전 방향 안전성을 제공하는 효율적인 RFID 보안 프로토콜☆

Efficient RFID Secure Protocol with Forward Secrecy

김 성 윤* 김 호 원**
Seongyun Kim Howon Kim

요 약

본 논문에서는 일 방향 해쉬 함수 기반의 저 비용 상호 인증 RFID 프로토콜(OHLCAP) 기반의 안전하고 효율적인 수동형 RFID 프로토콜을 제안한다. 제안하는 프로토콜의 기반이 되는 OHLCAP 프로토콜과 지금까지 알려진 OHLCAP의 보안 취약성을 소개하고 분석을 통해 취약성을 제거할 수 있는 방안을 제시한다. 이를 기반으로 제안하는 수동형 RFID 프로토콜을 제안하며, 프로토콜의 계산적 성능과 보안성을 증명한다. 본 논문에서 제안한 프로토콜은 도청 공격, 리더 또는 태그의 위장 공격, 비동기화 공격, 재전송 공격에 안전하며, 태그 추적이 불가능하고 전 방향 안전성 특성을 가진다.

ABSTRACT

We proposed the secure and efficient passive RFID protocol which is based on one-way hash based low-cost authentication protocol (OHLCAP). The paper introduces OHLCAP and the vulnerabilities of OHLCAP and suggests security solutions by analyzing them. Afterwards, The paper presents the proposed protocol and demonstrates computational performance and security of the protocol. This protocol not only has the resistances against eavesdropping attack, impersonation attack, desynchronization attack, and replay attack but also provides untraceability and forward secrecy.

☞ keyword : RFID Protocol(RFID 프로토콜), Security(보안), Mutual Authentication(상호 인증), Forward Secrecy(전 방향 안전성)

1. Introduction

Radio Frequency Identification (RFID) technology, typically consisting of tag, reader, and back-end database,has been widely used in many applications such as supply management, inventory management and theft detection. Due to relatively long identification range, low cost, and high speed, RFID technology system is expected to replace current barcode system. In spite of these advantages, however, there are many security vulnerabilities on current RFID standard such as exposure of access password and traceability[1]. In order to solve these problems and provide more security function, many researchers have studied hash-based secure protocols[4,6-11]. As a matter of fact, the hash algorithm was considered infeasible to apply to RFID tag because the chip area of its hardware implementation is too large to put into the tag and its throughput also doesn't meet RFID tag requirement. However, recent surveys show that the hardware implementations of SHA1 and one of the 5 final candidates of SHA3 competition, Keccak whose cryptography week point have not been found yet could be employed for RFID tag[2,3]. Therefore utilizing hash-based protocol could be the practical way to provide RFID security communication in the near future.

* 준 회 원 : 부산대학교 컴퓨터공학과 석사 과정
kims7y4@pusan.ac.kr
** 정 회 원 : 부산대학교 컴퓨터공학과 교수
howonkim@pusan.ac.kr (교신저자)
[2011/04/15 투고 – 2011/05/02 심사 – 2011/09/27 심사완료]
☆ 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.2010-0026621)
☆ A preliminary version of this paper appeared in ICONI/APIC-IST 2010, Dec 16-20, Mactan Island, Philippines.This version is improved considerably from the previous version by including new results and features.

The paper is organized as follows: Section 2 introduces a one-way hash based low-cost authentication protocol (OHLCAP) which the proposed protocol is based on. Section 3 presents all the known security vulnerabilities in OHLCAP and the analysis of these faults. Section 4 describes modified OHLCAP which is secure and efficient. Section 5 gives analysis of the proposed protocol in terms of its security and computational efficiency by comparing with other RFID secure protocols. Section 6 concludes the paper.

2. A One-Way Hash based Low-Cost Authentication Protocol (OHLCAP)

Choi et al proposed an one-way hash based low-cost authentication protocol (OHLCAP)[4]. This protocol consists of set-up and mutual authentication phases.

2.1 Notations

- $H(\ )$: one-way hash function, H: $0,1^{*} \rightarrow 0,1^{l}$
- ID: a confidential identification of tag, l bit
- PID: a previous identification of tag, l bit
- GI_i: i-th group index, l bit
- K: secret key shared by back-end database and all tags, l bit
- S: tag secret key shared by back-end database and a certain tag, l bit
- c: counter stored in tag, l bit
- r: random number generated by reader, l bit
- t: random number generated by tag, l bit
- B_R: right half of message B
- B_L: left half of message B
- $\oplus$: xor(exclusive-or) operation
- $\|$: concatenation of two operands
- m: the number of tags in every group at back-end database
- tn: the number of tags in back-end database

2.2 Set-up phase

The back-end database divides all tags into n groups and every group has m tags, so the total number of tags is up to $n \times m$. The database assigns index GI to every group and stores common secret key K, tag identification ID and tag secret key S. The database scheme is shown at Table 1.

(Table 1) Back-end Database Scheme

Goup Index	Shared Secret	ID	Tag Secret
GI_1	K	ID_1 ⋮ ID_m	S_1 ⋮ S_m
⋮	⋮	⋮	⋮
GI_n	K	$ID_{(n-1)m+1}$ ⋮ ID_{nm}	$S_{(n-1)m+1}$ ⋮ S_{nm}

The tag stores its own ID, GI, K, S and counter variable c which is initialized to arbitrary value. Whenever a tag receives a query from a reader, the tag increases a counter c.

2.3 Mutual Authentication phase

Mutual authentication protocol is comprised of 5 steps.

Step 1: A reader generates a random value r and sends Query with r to a nearby tag.

Step 2: After receiving Query and r, the tag checks a random value r whether it is all zero value or not. If r value is all zero, the tag sends "stop" message to the reader and halts the protocol. Otherwise, the tag

computes A^1, A^2, and B as

$A^1 = K \oplus c$,

$A^2 = ID + (GI \oplus r \oplus c) mod (2^l - 1)$,

$B = H(ID \| (S \oplus GI) \| (r \oplus c))$.

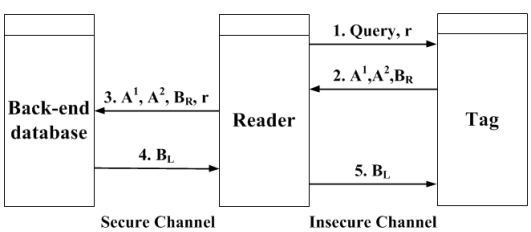
And then the tag sends A^1, A^2, and B_R to the reader.

Step 3: Upon receiving A^1, A^2, and B_R from the tag, the reader forwards A^1, A^2, B_R and r to the back-end database. The back-end database computes $c' = A^1 \oplus K$ and $ID_j' = A^2 - (GI_k' \oplus r \oplus c') mod (2^l - 1)$ using all group indices GI_k', $k \in \{1,...,n\}$, and ID indices $j \in \{1,...,m\}$. If the database doesn't match any < ID_j', GI_k' > pair for A^1, A^2 received from the tag, halt the protocol.

Step 4: If the database finds such a matched < ID_j', GI_k' > pair, it computes $B' = H(ID_j' \| S \oplus GI_k' \| s(r \oplus c))$ and checks whether B_R' equals to B_R received from the tag. If they are the same, back-end database authenticates this tag successfully and sends the left half of B', B_L', to the reader. The reader which receives B_L' forwards this message to the tag.

Step 5: The tag checks whether received B_L' is equal to computed B_L. If they are equal, the tag authenticates the back-end database.

The OHLCAP Mutual authentication protocol is shown at Fig.1. It is normally assumed that the channel between the back-end database and reader is considered secure whereas the channel between the reader and tag is thought to be insecure.



(Figure 1) OHLCAP Mutual Authentication Protocol

3. Security Vulnerabilities of OHLCAP

OHLCAP mutual authentication protocol was considered one of the most efficient RFID secure protocol in that tagcomputes only one hash function. However, some security vulnerabilities have been found by several researchers[5,6]. the sub sectionsshow two major security faults in OHLCAP and then we analyze all the known attacks.

3.1 Tag impersonation attack

Kwon et al proved that OHLCAP can be attacked by tag impersonation attack by sniffing only one communication session[5]. By sniffing one session, attacker can obtain following information.

- r: random number generated by the reader
- $A^1 (= K \oplus c)$
- $A^2 (= ID + (GI \oplus r \oplus c) mod (2^l - 1)$
- B_R, B_L $(B = H(ID \| (S \oplus GI) \| (r \oplus c)))$

Using this information, the attacker can generate valid responses ($\overline{A^1}$, $\overline{A^2}$, $\overline{B_R}$) of the tag to the query with random number $\bar{r}$ from the reader as follows:

$\overline{A^1} = A^1 \oplus r \oplus \bar{r}$, $\overline{A^2} = A^2$, $\overline{B_R} = B_R$

After receiving $\overline{A^1}$, $\overline{A^2}$, $\overline{B_R}$, back-end database will validate as following steps:

1. back-end database obtains the counter c
   $c = K \oplus \overline{A^1} = K \oplus A^1 \oplus r \oplus \bar{r} = c \oplus r \oplus \bar{r}$
2. back-end database gains the same ID and B with the previous sniffed session because
   $ID' = \overline{A^2} - (GI \oplus \bar{r} \oplus c') = A^2 - (GI \oplus \bar{r} \oplus c \oplus r \oplus \bar{r}) = A^2 - (GI \oplus r \oplus c) = ID$
   $B' = H(ID \| (S \oplus GI) \| (\bar{r} \oplus c')) = H(ID \| S \oplus GI|)$

$|(r \oplus c)) = B$ so the database would accept $B_R^{'}$ as valid value. Therefore attacker would succeed tag impersonation attack.

3.2 Traceability Using Counter Information

Ha et al proved that if a tag's messages are caught in two successive sessions, the attacker can trace the tag[6]. It is assumed that theattacker knows certain tag's responses from two successive sessions, which are $A_p^1 = K \oplus c_p$ and $A_c^1 = K \oplus c_c$. Because these messages are from successive sessions, $c_c = c_{p+1}$. By xoring responses of two successive sessions, the attacker obtains $A_p^1 \oplus A_c^1 = c_p \oplus c_c$. This value is always an l's-run value from LSB. Therefore the attacker can trace the tag by sniffing all the packets and matching whether the xored value of two messages is a 1's-run value from LSB or not.

3.3 Analyzing Vulnerabilities of OHLCAP

Except the previous major two attacks, there are three more known attacks in OHLCAP: 'impersonation by maliciously updating random number'[6], 'recovering a tag's ID'[5], and 'tracing using compromised tag ID'[5].

After analyzing all the known attacks, we find two factors to make OHLCAP vulnerable. One of them is using the counter. Because of the counter, the dependency exists between two consecutive sessions. Also some attacks using mathematical characteristic of the counter also exist. Counter-related attacks are 'traceability using counter information' presented at section 3.2, 'recovering a tag's ID', and 'tracing using compromised tag ID'. The other factor is caused by dependency between value r and c. Because A^2 and B include $r \oplus c$ expression, the attacker can impersonate tag by manipulating either r or c properly. 'Tag impersonation attack' showed at section 3.1, and 'impersonation by maliciously updating random number' are the attacks using dependency of r and c values.

As we mentioned before, OHLCAP protocol is efficient in light of the computation cost, so there are several attempts toremove such vulnerabilities to achieve not only the efficiency but also safety. Ha et al. replaced counter with random value t and eliminated dependency between r and c[6]. Since it is group-based protocol, however, Ha et al's protocol averagely computes $1/2 \times m$ time hash operations per each tag's session and it doesn't provide forward secrecy. He et al. adopted forward secrecy to OHLCAP but this protocol isn't efficient because 4 times of hash function have to be computed at both the back-end database and tag[7].

4. The Proposed Protocol

As we referred to, we analyzed that the security weak points of OHLCAP are the use of counter and dependency between value r and c. In this section, we suggests the efficient and secure passive RFID protocol based on OHLCAP by eliminating referred vulnerabilities and by adding the forward secrecy function.

In order to prevent desynchronizing between the reader and tags, we add previous ID, PID and previous tag secret S, PS to the back-end database. The other information is the same as section 2.1 except that the length of one-way hash function output is $2l$ bit. The modified database scheme is shown as Table 2.

(Table 2) Modified Back-end Database Scheme

Goup Index	Shared Secret	Previous ID	ID	Previous Tag Secret	Tag Secret
GI_1	K	PID_1 ⋮ PID_m	ID_1 ⋮ ID_m	PS_1 ⋮ PS_m	S_1 ⋮ S_m
⋮	⋮	⋮	⋮	⋮	⋮
GI_n	K	$PID_{((n-1)m+1)}$ ⋮ PID_{nm}	$ID_{((n-1)m+1)}$ ⋮ ID_{nm}	$PS_{((n-1)m+1)}$ ⋮ PS_{nm}	$S_{((n-1)m+1)}$ ⋮ S_{nm}

The proposed mutual RFID authentication protocol is shown as follow:

Step 1: A reader generates a random value r and sends Query with r to a nearby tag.

Step 2: After receiving Query and r, the tag checks a random value r whether it is all zero value or not. If r value is all zero, the tag sends "stop" message to the reader and halts the protocol. Otherwise, the tag computes A^1, A^2, and B as

$A^1 = K \oplus t$,

$A^2 = ID + (GI \oplus r \oplus t) mod(2^l - 1)$,

$B = H(ID \| (S \oplus GI) \| r \| t)$.

And then the tag sends A^1, A^2, and B_R to the reader.

Step 3: Upon receiving A^1, A^2, and B_R from the tag, the reader forwards A^1, A^2, B_R and r to the back-end database. The back-end database computes $t' = A^1 \oplus K$ and $ID_j' = A^2 - (GI_k' \oplus r \oplus t') mod(2^l - 1)$ using all group indices GI_k', $k \in \{1,...,n\}$, and ID_j' indices $j \in \{1,...,m\}$.

(a) If the database doesn't find any < ID_j', GI_k' > pair for received A^1 and A^2, it tries to match with all the < PID_j', GI_k' > pairs in the database. If the matched tuple exists, then change ID value as $ID = PID$ and S value as $S = PS$. If the database doesn't match any < PID_j', GI_k' > pair for A^1, A^2 received from the tag, the protocol is halted.

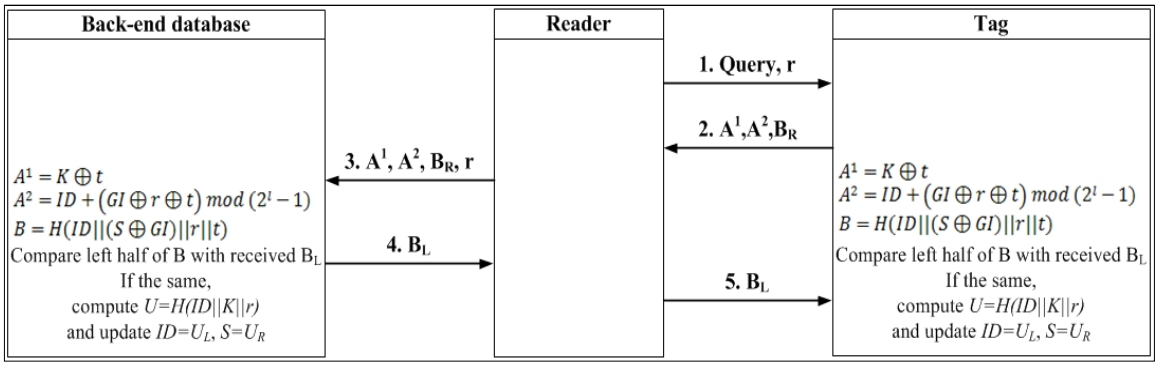
Step 4: If the database finds such a matched < ID_j', GI_k' > pair, it computes $B' = H(ID_j' \| S \oplus GI_k' \| r \| t)$ and checks whether B_R' equals to B_R received by the tag. If they are the same, back-end database authenticates this tag successfully, sends the left half of B', B_L', to the reader, computes $U = H(ID \| S \| t)$ and then updates $PID = ID$, $PS = S$, $S = U_L$, and $ID = U_R$. The reader which receives B_L' forwards this message to the tag.

Step 5: The tag checks whether received B_L' is equal to computed B_L. If they are equal, the tag authenticates the back-end database.

To provide forward secrecy and provide synchronization in the proposed protocol, we add previous PID and previous tag secret PS into the back-end database and also make sub procedure (a) in step 3.

c value could cause counter characteristic related other attacks, so we replace c with t, random number generated at the tag. To remove the dependency between r and t, we alter xor operation to concatenation operation. By doing these, the proposed protocol is safe from all the known attack for OHLCAP.

The security attributes of the proposed protocol is mainly originated from the preimage resistance property of hash function. Because tag secret value is one of the hash function input, the safety of the protocol counts on the length of tag secret. When we use SHA1 hash function, the attacker has to compute 2^{80} hash operations for finding tag secret if the invader uses brute force attack.



(Figure 2) the detail of the proposed protocol

The detail illustration is shown at figure 2.

5. Analysis of the Proposed Protocol

we analyze the proposed protocol in terms of the security and computational efficiency aspect.

5.1 Security Analysis

The goal of the attacker is to spoof the reader or tag by making the valid response of the query from the reader or making the verification data for tag individually. Adversary can selectively eavesdrop, reply, drop, and modify arbitrary messages and its computation ability is limited but very high.

■ Eavesdropping attack resistance

The attacker has to obtain the secret information through the received information A^1, A^2, $\mathrm{B_R}$ and $\mathrm{B_L}$ without knowing any secret value. Even if $\mathrm{A}^1(=\mathrm{K}\oplus\mathrm{t})$ is eavesdropped, the attacker doesn't obtain K without knowing t. In case $A^2(=ID+(GI\oplus r\oplus t)mod(2^l-1))$ is captured, the attacker can't extract ID, GI, and t. In case $B(=H(ID\|(S\oplus GI)\|r\|t))$ is captured, the attacker doesn't acquire any secret information because hash function is one-way. Even if the adversary obtains t, ID, GI somehow, the attacker can't make the valid response in the next session because of updating ID, S values of the tag.

■ Impersonation attack resistance

The impersonation attacks can be categorized as the reader impersonation attack and tag impersonation attack. Because the proposed protocol is mutual authentication protocol, the protocol has to guarantee that it is impossible for any unauthorized tag or reader to spoof legal tag or reader respectively.

The reader impersonation attack is impossible because the attacker can't make legitimate B_L using A^1, A^2 and B_R without knowing any secret information.

In case of tag impersonation attack, the attacker tag has to generate proper A^1, A^2 and B_R. However since the attacker doesn't know any securely shared values, the attacker hardly make legitimate response from the reader. Moreover, the previous known tag impersonationattacks depend on the relation between r and c, but by replacing $r\oplus c$ with $r\|t$ in B, the proposed protocol is safe from all the known tag impersonation attacks.

■ Untraceability

The attacker should not distinguish two tags when two tags are adjacent. The proposed protocol updates r, t value per every session so it is strongly resistant to not only traceability but also replay attack because all the responses A^1, A^2 and B_R from the tag get changed. Moreover, in last phase, the proposed protocol updates ID by using hash operation if the protocol successfully ends. Therefore, it is hard for the adversary to tell tag's identification between the two.

■ Desyncronization attack prevention

Desyncronization attack can take place at step 2 and 4 in the mutual authentication protocol. At step 2, even though A^1, A^2 and B_R messages are lost, this attack doesn't affect any changes at either the tag or the database. When the message is lost in step 4, on the other hand, back-end the database updates ID and S, but the tag doesn't, so this attack could lead to desyncronization state. To prevent this attack, back-end database temporarily store previous ID, PID and previous tag secret S, PS as a backup. Once desyncronization happens, back-end database executes step 3 (a) in mutual authentication protocol so that back-end database can restore the condition before the desyncronization state. This restoration has some computation overhead but it is only for the back-end database not for the tag so a minor damage for the entire system.

■ Providing forward secrecy

The forward secrecy means that even if the attacker knows all the secret information, the previous communication traffic is locked securely in the past. To provide forward secrecy in the protocol, weadd the process of updating tag identification and secret value at the end of the protocol. By altering the tag information and secret value, it needs 2^{2l} hash operations for the invader to find previous tag information when he uses brute force attack. Updating some secret information individually could lead to desyncronization state between the back-end database and tag so we add PID, PS to the database and make additional process to prevent the wrong state.

■ Comparison with other protocols

Table 3 compares security functions in some passive RFID protocols. The proposed protocol can provide all the functions.

(Table 3) Provided Security Functions

Protocol	Impersonation attack resistance	Untrace ability	Forward Secrecy
CRAP[8]	O	O	X
LCAP[9]	X	O	O
LCRP[10]	O	X	O
LRMAP[11]	O	O	O
Proposed	O	O	O

5.2 Efficiency analysis

Although the proposed protocol has space overhead at the back-end database, our protocol needs just 0(1) hash operations to search the proper tag. Table 4 shows the computational overhead in passive RFID protocols. The result of the proposed protocol is shown as follow.

(Table 4) Computational Overhead

Protocol	# DB hash	# Tag hash	# Tag RN generate
CRAP[8]	*tn* /2 +1*	2	1
LCAP[9]	*tn* /2 +1*	3	1
LCRP[10]	4	4	1
LRMAP[11]	3	3	1
Proposed	2	2	1

*: average

6. Conclusions

RFID is not only power, time, and memory constrained system but also user-close system so it is very important to develop light and secure protocol. In this paper, we introduce original OHLCAP and show vulnerabilities of this protocol. After analyzing all the known vulnerabilities, we find countermeasures to eliminate security weak points of OHLCAP and then make secure OHLCAP with forward secrecy which is the proposed protocol. Later, we analyze the computational performance and security of the protocol. The result indicates this protocol increases only one more hash computation to both the back-end database and the tag even though this protocol provides forward secrecy and is resistant to eavesdropping, impersonation, desyncronization, replay, and tracing attacks.

References

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

[2] Elif Bilge Kavun and Tolga Yalcin, "A Lightweight Implementation of Keccak Hash Function for Radio-Frequency Identification Applications", *RFIDSec 2010*, vol. 6370, pp.258-269, 2010

[3] M. O'neill, "Low-Cost SHA-1 Hash Function Architecture for RFID Tags", *RFIDSec 2008*, 2010

[4] Eun Young Choi, Su Mi Lee, Dong Hong Lee, "Efficient RFID Authentication Protocol for Ubiquitous Computing Environment", *Embedded and Ubiquitous Computing*, vol. 3823, pp. 945 – .954, 2005

[5] D. Kwon, D. Han, J. Lee, and Y. Yeom, "Vulnerability of an RFID Authentication Protocol Proposed in at SecUbiq2005", *EUC2006, LNCS 4097*, pp. 262-270, 2006.

[6] JeaCheol Ha, SangJae Moon, Juan Manuel Gonzalez Nieto, Colin Boyd, "Security Analysis and Enhancement of One-Way Hash Based Low-Cost Authentication Protocol", *Emerging Technologies in Knowledge Discovery and Data Mining*, vol. 4819, pp. 574-583, 2007.

[7] He Lei, Lu Xin-mei, Jin Song-he, Cai Zeng-yu, "A One-way Hash based Low-cost Authentication Protocol with Forward Security in RFID System", *2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics*, vol.2, pp.269-272, 2010.

[8] K. Rhee, J. Kwak, S. Kim, and D. Won, "Challenge-Response Based on RFID Authentication Protocol for Distributed Database Environment", *SPC'05, LNCS 3450*, pp. 70-84, 2005.

[9] S. Lee, Y. Hwang, D. Lee, and J. Lim, "Efficient Authentication for Low-cost RFID Systems", *ICCSA'05, LNCS 3480*, pp. 619-627, 2005.

[10] Dimitriou, "A lightweight RFID protocol to protect against traceability and cloning attacks", *Security and Privacy for Emerging Areas in Communications Networks-2005*, pp. 59-66, Sep. 2005

[11] J.C. Ha, J.H. Ha, S.J. Moon, and C. Boyd, "LRMAP: Lightweight and Re- synchronous Mutual Authentication Protocol for RFID System", *ICUCT'06, LNCS 4412*, pp. 80-89, 2006.

◐ 저 자 소 개 ◑

김 성 윤

2009년 부산대학교 정보컴퓨터공학과 졸업(학사)
2011년 부산대학교 대학원 컴퓨터공학과 졸업(석사)
2011년~현재 LG전자 차세대통신연구소 연구원
관심분야 : RFID/USN 정보보호 기술, 지그비 보안, etc.
E-mail : kims7y4@pusan.ac.kr

김 호 원

1993년 경북대학교 전자공학과 졸업(학사)
1995년 포항공과대학교 대학원 전자전기공학과 졸업(석사)
1999년 포항공과대학교 대학원 전자전기공학과 졸업(박사)
1998년~2008년 한국전자통신연구원(ETRI) 정보보호연구단 선임연구원/팀장
2008년~현재 부산대학교 정보컴퓨터공학부 조교수
관심분야 : RFID/USN 정보보호 기술, 타원곡선 및 초타원곡선 암호 이론, VLSI 설계, embedded system, etc.
E-mail : howonkim@pusan.ac.kr