ and uses it to perturb y, say $\bar{y} = y \oplus \delta$. Then say $r = \mathrm{ROT}(x, y)$ and $\bar{r} = \mathrm{ROT}(x, \bar{y})$. $x[\ell]$ denotes the ℓ^{th} bit of x for $\ell \in [0, 95]$. Because x is a random operand, we know that:

$$x \in_R \{\{0,1\}^{96} | \Pr[x[\ell] = 1] = \frac{1}{2} \text{ for } 0 \leq \ell \leq 95\}.$$

This observation implies that if $d = |y - \bar{y}|$,

$$\Pr[r[\ell] = \bar{r}[(\ell \pm d) \bmod 96\ ] = \frac{1}{2}.$$

Therefore,if different rotate amounts are applied to the same operand x, each bit of the result of the ROT operation will be changed with a $\frac{1}{2}$ probability. Algorithm $\boldsymbol{f}$ is 24-fold of this observation. The inputs and the output are all 96 bits. *amount*, which is 96-bit in length, is divided into 24 substrings, and each substring, $amount_i$, is used to decide the rotate amount for the i^{th} loop at line 3. Thus, if the integrity of input values, a and b, is broken, the output value c will be totally different.

The three phases of the protocol are as follows:

Tag identification. In every instance of the protocol, the reader initially sends *"hello"* to the tag, which then responds with its *potential next IDS*. If the reader finds a matched entry with that response, it steps into the mutual authentication phase; otherwise, it sends *"hello"* again, and the tag responses with the *old IDS*.

Mutual authentication. With the matched entry to either *old* or *potential next IDS*, and the randomly generated number n, the reader constructs A and computes B. The tag extracts n from A and verifies the value of B sent by the reader. If the verification succeeds, the tag computes C with its local values. Upon receiving C, the reader also verifies the value of C by comparing it with locally computed value.

Pseudonym updating and key updating. After the tag and the reader authenticate each other, they update their shared pseudonym and key values. To resist possible desynchronization attacks, the tag keeps the *old* entry.

The random number generation is required on the reader-side only, and the tag need operate only bitwise operations: bitwise XOR ($\oplus$), AND ($\wedge$) and left rotate ($ROT(x,y)$).

Algorithm 3 f

Input: $a,\ b$

Output: c

1. Initialize $amount \leftarrow a \oplus b$, $c \leftarrow a \wedge b$
2. for $i = 0$ to 23
3. b = ROT (b, $amount_i$)
4. c = c $\oplus$ b

4.3 Security and Performance Analysis

4.3.1 Security Analysis

The security of *UFO* mainly depends on the diffusion effect that Algorithm $\boldsymbol{f}$ provides.

Suppose the attacker modifies the j^{th} substring of the message A, that is A_j, among 24 substrings, and call this modified message $\bar{A}$. This means that n_j is changed, which in turn means that $amount_j$ in Algorithm $\boldsymbol{f}$ is also changed at the first invocation of Algorithm $\boldsymbol{f}$ on the tag-side. Specifically, let $\bar{n}$ denote the derived random number from $\bar{A}$. Note that at the first invocation of Algorithm $\boldsymbol{f}$ on the tag-side, $n = b$. This implies that $\bar{n}$ is delivered to Algorithm $\boldsymbol{f}$ as the second input,

say $\bar{b}$. Then we have $\overline{amount} = a \oplus \bar{b}$. Let $[b_i]$ denote the result of the ROT operation for the i^{th} loop at line 3. At the j^{th} loop, because $amount_j \neq \overline{amount}_j$, we have:

$$\Pr[[b_j] = [\overline{b_j}]] = \left(\frac{1}{2}\right)^{96},$$

and for $i \geq j$,

$$\Pr[[b_i][\ell] = [\overline{b_i}][\ell]] = \frac{1}{2}, \text{ for } 0 \leq \ell \leq 95.$$

Hence, for $i \geq j$, at the i^{th} loop, totally different operands $[b_i]$ and $[\bar{b}_i]$ may be applied to each loop at line 4, and thus the output of Algorithm $\boldsymbol{f}$ will be totally different. Consequently, if the attacker changes any bit of A, there will be a negligible probability that the output of Algorithm $\boldsymbol{f}$ remains the same. In addition, for any a and b, and for any $\bar{b} \neq b$,

$$\Pr[\text{the outputs of } f(a,b) \text{ and } f(a,\bar{b}) \text{ differ in } m \text{ bits}] = \binom{96}{m}\left(\frac{1}{2}\right)^{96}.$$

In average, half bits of the output of $f(a,\bar{b})$ will be different to the output of $f(a,b)$.

Based on all these facts, Algorithm $\boldsymbol{f}$ strongly provides security properties including strong authentication and strong integrity enough to withstand all possible attacks as follows.

Data confidentiality. All messages in transmission involve secret values shared only by the legitimate readers and the genuine tags; so, the eavesdroppers cannot obtain the static identifier and the secret values.

Tag anonymity and Resistance to tracking. The pseudonym of each tag, *IDS*, is updated every successful authentication session, and its update process involves a random number and the secret values. Thus, on successive queries from the reader, the tag responses with different *IDS*, and an attacker cannot track the tag. Of course, if the attacker queries the same tag repeatedly between two successful authentications, the tag will respond with the same *IDS*. However, this is not a practical scenario.

Mutual authentication and Data integrity. Only the legitimate reader who possesses the secret values can construct B from A, and only the genuine tag can correctly extract the random number n and then generate C. Because C involves the current/potential next secret keys, and the random number sent by the reader, our scheme can ensure the authenticity and the integrity of the messages.

Forward security. The forward security property means that the knowledge of a tag's internal state at time t cannot identify the past communications of the tag that occurred at a time $t' < t$. In UFO, even after compromising a tag and revealing two entries of *(ID, IDS, K)*, an attacker still cannot infer the previous secret values of that tag because the update process and every message construction involve a random number which is in turn protected by the secret value at that time. Thus, the attacker cannot jeopardize the past communication.

Resistance to replay attacks. To impersonate the tag, the attacker may replay the response C. However, the reader does not accept this response because the random number in the message A is different every authentication session. In another scenario, the attacker may pretend as if the message C of the last session was lost. Upon receiving *old IDS* from the tag, the attacker replays the eavesdropped values of $A\|B$ corresponding to *old IDS*. Even if this is successful, this scenario does not change or disclose any internal state of the genuine tag.

Resistance to man-in-the-middle attacks. UFO always provides the authenticity and the integrity of the messages; any modification on the message A will produce a significant change in B and C. Thus, the attacker can not modify the messages in transmission without being noticed. Resistance to disclosure attacks is also accomplished for the same reason.

Resistance to de-synchronization attacks. There are two ways to desynchronize the shared values between the tag and the reader. The first way is

to intercept the response C from the tag. The second one is to make the tag and the reader use different n to update the shared values. In the case of the first one, the tag and the reader still can authenticate each other because the tag always keeps two entries of (*IDS, K*): one is for *old* values before update, the other is for *potential next* values after update. The second case does not work in UFO because it is infeasible to modify the messages in transmission without being noticed.

4.3.2 Performance Analysis

This section analyzes the performance of UFO in terms of computational cost, transmission cost, and storage requirement.

Computational cost. Algorithm $\boldsymbol{f}$ accounts for most computations in UFO. For 96-bit operands, 25 XOR operations and 1 AND operations are operated to execute Algorithm $\boldsymbol{f}$. Rotate amount varies from case to case, ranging from 0 to 15 at each loop. To complete the *mutual authentication phase* and the *updating phase*, Algorithm $\boldsymbol{f}$ will be executed 4 times including extra 2 XOR operations. In total, 102 XOR operations and 4 AND operations are required in UFO. Note that there is no modulus operation involved in rotate operations because that rotate amount, which is determined by each 4-bit substring of *amount*, ranges from 0 to 15.

Storage requirement. Each tag shares its static identifier *ID*, two entries of (*IDS, K*) with the reader in advance. A ROM memory is required to store static values *ID* which are set to 96 bits. Besides, a 384-bit rewritable memory is required to store the two updatable entries (*IDS, K*). An additional storage to store a 96-bit nonce is also required.

Transmission cost. Initially, *Hello* and *IDS* messages are sent, and three additional messages, *A, B,* and *C*, come and go between the reader and the tag to accomplish mutual authentication. The *Hello* message accounts for 5 bytes, and the others do 96 bits each.

Table 4.1: A Simple Comparison with the Gossamer protocol ([*], * indicates the length of the operands in bits)

	UFO	Gossamer
Computational cost	102 XORs [96] 4 ANDs [96]	332 modular additions [96] 6 XORs [96] 18 (x mods 96) [96]
Storage requirement	576 bits	1248 bits
Transmission cost	328 bits	424 bits

Table 4.1 shows that UFO requires lower costs in terms of all three criteria, compared to the Gossamer protocol. Note that the author of the Gossamer protocol did not include the storage amount for π when evaluating the storage requirement for the Gossamer protocol [8]. The storage requirement for the Gossamer protocol in Table 4.1 is the sum of all secret values, π, and five nonces; all are 96 bits in length.

Reader		**Tag**
	$\xrightarrow{Hello}$	
Find the matched	$\xleftarrow{IDS}$	
entry from DB		
Choose n		
$A = ID \oplus K \oplus n$		
$B = f(K,\ n)$		
	$\xrightarrow{A\|\|B}$	extract n from A
		$\bar{B} = f(K,\ n)$
		if $\bar{B} = B$
		accept the reader
$\bar{K} = f(B \oplus K, n)$		$\bar{K} = f(B \oplus K, n)$
$\bar{C} = f(n \oplus \bar{K},\ B)$		$C = f(n \oplus \bar{K},\ B)$
	$\xleftarrow{C}$	
if $\bar{C} = C$		
accept the tag		
$\mathrm{IDS}_{next} = f(ID,\ n)$		$\mathrm{IDS}_{next} = f(ID,\ n)$
		$\mathrm{IDS}_{old} = \mathrm{IDS}$
$\mathrm{K}_{next} = \bar{\mathrm{K}}$		$\mathrm{K}_{next} = \bar{\mathrm{K}}$
		$\mathrm{K}_{old} = \mathrm{K}$

Figure 4.2: UFO.

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In this thesis, we proposed lightweight and ultralightweight RFID authentication protocols for low-cost tags. As lightweight protocols, we improved HB^+ and $HB^\#$ against the MITM attacks. Also, we proposed a new ultralightweight RFID authentication protocol involving only bitwise operations.

In Chapter 2, we described the GRS attack to HB^+. To prevent the GRS attack, we proposed an improved version of HB^+, named HB-SK and analyzed its security. Compared to HB^+, HB-SK requires $\frac{k+1}{2k+1}$ of the transmission cost of HB^+, and reduces the computational cost on the tag-side.

In Chapter 3, we described the OOV attack to $HB^\#$. We then proposed an modified version of $HB^\#$, named $HB^\#$-SK, in such a way to prevent the OOV attack. Furthermore, we showed that $HB^\#$-SK still provides the same level of resistance to the attack models in $HB^\#$. Compared to $HB^\#$, $HB^\#$-SK introduced two additional secret vectors and two more XOR operations.

In Chapter 4, we first explained why the Gossamer protocol is relatively heavy-weight compared to other ultralightweight RFID authentication protocols. The Gossamer protocol involves modular additions in order of hundreds times and modulo operations with modulus 96, not powers of 2, to provide diffusion effect. Modular additions require more resources to be implemented than logical bitwise operations do, and modulo operations with modulus not powers of 2 may involve multiplications. Thus, we proposed a new ultralightweight RFID authentication protocol, named UFO. UFO involves only bitwise operations. We proved its security, then showed that UFO requires lower computational cost, storage, and transmission cost compared to the Gossamer protocol.

5.2 Future Work

Some research directions have come up during this research. Those are as follows:

- As HB$^+$ can be parallelized [18], HB-SK, which is a $2r$-pass protocol, can be parallelized as a two pass protocol to reduce complexity of the tag-reader communication. In the parallel version of the protocol, the data transferred between the tag and the reader is packed into two passes of rk and r bits, respectively. However, sophisticated security proof on this parallelized version of the HB-SK protocol should be given.

- To reduce the memory requirement and the computational cost of our proposed protocol, HB$^{\#}$-SK, we can make use of a circulant matrix like HB-CM [19] rather than using the two random or Toeplitz matrices. The main challenge is to prove the security of adapting only one circulant matrix in the protocol.

- To provide more concrete security of UFO, we need to provide a formal proof that UFO is resistant to the attacks.

References

[1] H.-Y. Chien, "Sasi: A new ultralightweight rfid authentication protocol providing strong authentication and strong integrity," *IEEE Trans. Dependable Secur. Comput.*, vol. 4, no. 4, pp. 337–340, 2007.

[2] A. Juels and S. A. Weis, "Authenticating pervasive devices with human protocols," in *Advances in Cryptology - CRYPTO*, (Berlin, Heidelberg), pp. 293–308, Springer, 2005.

[3] H. Gilbert, M. Robshaw, and H. Sibert, "Active attack against hb+: a provably secure lightweight authentication protocol," *Electronics Letters*, vol. 41, no. 21, pp. 1169 – 1170, 2005.

[4] H. Gilbert, M. J. B. Robshaw, and Y. Seurin, "Good variants of HB^+ are hard to find," in *Financial Cryptography and Data Security, LNCS*, (Berlin, Heidelberg), pp. 156–170, Springer, 2008.

[5] H. Gilbert, M. J. B. Robshaw, and Y. Seurin, "$HB^\#$: increasing the security and efficiency of HB^+," in *EUROCRYPT'08: Proceedings of the theory and applications of cryptographic techniques 27th annual international conference on Advances in cryptology*, (Berlin, Heidelberg), pp. 361–378, Springer-Verlag, 2008.

[6] H. Gilbert, M. J. B. Robshaw, and Y. Seurin, "HB$^{\#}$: increasing the security and efficiency of HB^{+}, full version," in *Cryptology ePrint Archive, Report 2008/028*, 2008.

[7] K. Ouafi, R. Overbeck, and S. Vaudenay, "On the security of HB$^{\#}$ against a man-in-the-middle attack," in *ASIACRYPT '08: Proceedings of the 14th International Conference on the Theory and Application of Cryptology and Information Security*, (Berlin, Heidelberg), pp. 108–124, Springer-Verlag, 2008.

[8] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Tapiador, and A. Ribagorda, "Advances in ultralightweight cryptography for low-cost rfid tags: Gossamer protocol," in *Information Security Applications: 9th International Workshop, WISA 2008, Jeju Island, Korea, September 23-25, 2008, Revised Selected Papers*, (Berlin, Heidelberg), pp. 56–68, Springer-Verlag, 2009.

[9] N. J. Hopper and M. Blum, "Secure human identification protocols," in *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology*, ASIACRYPT '01, (London, UK), pp. 52–66, Springer-Verlag, 2001.

[10] H.-M. Sun, W.-C. Ting, and K.-H. Wang, "On the security of chien's ultralightweight rfid authentication protocol," *IEEE Transactions on Dependable and Secure Computing*, vol. 99, no. PrePrints, 2009.

[11] P. D'Arco and A. De Santis, "Weaknesses in a recent ultra-lightweight rfid authentication protocol," in *AFRICACRYPT'08: Proceedings of the Cryptology in Africa 1st international conference on Progress in cryptology*, (Berlin, Heidelberg), pp. 27–39, Springer-Verlag, 2008.

[12] T. Cao, E. Bertino, and H. Lei, "Security analysis of the sasi protocol," *IEEE Trans. Dependable Secur. Comput.*, vol. 6, no. 1, pp. 73–77, 2009.

[13] P. Peris-lopez, J. C. Hern, J. M. Estevez-tapiador, and A. Ribagorda, "Emap: An efficient mutual authentication protocol for low-cost rfid tags," in *OTM Federated Conferences and Workshop: IS Workshop*, pp. 352–361, Springer-Verlag, 2006.

[14] P. Peris-lopez, J. C. Hern, J. M. E. Tapiador, and A. Ribagorda, "Lmap: A real lightweight mutual authentication protocol for low-cost rfid tags," in *Proc. of 2nd Workshop on RFID Security*, p. 06, Ecrypt, 2006.

[15] P. Peris-lopez, J. C. Hern, J. M. Estevez-tapiador, and A. Ribagorda, "M 2 ap: A minimalist mutual-authentication protocol for low-cost rfid tags," in *Proc. of International Conference on Ubiquitous Intelligence and Computing UIC'06, LNCS 4159*, pp. 912–923, Springer-Verlag, 2006.

[16] T. Li and G. Wang, "Security analysis of two ultra-lightweight rfid authentication protocols," in *In IFIP SEC*, pp. 14–16, 2007.

[17] E. Gamal Ahmed, E. Shaaban, and M. Hashem, "Lightweight Mutual Authentication Protocol for Low Cost RFID Tags," *ArXiv e-prints*, May 2010.

[18] J. Katz and J. S. Shin, "Parallel and concurrent security of the hb and hb+ protocols," in *In Proc. Advances in Crypotology (EUROCRYPT 2006) (2006), LNCS*, pp. 73–87, Springer, 2006.

[19] Z. Li, G. gong, and Z. Qin, "Secure and efficient HB-CM entity authentication protocol," in *Cryptology ePrint Archive, Report 2009/444*, 2009.

요약문

저가형 RFID 태그의 활용이 광범위해짐에 따라 태그 위조로 인한 피해가 심각한 문제로 대두되고 있다. 이 뿐만 아니라, RFID 태그가 특정 물체나 개인을 식별할 수 있기 때문에 개인 정보에 관한 논란도 발생하고 있다. 이러한 문제들을 해결하기 위하여 고가의 태그는 표준 암호화 알고리즘을 이용하여 인증 프로토콜을 실행할 수 있다. 그러나 대부분의 저가형 태그가 가진 자원으로는 표준 암호화 알고리즘을 설계하기 어렵다는 문제점이 있다. 지금까지 이러한 저가형 태그에 적합한 안전한 인증 프로토콜을 위한 많은 연구가 진행되어 왔고, 그것들은 다시 경량 인증 프로토콜과 초경량 인증 프로토콜로 분류할 수 있다.

경량 프로토콜은 태그에서 난수 생성 알고리즘이나 주기적 덧붙임 검사 코드(Cyclic Redundancy Checksum code)와 같은 함수를 포함할 수 있지만 해쉬 함수를 포함하여 그 이상의 자원을 요구하는 함수를 포함하지 않는 프로토콜의 분류를 말한다. 반면, 초경량 프로토콜은 태그에서 비트 단위 연산만을 이용하여 구성된 프로토콜을 일컫는다. 초경량 프로토콜은 경량 프로토콜에 비하여 작은 양의 자원으로 설계할 수 있다는 장점이 있지만 태그의 이동 동선이 추적되는 것을 피하기 위하여 매 세션마다 태그와 리더가 공유되는 비밀키를 갱신해야하고 그 과정에서 분산된 데이터 베이스 사이에 동기화 문제가 발생할 수 있는 것이 단점이 될 수 있다. 또한 태그가 프로토콜에서 활용될 수 있는 난수를 생성할 수 없기 때문에 경량 프로토콜에 비하여 공격자가 비밀키를 분석하고 추출하는 과정이 상대적으로 수월해질 수 있는 여지가 있다.

경량 프로토콜 중에서는 HB^+ 프로토콜이 연산의 효율성이 우수하지만 GRS이라 불리는 MITM 공격에 취약한 것으로 알려졌다. $HB^\#$ 프로토콜은 GRS 공격에 대한 HB^+ 프로토콜의 취약점을 보안하였다. HB^+ 프로토콜이 멀티 라운드 프로토콜이고 태그와 리더가 벡터를 비밀키로 공유했던 것과는 달리 $HB^\#$ 프로토콜은 단일 라운드 프로토콜이며 태그와 리더는 행렬을 비밀키로 공유한다는 차이점이 있다. 그 과정에서 HB^+ 프로토콜보다 태그와 리더 사이의 통신양이 감소한 반면, 메모리 요구량이 크게 증가하였다. 그러나, $HB^\#$ 프로토콜은 이후에 OOV라고 지

명된 새로운 형태의 MITM 공격에 취약한 것으로 밝혀졌다. 가장 최근에 발표된 초경량 프로토콜인 Gossamer 프로토콜은 기존 초경량 프로토콜들이 가지고 있던 문제점들을 보완하였지만 그 과정에서 빈번한 modular 덧셈과 96을 modulus로 한 modulo 연산을 사용하였다. modular 덧셈을 구현하기 위해서는 비트 단위의 연산을 구현할 때 보다 많은 자원을 필요로 하게 되고 modulo 연산을 수행함에 있어 modulus가 2의 제곱이 아닐 경우 곱셈 연산을 동반하게 된다. 따라서 Gossamer 프로토콜은 비트 단위 연산만을 이용하여 구현한다는 초경량 프로토콜의 분류 기준에 비추어 과도한 양의 연산 및 자원을 요구하게 된다.

본 논문은 크게 세 가지의 문제를 해결한다. 첫 째, HB^{+} 프로토콜보다 작은 통신양으로 GRS 공격을 방어할 수 있는 수정 HB^{+} 프로토콜을 제시하고 안전성을 분석한다. 둘 째, $HB^{\#}$ 프로토콜에서 두 개의 비밀키 벡터를 추가함으로 OOV 공격을 방어하는 동시에 $HB^{\#}$ 프로토콜이 방어하던 공격들 또한 막아낼 수 있음을 보인다. 셋 째, 태그에서 비트 단위 연산만을 사용하면서 Gossamer 프로토콜과 같은 수준의 안전성을 제공할 수 있는 새로운 초경량 RFID 인증 프로토콜을 제시하고 그 성능을 분석한다.

감사의 글

생각했던 것 보다 시간이 빠르게 지나갔습니다. 졸업을 하게 되었군요. 가까이서, 멀리서 정성 껏 지도해 주신 홍성제 교수님, 감사합니다. 예원에서 사주신 갈치는 정말 맛있었습니다. 다음 번에는 제가 교수님께서 좋아하시는 굴비 사드리고 싶습니다. 바쁘신 중에 시간을 내어 지도해주신 김 종 교수님과 논문 심사에 참여해 주시고 열정을 가르쳐 주신 김 장우 교수님께도 감사를 드리고 싶습니다. 앞으로 더욱 좋은 모습으로 보답하겠습니다.

그 동안 삼촌처럼 저를 돌봐주신 진석이형, 무언가 안될 때 마다 마냥 형 자리에 찾아가고 싶었습니다. 의지할 사람이 되어 주셔서 감사합니다. 못하시는게 없는 유나 누나, 좋아하시는 번 한번 못사드리고 헤어지니 아쉽네요. 영민이형, 부디 시크릿 쉐어링이 잘 되서 행복하시길. 묵묵히 자기 일을 다 하는 상호형, 그 동안 고마운 일이 많았어요. 형이 볼링장에서 스핀 넣는 날이 속히 오길. 디젤카페를 바라보던 탁균이 형의 흐뭇한 미소와 태호형의 레고 머리가 곧 그리워질 겁니다. 형들이 있어 든든했어요. 병영이 형의 청명한 키보드 소리, 여전히 적응 안되는 종혁이의 친철한 말투, 재혁이의 뾰족 머리, 정신 없이 택배를 수령하던 나경이도 그리울 겁니다. 그리고 올해의 배달왕 지훈이, 수고했습니다. 여러분과 함께 했던 소소한 일상을 오랫동안 잊지 않고 간직하겠습니다. 고맙습니다.

무료한 시기에 즐거움을 더해주었던 보미·은진 누나 무탈하세요. 사랑하는 포항 중앙 교회 19기 친구들. 우성·종호·홍식·은호·효준·상오·성철·윤민· 상욱·윤기·태형·도숙·건호· 진이·세진이·현지·유리·민지·선애·민혜, 등등… 여러분을 만난 것이 큰 축복입니다. 그 동안 걱정해주시고, 찾아 주시고, 기도해 주셔서 감사합니다. 그리고 즐거웠습니다. 타지에 가면 금방 보고싶을 겁니다. 또 봅시다.

마지막으로, 항상 저의 편이 되어주신 부모님과 사랑하는 누이에게 지난 2년을 바칩니다. 먼 타지 생활을 하는 동안, 언제든지 저를 반겨줄 가족이 있어 외로운 시간을 이겨낼 수 있었습니다.

대학원에 들어오기 전, 좋은 사람들 만날 수 있게 해달라고 기도했습니다. 그 기도를 들어주신 하나님께 감사를 드립니다. 교회 이야기만 나오면 눈에 쌍심지를

켜고 달려드는 우리 탁균이 형이 이 문단에 노여워하지 않길 바라며 부디 교회에 대한 잘못된 오해를 풀 수 있는 날이 있길 바랍니다.

오늘 밤, 여러분과 저 느려터진 정통연의 엘리베이터를 타온 날들이, 무척이나 소중합니다.

이력서

성　　명 : 이승광

학력

2005 – 2009　한동대학교 전산전자공학부　(B.S.)
2009 – 2011　포항공과대학교 컴퓨터공학과　(M.S.)

논문실적

- 이승광, 홍성제, 김종, “비동기화 공격 방지를 위한 수정 SASI 프로토콜”, 한국 정보 보호학회 영남지부 학술발표대회, pp 126-134, 2010년 4월 3일.