해쉬 기반 RFID 태그를 위한 인증 프로토콜의 보안성 향상☆

Security Enhancing of Authentication Protocol for Hash Based RFID Tag

전 진 오* 강 민 섭**
Jin-Oh Jeon Min-Sup Kang

요 약

본 논문에서는 해쉬 기반 RFID 태그를 위한 보안성이 향상된 인증 프로토콜을 제안하고, 제안한 인증 프로토콜 기반으로 한 RFID 태그의 디지털 코덱을 설계한다. 제안한 프로토콜은 태그와 back-end 서버 사이에서 3-way 질의 응답 인증 프로토콜을 기본으로 하고 있으며, 안전한 인증 메커니즘을 구현하기 위해, ISO/IEC 18000-3 표준에서 규정된 3가지 타입의 프로토콜 패킷을 개선된 형태로 수정한다. 제안한 방법은 Man-in-the-middle과 Replay attacks과 같은 능동 공격의 방어에 특히 유효하다.

제안된 프로토콜의 효과를 검증하기 위하여 RFID 태그에서의 디지털 코덱은 Verilog HDL을 사용하여 설계하였고 Hynix 0.25 μm standard-cell library을 갖춘 Synopsys Design Compiler을 이용하여 합성하였다. 보안 분석 및 실험결과를 통해, 본 논문에서 제안된 방법이 사용자의 데이터 보안, 태그 익명성, Man-in-the-middle attack 예방, replay attack, 위조방지 및 위치 추적 등에서 성능이 개선됨을 보였다.

ABSTRACT

In this paper, we first propose the security enhancing of authentication protocol for Hash based RFID tag, and then a digital Codec for RFID tag is designed based on the proposed authentication protocol. The protocol is based on a three-way challenge response authentication protocol between the tags and a back-end server. In order to realize a secure cryptographic authentication mechanism, we modify three types of the protocol packets which defined in the ISO/IEC 18000-3 standard. Thus active attacks such as the Man-in-the-middle and Replay attacks can be easily protected.

In order to verify effectiveness of the proposed protocol, a digital Codec for RFID tag is designed using Verilog HDL, and also synthesized using Synopsys Design Compiler with Hynix 0.25 μm standard-cell library. Through security analysis and comparison result, we will show that the proposed scheme has better performance in user data confidentiality, tag anonymity, Man-in-the-middle attack prevention, replay attack, forgery resistance and location tracking.

☞ KeyWords : three-way 인증 프로토콜, SHA-1 해쉬알고리듬, RFID 태그, three-way 질의응답, ISO/IEC 1800-3 표준, 디지털 코덱 설계, three-way authentication protocol, SHA-1 hash algorithm, RFID Tag, three-way challenge response, ISO/IEC 1800-3 standard, Digital Codec design.

1. Introduction

Radio Frequency Identification System (RFID) is the latest technology to play an important role for object identification as a ubiquitous infrastructure, which is used for identification of various products and goods in our automated world. Recently, with the advance of antenna and microchip design technology, it has the diversified application such as automatic tariffs payment, animal identification, tracking of product, automated manufacturing, and logistic control [1, 2].

The tags attached to products are used to identify

* 준 회 원 : Dept. of Computer Engineering, Anyang University, Adjunct Professor
jazzman@anyang.ac.kr
** 정 회 원 : Dept. of Computer Engineering, Anyang University, Professor
mskang@anyang.ac.kr
[2010/04/08 투고 - 2010/04/28 심사 - 2010/07/14 심사완료]
☆ This work was supported by IDEC, KAIST in Korea.

the object during production or in uses via radio frequency which may be passive or active [2].

RFID system consists of three different components; RFID tag, or transponder, and RFID reader, or transceiver, and back-end server [3].

The reader can inquire tags of their contents by broadcasting an RF signal, at a rate of several hundred tags per second, and from a range of several meters. RFID tags attached to products are used to identify the object during production or in uses via radio frequency which may be passive or active. A mutual authentication protocol for RFID system has been presented by Ohkubo et al. [4] based on hashing chain, which aimed to provide the forward secrecy. Unfortunately, the protocol cannot resist the replay attack [3].

The ISO/IEC 18000-3 standard defines a protocol for RFID tags that handles bi-directional communication between a reader and the tags [3,5,6]. Unfortunately, the data exchange between the reader and the tags on this protocol is not secure, and there are no mechanisms defined to authenticate a tag to the reader.

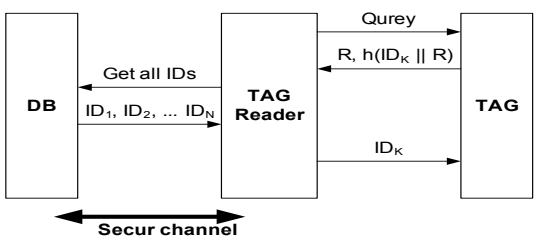
When this protocol is used for the challenge-response process, it is vulnerable to a man-in-the-middle attack, and then an eavesdropper can capture texts for both challenge and response by just sniffing with a protocol analyzer. Thus, a cryptographic authentication algorithm is necessary to protect branded goods from forgery.

An approach based on the AES cipher algorithm has been introduced for implementing strong cryptographic authentication on the tags [3]. However, main drawback of this algorithm requires a mount of hardware resources with much latency when implementing in hardware. In [7], the robust mutual authentication protocol has been proposed for the low-cost system to meet the privacy protection for tag bearers. However, this approach is has not given any results implemented in hardware. In addition, the existing protocols based on the ISO/IEC 18000-3 standard do not provide cryptographic authentication mechanism.

This paper proposes the security enhancing of authentication protocol for Hash based RFID tag [8]. The proposed protocol is based on a three-way challenge response authentication protocol between the tags and a back-end server. In addition, three types of the protocol packets are modified for protecting the active attacks such as the Man-in-the-middle and Replay attacks. In order to verify effectiveness of the proposed protocol, a digital Codec for RFID tag is designed using Verilog HDL, and also synthesized using Synopsys Design Compiler with Hynix 0.25 μm standard-cell library.

2. Related Work

RFID system is used for the automated identification of products, which is similar to smart cards. In this system, data can be stored and processed on the chip. In general, RFID system is composed of three components such as RFID reader, RFID tag, and Back-end server with database. Typical RFID system is shown in Fig. 1 [6].



(Fig 1) Typical RFID system

The reader includes antenna, transceiver and decoder which communicate with the tag, and it is

also used as an interface between the server and the tags. The tag which is placed on the object to be identified contains a transponder with a memory chip such as EPROM that possesses a Unique ID(UID). The server which is secure server has a database which stores the various information of each tag obtained from the reader in some useful manner. In general, Tag Reader can read all messages, and all communications are insecure, if Tag Reader is compromised.

The various command signals (queries) are generated in the reader, and the signals can be received in a tag when the tag is within range of the signal. The tag sends out its identification (R,h(IDx||R)) or encoded data to the reader, when responding to commands from the reader. The received ID then should send to the server to be processed.

Hash lock protocol based on hash function has been presented by MIT [8]. The reader has key k for each tag, and each tag holds the result metaID, metaID = hash(k) of a hash function. Although, this protocol offers good reliability at low cost, since metaID is fixed, the adversary can track the tag via metaID.

To resolve this problem, Randomized hash lock protocol has been introduced by MIT, which is an extension of the hash lock type protocol [3]. It requires the tag to have a hash function and a Random Number Generator(RNG). Then, each tag calculates the hash function based on ID and r generated by RNG, i.e., c = hash(ID ‖ r). The tag then sends c and r to the reader.

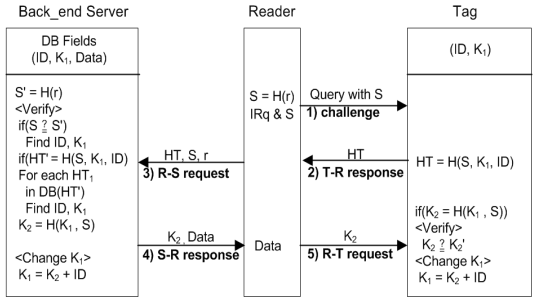
The reader sends the data to the server, and then the server calculates the hash function using inputs of the received r and each ID stored in the server. However, this protocol allows the location history of the tag to be traced if the secret information in the tag is revealed. Thus, this protocol cannot satisfy the forward security requirement.

3. Extended Mutual Authentication Protocol

3.1 Three-way authentication protocol

As described above, Randomized hash lock protocol requires RNG embedded into each tag in order to achieve location privacy. In this approach, another problem is needed a large processing time for computing hash function on the reader and for finding matched tag's ID list within the server [3,6]. The enhanced protocol is designed on the basis of the concept of 'the reader talks first'[9]. This means that any tag does not start transmitting unless it has received and properly decoded a command sent by the reader.

In our approach, the random number uses a method generated RNG in the reader for realizing low-cost RFID tag by reducing hardware-overhead of the tag. The proposed authentication protocol is based on a three-way challenge response protocol, which is an exchange of a request from reader to tag and a response from the tag to the reader. An execution of the protocol based on the cryptographic hash function is shown in Fig. 2.



(Fig 2) The proposed authentication protocol

In Fig. 2, we assume that the server stores tag information (ID, K_1, and data) in its database, where ID is chip serial number and K_1 represents a secret key. This key is stored in the server and the tag. Also, K_2 is used as shared key between the server and the tag, where S denotes one-way hash value obtained from hash function H(r), i.e. S = H(r). In this paper, we consider that the hash function is used for generating hash value in three components.

A detailed procedure for the proposed protocol will be given below, which operates as three-pass mutual authentication method. In this procedure, we will show how the proposed protocol of Fig. 2 is performed using extended three packets.

Step 1 (Challenge)

In the reader, RNG generates a random number, r and we obtain a hashed value S by calculating S = H(r). R(Reader) generates an extended IRq packet within S by modifying Inventory request(IRq) packet defined in the ISO/IEC 18000-3 standard[5], and then the packet sends to the T(Tag) as challenge. Table 1 shows an example of extended Inventory request format with S.

Step 2 (T → R response)

When T receives the extended IRq from R, T should generate hash value HT calculated by using K_1, ID and S, i.e., HT = H(S, K_1, ID). Then, HT is inserted into the extended IRs packet by modifying Inventory response packet (IRs) defined in the standard. T sends the extended IRq packet back to R as response. Table 2 shows an example of extended Inventory response format with HT.

Step 3 (R → B request)

R sends the received data set (HT, S, and r) to B (Back-end server) in order to detect the Man-in-the-middle attack. R sends the relevant data to the DB (database) safely. DB compares its data (ID, K_1) with the received data to find out the corresponding K_1 and ID through the following two steps. At the first step, it looks for ID and K_1 after checking of a condition, S=S'. At the second step, it proves ID and K_1 by applying the newly created S through 'if(HT' = H(S, K_1, ID))' calculation. That is, it confirms ID and K_1 after performing 'if(S=H(r)) && if(HT' = H(S, K_1, ID))' calculation to identify the user. In this step, authentication between B(R) and T is performed as the following two steps.

(1) First, B verifies whether r (received from R) is valid or not by comparing S with S', where S'=H(r). Since both components use the same hash function, illegal R can be easily detected by this verification. If both values of S and S' are identical, this proves the authenticity of R. Thus, the man-in-the-middle attack by illegitimate R and eavesdropper can be easily prevented.

(2) Next, if R is valid, then we obtain HT' by calculating H(S, K_1, ID) in B, i.e., HT' = H(S, K_1, ID). Note that HT' is used for authenticating T. Then, HT is also compared with HT' in entry its database. The process is iteratively repeated for each entry until it finds a match. If it can find a match, this means that the authentication of the T is succeeded; otherwise, it sends a fail message to the R to stop the process.

Step 4 (B → R response)

If the authentication process is successfully terminated in B(in Step 3), B should generate a new key K_2 for shared key by calculating H(K_1, S), i.e.,

$K_2 = H(K_1, S)$. This key can be used for updating K_1 key in Step 5, and then B forwards K_2 and its data to R.

Step 5 (R → T request)

R stores the received data sets in own memory, and it generates an extended SRq packet within K_2 by modifying Select request (SRq) packet defined in the standard [4]. And then the extended SRq packet is sent to the T as request. Table 3 shows an example of extended Select request format with K_2. As the similar manner in Step 4, K_2' is obtained by calculating $H(K_1, S)$ in T, i.e., $K_2' = H(K_1, S)$. Then, T verifies whether K_2 is valid or not by comparing K_2 with K_2'. If K_2 and K_2' are identical, T updates original key K_1, i. e., $K_1 = K_2 + ID$. Note that the changed new key K_1 is used to prevent replay attack on the used tag once.

3.2 Extended protocol frame formats

The protocol defined in the ISO/IEC 18000-3 standard can communicate at a frequency of 13.56Mhz [5,7], and it defines the mechanism to exchange instructions and data between two units (reader and tag) in both directions. For communication between two units, a reader sends a Request data to a tag, and receives a Response data from the tag. Request and Response packets are contained within a frame with Start-of-Frame (SOF) and End-of-Frame (EOF) the delimiters. In general, General Request format consists of SOF, Flag, Command Code, Parameters, Data, CRC and EOF. In the Command code, four types of command are defined: Mandatory, Optional, Custom, and Proprietary.

In the proposed authentication mechanism, two kinds of commands are used: Inventory and Select commands defined in Mandatory and Optional to communicate with two units, respectively.

Table 1 shows an example of the extended Inventory request packet format with the hashed value S obtained by modifying standard Inventory request format.

(Table 1) Example of extended Inventory request format with S

SOF	Flags	Invent.	Opt. AFI	Mask length	Mask-value	S	CRC	EOF
	8 bits	8 bits	8 bits	8 bits	0 - 64 bits	16 bits	16 bits	

The standard Request packet contains fields of Flags, Inventory, Optional AFI, Mask Length, Mask Value, and CRC [5]. In the extended version, field S is newly added to it. Before issuing the Inventory command for identifying Tag, Reader should be set the hashed value S to a field S within the format as shown in Table 1. S is assigned to 16 bits considering heavy load during transmission of the data, which is described in Step 1 of this chapter.

When receiving the Inventory request, Tag performs the Collision Management sequence. In other word, the purpose of the anti-collision sequence is to inventory the Tag present in the Reader field by their UID. Table 2 shows an example of the extended Inventory response packet containing HT which modifies standard Inventory response format [5].

(Table 2) Example of extended Inventory response format with HT

SOF	Flags	DSFID	HT	CRC	EOF
	8 bits	8 bits	64 bits	16 bits	

The standard Inventory response format contains the fields of Flags, DSFID, UID, and CRC [5]. In

the extended version, UID field is replaced to a field HT which has the hashed values described in Step 2 of this chapter, where the same data bits are assigned. The 64-bit UID is used to identify Tag sending the response. Thus, the proposed protocol(Fig. 2) based on the extended Inventory response packet provides location privacy because data HT are useless to an attackers.

Table 3 shows an example of the extended Select request packet with K_2 which modifies standard Select request format [5].

(Table 3) Example of extended Select request format with K_2

SOF	Flags	Select	UID	K_2	CRC	EOF
	8 bits	8 bits	64 bits	16 bits	16 bits	

The standard Select request contains the fields of Flags, Select, UID, and CRC. In the extended version, K_2 field is newly added to it. As we can see from our protocol shown in Fig. 2, K_2 is forwarded to Reader together the date retrieved from the memory, which is generated by hash function on the server.

After receiving the Select command, if K_2 is equal to K_2', i. e., K_2' = H(K_1, S), T changes the used key K_1 to a new key by adding K_2 to ID, i.e., K_1 = K_2 + ID (see Fig. 2). Note that K_1 is used to prevent replay attack on the tag.

4. Security Analysis

The proposed scheme has been evaluated the view point of the security requirement and compared between some protocols [7,8,9]. If the unique serial number is wiped at the supermarket checkout, and only product and manufacturer codes remain, a significant location privacy attack is still possible through tracking combinations of specific brands.

The challenge and response technique is used to ensure mutual authentication of Reader (Server) and Tag. For this, in the proposed protocol, all messages from Tag have been hashed such as HT=H(S, K_1, ID), and K_2 = H(K_1, S) as shown in Step 2 and Step 5, respectively. Since the hashed data means that eavesdropping is meaningless, data confidentiality of tag bearers is certainly guaranteed. In addition, the required all data for an application are stored in the server shown in Fig. 2 while user's privacy information is not stored in the tag in conventional approach [3,9]. Thus, the user privacy on data is strongly protected.

A Man-in-the-middle attack is not possible because our protocol is based on a mutual authentication. That is, the hashed value HT is used through procedures Step 1 to Step 3, and then this attack is prevented in Step 3.

Replay attack is that attackers eavesdrop all messages from each T, and then retransmit the message to the legitimate R. As described the previous chapter, we use two keys; K_1 for secret key and K_2 for shared key. K_1 in T is updated for every session, where K_1= K_2 +ID. Thus, the replay attack for T is detected and prohibited in Step 5.

In the proposed protocol, all data exchanges between T and R(Server) use the hashed value S generated in R, and then K_1 is changed by "K_2 + ID" for every session as shown in Step 5, where K_2 =H(K_1, S). Thus, tag anonymity is guaranteed and the location privacy of a tag bearer is not compromised.

To realize the forgery resistance, our approach uses the hashed data, HT = H(S, K_1, ID), shown in Step 2 and Step 3, where ID represents chip serial number embedded during the chip manufacturing. Whenever T generates HT, it refers to S. Thus,

forgery like simple copy is prevented. Intercepting or blocking of messages is a denial-of-service attack preventing tag identification. Our protocol does not particularly focus on providing data recovery. Table 4 shows the comparison of the security requirements and the possible attacks.

(Table 4) Comparison between authentication protocols

Protocols / Requirements	MAP [7]	HLS [8]	EHLS [8]	HBVI [9]	Proposed scheme
User data confidentiality	O	X	△	△	O
Tag anonymity	O	X	△	△	O
Mutual authentication	O	△	△	△	O
Reader authentication	O	X	X	X	O
Man-in-the-middle attack prevention	O	△	△	X	O
Replay attack prevention	O	△	△	O	O
Forgery Resistance	O	X	X	X	O
Back tracking	△	X	X	X	O
Tag Manufacturing Costs	O	X	X	X	O

Notation
O : Good △ : Middle X : Not Good

Data items for comparison of several protocols make reference to results shown in [7]. From security analysis and comparison result, we have shown that the proposed scheme has better performance in user data confidentiality, tag anonymity, Man-in-the-middle attack prevention, replay attack, forgery resistance and back tracking.

5. Digital Codec Design and Verification

5.1 Codec design

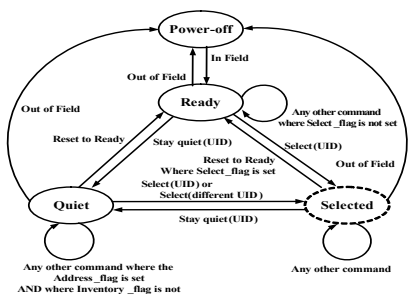
A tag is divided into two types of active tag and passive tag. The former uses a battery to transmit a signal to a reader, and the latter is powered by the electromagnetic field (radio wave) generated by the reader. In this paper, a passive tag is considered for realizing low-cost tags.

The RFID system is divided into two parts of analog front-end and digital parts. The analog front-end part is responsible for modulation and demodulation of data for the power supply of the tag and the digital part(Codec) handles control functions and data processing tasks [6,10].

The digital Codec for a Tag is composed of Packet Processor, CRC (Cycle Redundancy Check) Calculator, System Controller, SHA-1 hash algorithm, and EPROM [9].

Packet-processor filters the required data after analyzing commands of various packets received from RF/analog front-end. It is possible for reconstruction of the packed data which will be sent to the reader. CRC calculator block calculates CRC value on data for transmitting and receiving and it compares the CRC value with the received one for detecting errors during transmission [6,10]. It is also read some information from tag memory, EPROM.

The EPROM has been stored in unique information of tag's ID and key value K_1. SHA-1 hash algorithm [8] takes as input a message with a maximum length of less than 512 bits and produces as output 160-bit message digest(hashed data).



(Fig 3) A state transition diagram for Codec

Fig. 3 shows a state transition diagram for Codec design of Tag. The Codec designed for a tag has been described in Verilog HDL at the Behavioral level. Synthesis has been performed using Synopsys Design Compiler with 0.25 μm standard-cell library from Hynix Corp [11]. Table 5 shows synthesis results of the implemented Codec.

(Table 5) Synthesis results

Block names	Used area
ADDAEK	7,107
FFunc	7,257
SHA_St_gen	633
exchange	17,038
inReg	33,045
CMD_State	115
CONTROL_Block	42,157
MEM_Block	15,171
Out_Gen_Block	13,731
Out_Packet_Gen	16,341
Packet_Processor	64,080
Buf_REG	143,216
No. of transistors	370,000

5.2 Verification

In order to fully validate the designed Codec, timing simulation has been performed with Mentor Graphics' ModelSim. Table 6 shows initial vector for simulating EPROM of the Codec.

(Table 6) Initial vector for EPROM

UID	DSFID	Init Key
0xFFFFFFFFFFFFFFFF	0xCC	0xAAAA

Initial vector of the memory has three fields of UID of tag, DSFID (Data Storage Format Identifier), and Init Key for initial key value. DSFID indicates how the data is structured in the tag memory EPROM. In addition, three types of the packet data are used as input vectors for simulation, which is given in Table 7, 8 and 9.

(Table 7) Input vector for extended Inventory request format

SOF	Flags	Invent.	Opt. AFI	Mask length	Mask-value	S	CRC	EOF
	0x20	0x01	null	0x11	0xFFFFFFFF FFFFFFFF	0x4444	0x249C	

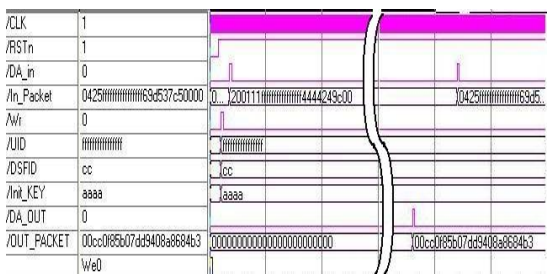
(Table 8) Input vector for extended Inventory response format

SOF	Flags	DSFID	HT	CRC	EOF
	0x00	0xCC	0x0F85B07D D9408A86	0x84B3	

(Table 9) Input vector for extended Select request format

SOF	Flags	Select	UID	K_2	CRC	EOF
	0x04	0x25	0xFFFFFFFFFFFFFFFF	0x69D5	0x37C5	

The packet data shown in Table 7 is used as input vector for simulating the extended Inventory request format. Two packets of Table 8 and 9 are also used as input vectors for the extended Inventory response packet format and the extended Select request format, respectively. Fig. 4 shows the result which performs timing simulation using initial vector and the data shown in Table 9.



(Fig 4) Timing simulation result for Codec

Input vectors used for simulation are as follows;
Input packet :
"0425ffffffffffffffffff69d537c500000"
UID : "ffffffffffffffff"
DSFID : "CC"
Init_Key : "AAAA"

Output result obtained by performing simulation is "00cc0f085b07dd9408a8684b3", and it is used as data for output packet.

From the simulation results, we showed that the proposed hash based protocol is correctly operated in environment which uses the Hash based RFID tag. The designed system operates at a clock frequency of 45 MHz with gate count of 35K. Depending on design tool used, the designed Codec has trade-off between the system performance and the hardware cost.

6. Conclusion

In this paper, we have presented the security enhancing of authentication protocol for Hash based RFID tag. The protocol is based on a three-way challenge response authentication technique between a RFID tag and a back-end server. In addition, the extended three types of protocol packets have been described for realizing a secure authentication mechanism by modifying the protocol defined in the ISO/IEC standard [5].

The designed system is described in Verilog HDL and synthesized using the Synopsys Design Compiler with Hynix 0.25 μm standard-cell library [11].

From the simulation results, we showed that the proposed hash based protocol is correctly operated in environment which uses the Hash based RFID tag. The designed system operates at a clock frequency of 45 MHz with gate count of 35K, and found that the proposed scheme is well suite to implement robust RFID system against active attacks such as the man-in-the-middle attack and the replay attack. From comparison and implementation results, we have shown that our scheme is a well-designed secure protocol which satisfies various security requirements in RFID system environment.

References

[1] Markus Jakobsson and David Pointcheval, "Mutual Authentication for Low-power Mobile Devices", Lecture Notes in Computer Science, Vol. 178 – 195, 2002.

[2] Stephen A. Weis, Sanjay E.Sarma, Ronald L. Rivest and Daiel W. Engels, "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems", First International Conference on Security in Pervasive Computing, 2003. http://theory.lcs.mit.edu/ sweis/spc-rfid.pdf

[3] S. E. Sarma, S. A. Weis, and D. W. Engels, "RFID System and Security and Privacy Implications", CHES 2002, Lecture Notes in Computer Science, Vol. 2523, pp. 454 – 469, 2003.

[4] M. Ohkubo, K. Suzki, S. Kinoshita, Cryptographic Approach to 'privacy friendly' tags, RFID Privacy Workshop, 2003.

[5] International Organization for Standardization, "ISO/IEC 18000-3, Information Technology AIDC Techniques - RFID for Item Management," March 2003.

[6] Martin Feldhofer, "A Proposal for an Authentication Protocol in a Security Layer for RFID Smart Tags", IEEE Proceedings of MELECON 2004, Vol. 2, pp. 759 – 762, 2004.

[7] J. Yang, K. Ren, and K. Kim, "Security and Privacy on Authentication Protocol for Low-cost

RFID", Proceedings of SCIS2005, pp. 25 - 28, Jan. 2005.
[8] D. Eastlake and P. Jones, "US Secure Hash Algorithm 1 (SHA-1)", Internet RFC 3174, September 2001.
[9] Weis, S., Sarma, S., Rivest, R., and Engels, D., "Security and Privacy Aspects of Low-Cost RFIDs", Security in Pervasive Computing, Lecture Notes in Computer Science, Vol. 2802, pp. 201 - 212, 2003.
[10] S. B. Ryu, J. O. Jeon, and M. S. Kang, "FPGA Design of Digital Codec for Passive RFID Tag", in ALPIT 2007, August 2007.
[11] Hynix Semiconductor Inc., Macrocell Cell DataBook: 0.25-Micron 2.5V Standard Cell Library HSM222A, Aug. 2001.

◐ 저 자 소 개 ◑

전 진 오(Jin-Oh Jeon)

1997 : B.S. degree. in Computer Engineering from Korea National Open Univ.
2001 : M.S. degree. in Computer Engineering from Kookmin Univ.
2005 : Ph.D. course(doctor´s course) at Anyang Univ. since 2005.
2006 ~ current : Dept. of Computer Engineering, Anyang University, Adjunct Professor
Research interests : mobile DRM system, Information Security, Multimedia Processing, etc.
E-mail : jazzman@anyang.ac.kr

강 민 섭(Min-Sup Kang)

1979 : B.S. degree from the Kwangwoon University, Seoul, Korea.
1984 : M.S. degree from the Hanyang University.
1992 : Ph.D. degree from the Osaka University, Japan.
1984 ~ 1993 : Electronics and Telecommunications Research Institute as a senior member of engineering staff. He also worked as a visiting scholar at the University of California, Irvine, in 2001. Since 1993, he has been with Dept. of Computer Engineering, Anyang University, Korea, as a professor and the dean of the College of Science and Engineering, Anyang University, from 2002 to 2003. and RFID/USN. Professor Kang is a member of IEEE, KIPS, KIISE, and IEEK.
Research interests : cipher processor design, signal processing, network security, etc.
E-mail : mskang@anyang.ac.kr