SCIS 2009 The 2009 Symposium on
Cryptography and Information Security
Otsu, Japan, Jan. 20-23, 2009
The Institute of Electronics,
Information and Communication Engineers

Desynchronization and Cloning resistant light-weight RFID authentication protocol using integer arithmetic for low-cost tags

Minhea Kwak [*] Jangseong Kim [*] Kwangjo Kim [*]

Abstract— The cost of the tag is one of the important factors to their proliferation. Designing a secure and efficient light-weight authentication protocol is imperative to resist against all feasible attacks. In general, the low-cost tag is difficult to implement the traditional public key cryptosystem since the tag's limited storage capacity(25-3K memory and 5-10K logic gates). Over the past years, several streams of research have emerged to resolve the RFID security and authentication issues. Most of the previous light-weight RFID authentication protocols based on random number generator, Cyclic Redundancy Code(CRC) or bitwise operations (e.g., XOR, AND and OR)are vulnerable to both passive and active attacks [21, 22]. For instance, anyone can obtain the tag identity and secret key through the consecutive eavesdropping.[9] In this paper, we propose a light-weight and secure authentication protocol that enhances Stephane *et al.*'s [19] protocol based on a random number generator and abstract of integer arithmetic (AIA), which generates secret key pool from the subset of the remainders and the carries of the integer multiplication. While requiring only 82 bit of RAM, 20 bit of ROM and 300-400 logic gates, our protocol can satisfy security requirements for RFID system. In conclusion, our protocol may be scaled to provide high level of security, using relatively little computational resources and be good alternative of the previous schemes based on bitwise operation.

Keywords: lightweight protocol, RFID, integer arithmetic, AIA.

1 Introduction

Radio Frequency Identification (RFID) refers to technologies and systems that use radio waves to transmit and uniquely identify objects. RFID transponder or tag, consisting of single chip and antenna, to identify and track the target object is involved in RFID technology and systems. The tag is classified into three types according to the ability of the power and the capacity: passive, semi-passive and active tags. The passive tag can only store 250-3K bit and 5K-10K logic gates which are used for security function. The system employing RFID tags used for various industries (e.g., distribution, logistics, medical attendance and education service) instead of barcodes is emerging one of the most pervasive computing technologies.

Although the advantages of the RFID(e.g., portable database, no line of sight, multiple tag read/write and traceability), RFID still has unsolved problems in security and privacy aspects. Since most existing RFID systems are not complete and leak information about the attached object, the adversary can track the object silently. Some common types of attacks on RFID system include eavesdropping, replay attack, man-in-the-middle attack, loss of data including denial of service (DoS) and message hijacking, skimming and forgery (including cloning), and physical attack. Many researchers proposed the RFID authentication schemes to address these security issues. The light-weight protocols refers to those protocols [5, 7, 8] that require random number generator and simple function like Cyclic Redundancy Code(CRC) checksum or simple bit-wise operation (e.g., XOR, AND and OR) on tags.

As low-cost RFID becomes more and more popular, designing a secure and efficient light-weight authentication protocol is imperative to resist against all feasible attacks. However, most of the previous light-weight RFID authentication protocols are vulnerable to active attacks; Some researchers reported the weakness on the previous light-weight and ultra light-weight schemes[21, 22].

The contribution of this paper is two-fold: (i) we present privacy and security vulnerabilities in the previous light-weight protocols identified by other researchers as well as new ones. (ii) we propose a light-weight and secure authentication protocol that improves Stephane *et al.*'s [9] protocol based on a random number generator and abstraction of integer arithmetic (AIA), which generates secret key pool from the subset of the remainders and the carries of the integer multiplication.

This paper is organized as follows: In Section 2, we introduce an algebraic structure which these term will usually be denoted as AIA, and illustrate how to represent each as a short binary string. Section 3 presents an overview of the previous light-weight solutions. Sec-

[*] Information and Communications University, Munji-dong, Yuseong-gu, Daejeon, 305-732 Korea, {minhea,withkals,kkj}@icu.ac.kr

tion 4 reviews Stephane *et al.*'s protocol and describes our mutual light-weight authentication algorithm suitable for low-cost RFID technology. Section 5 gives a security and performance analysis of our authentication protocol. Finally, concluding remarks appear in Section 6.

2 Abstraction of Integer Arithmetic

Stephane *et al.*[19] introduces a light-weight authentication protocol based on AIA concept. The specific multiplication of two integers can actually be viewed as a complex binary operation on strings of digits involving multiple iterations of two interlocking binary operations $(\otimes, \oplus)$ which acts on pairs of digits. If we consider the product of an n digit integer K and a p digit integer M in some unspecified base b. The result is labeled $E = e_{p+n}...e_2e_1$. Figure 1 shows detailed description of the integer multiplication.

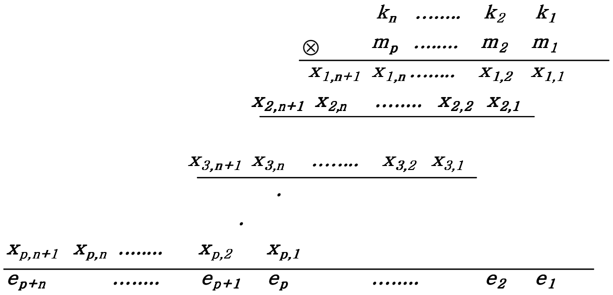
k_n .... k_2 k_1
$\otimes$ m_p .... m_2 m_1
$x_{1,n+1}$ $x_{1,n}$ .... $x_{1,2}$ $x_{1,1}$
$x_{2,n+1}$ $x_{2,n}$ .... $x_{2,2}$ $x_{2,1}$
$x_{3,n+1}$ $x_{3,n}$ .... $x_{3,2}$ $x_{3,1}$
$x_{p,n+1}$ $x_{p,n}$ .... $x_{p,2}$ $x_{p,1}$
e_{p+n} .... e_{p+1} e_p .... e_2 e_1

Figure 1: Regular Integer multiplication algorithm

We can then uses the regular steps commonly accepted for multiplying two integers 'by hand' to write each digit in the product of the string $k_n...k_2k_1$ and the digit m_i. For example,
$x_{i,1} = (k_1 \otimes m_i)_r$,
$x_{i,2} = ((k_2 \otimes m_i)_r \oplus (k_2 \otimes m_i)_c)_r$,
$x_{i,3} = ((k_3 \otimes m_i)_r \oplus ((k_2 \otimes m_i)_c \oplus ((k_2 \otimes m_i)r \oplus (k_1 \otimes m_i)_c)_c)_r)_r$.
We elucidate a number of interesting properties of integer multiplication [19]:
1. Both digit-wise addition, $\oplus$ and digit-wise multiplication, $\otimes$ are binary operations that map each pair of digits (with respect to a given base b) to another pair of digits, namely the remainder and carry.
2. The algorithm for multiplication of integers works independent to the choices of output for the operations $\oplus$ and $\otimes$. That is, for each of $\oplus$ and $\otimes$, if we change the output associated with one or more ordered pairs of digits, then the integer multiplication algorithm will still work but will produce different output strings.
3. Changing the outputs of $\oplus$ and $\otimes$ can alter the algebraic properties of the resulting string-wise multiplication.
Since given algorithm for basic arithmetic is common knowledge, in order to define a new string-wise multiplication, AIA would be list as a table format or an ordered string the remainders and carries associated with each ordered pair of digits for the $\oplus$ and $\otimes$ operations. The subsequent derived string, 0001020102100210110000 000000102000211, gives the remainders and carries for actual addition and multiplication in base 3.
Stephane *et al.* defines AIA as follows:

Definition 1 (Abstraction of Integer Arithmetic) *Let* B *be the set of all base-*b *strings of finite length. Then any base-*b *string,* s*, of length* $4b^2$ *defines a binary operation,* $\times_s$ *on* B *using the algorithm for regular integer multiplication but with the remainders and carries of digit-wise multiplication and addition taken from s as detailed above. We call the pair (*b,$\times_s$*) an abstraction of integer arithmetic, or* AIA *for short.*

3 Related work

We can classify the previous RFID authentication protocols into four types: Full-fledged, Simple, Light-weight and Ultra light-weight.
The protocols [1, 2] belonging to the full-fledged class support classical cryptographic functions like hashing, encryption, and even public key algorithms on tags. Juel *et al.*[1] raised concerns as to whether data on the chip embedded in an e-passport could be covertly collected by means of skimming or eavesdropping.
The tags in the protocols of the simple class should support random number function and hash functions but not encryption functions or public key algorithms. Examples are like [3, 4], where Molnar and Wagner [3] proposed a tree based scheme in which a tag contains not one symmetric key, but multiple keys in a hierarchical structure defined by the tree S. The basic idea in [4] is to modify the identifier each time so that the tag is recognized by authorized parties only. Avoine *et al.* [12] reported the replay attack and the unscalability of Ohkubo *et al.*'s scheme [4].
The third class called light-weight refers to those protocols [3, 4] that do not require hashing function on tags. Some researchers present the hash based protocol [10, 11] as the light-weight protocol, but current cryptographic hash functions is difficult to implemented on the passive tag. The EPC global also announces Class-1 Gen-2 RFID tag [20] containing Pseudo-Random Number Generator (PRNG) and Cyclic Redundancy Code (CRC)checksum but not hashing function. The protocols [13, 14] belong to this class, where Juels [13] proposed a challenge-response protocol using short pseudonym list in the tags. Chien and Chen [15] reported the DOS attack, replay attack, tracking attack and spoofing tag problem on the scheme [14], which based on simple XOR and matrix operations, where matrices M1 and M2 are stored on each tag and the reader as the shared secret key, designed an efficient tag identification and reader authentication scheme for GEN-2 RFID. The HB-series [16, 17, 18] can also be classified into this class, since they demand the support of random number function but not hash function on the tags. Hopper and Blum [16] first introduced the Human-computer protocol based on the learning parity with noise (LPN) problem. Later, the HB protocol was attacked and

improved by its sister works [17, 18]. Actually, the HB-series cannot be regarded as complete, since these protocols only consider the authentication on the tags. They neglected the security issues of the authentication of the readers, the tracking problem, the anonymity issue and even the privacy of the tag identification.

Recently, Peris-Lopez *et al.* proposed a series of ultra-lightweight authentication protocols [6, 7, 8] where the tags involve only simple bit-wise operations like XOR, AND, OR and addition mod 2^m. These schemes are very efficient that only require about 300-400 gates. Unfortunately, some researchers[21, 22] reported the desynchronization attack and the full-disclosure attack on these protocols and sister protocols . The previous schemes [6, 7, 8, 9] only provide weak authentication and weak integrity protection, which make them vulnerable to both passive and active attacks. Most of the light-weight and ultra lightweight protocol based on PRNG, CRC or bitwise operation are obivously efficient but has fundamental security flaws that the adversary can reveal the tag's identity and even the security key by consecutive eavesdropping.

4 Our Improved Protocol

4.1 Review on Stephane *et al.*'s scheme

Stephane *et al.* proposed a light-weight RFID authentication protocol using pseudonyms and integer arithmetic suitable for low-cost tags. Their protocol generates and exchanges messages using AIA each other until they convince that sharing a same secret key.

Initially, both the reader and the tag assume that have the same secret key (AIA, K, d). Then, they begin by choose $m_1 = d$ as initial base, and share $X = (K_d)'$ where the leftmost n digits of K_d.

In the mutual authentication phase, the reader randomly choosing m_2, calculating e_2 where right most bit of $(X +_{AIA} K_{m_2})$, and transmitting (m_2, e_2). The tag performs the same addition as the reader to verify the received message. If so, the tag updates the register. Otherwise, the tag fails to authenticate. The process should be repeated for a preset number of rounds at which time both parties will be convinced that they share the same secret key. Each time the i^{th} round looks like Table 1:

Table 1: i^th round of Stephane et al.'s protocol

Step 1	Receive m_{i-1}, e_{i-1}
Step 2	If $\{(X +_A K_{m_{i-1}})_1 \neq e_{i-1}\}$ then QUIT.
Step 3	Update $X \leftarrow (X +_{AIA} K_{mi} - 1)'$
Step 4	Generate $m_i \in \{1, 2, ..., b\}$
Step 5	Calculate $(X +_{AIA} K_{mi})$
Step 6	Transmit (m_i, e_i)
Step 7	Update $X \leftarrow (X +_{AIA} K_{mi})'$

A hard problem of Stephane *et al.*'s protocol is as difficult as uncovering the secret key (*K*, *AIA*, *i*). Brute force would require uncovering all string of $K_1, K_2, ..., K_b$ as well as b and AIA_i. Each AIA is designed to store unique logic gate so that the attacker wishing to clone the tag should to read the logic gate configuration on the tag and produce new tags with this same logic gate configuration in order to imitate the original tag. However, the adversary can guess one bit messages (m_i, e_i) with probability $\frac{1}{b^2}$. Moreover, the cost of computation grows heavier as the number of authentication routine increases; repeats authentication at least 40 times to avoid the brute force attack. We enhance Stephane *et al.*'s protocol in terms of performance as well as security.

4.2 Assumption and Notation

The following assumptions are made:

· The authentication will occur between a reader and a tag.

· The channel between the reader and the backend server is assumed to be secure, but that between the reader and the tag is susceptible to all the possible attacks.

· The reader will store all secret keys, each corresponding to a different RFID tag, and has infinite power.

· The tag will have a single secret key, *K*, in memory. The rest of the secret key, *AIA*, will be implemented as hardware, in the form of logic gates on the tag.

· The tag has a random number generator and can perform simple calculations provided the maximum allowable gate count to perform these calculations is not exceeded.

Notations for the protocol are summarized in Table 2:

Table 2: Notation

Item	Description
K	Secret key, $K = \{K_1, K_2, ..., K_n\}$
N_i	Random base, $N_i = \{m_p m_{p-1} ... m_2 m_1\}$
K_{mi}	$K \times_{AIA} m_i = \{t_{n+1} t_n ... t_2 t_1\}$
M_i	$K \times_{AIA} N_i = \{e_{n+p} ... e_n ... e_2\}$
AIA	$AIA = \{AIA_1, AIA_2, ..., AIA_n\}$
X	Register
X_i	i^{th} right-most digit of X
X'	Left-most *n-1* digits of X
M_{i-R}	Right half of M_i, $\{e_{(p+n)/2} ... e_2 e_1\}$
M_{i-L}	Left half of M_i, $\{e_{p+n} ... e_{1+(p+n)/2}\}$
flag	Session state,(normal:0, abnormal:1)

4.3 Description

We begin by sharing the same secret key (K, AIA) described above, and the reader and the tag participate in an message computation algorithm in Table 3 to generate a message that will be exchanged between two parties.

Our protocol consists of two part; tag identification and mutual authentication and key updating phase.

Tag identification : The reader sends *query* to the tag, which first responds with its AIA_i and a random base string N_1. If the reader could find a matched entry in the database, it steps into the mutual authentication

Table 3: Message Computation

Input	$K = k_n k_{n-1} ... k_2 k_1, N_i = m_p m_{p-1} ... m_2 m_1$
Output	$M_i = K \times_{AIA} N_i$
Step 1	For i=1 to p
Step 2	$t_1 \leftarrow (k_1 \otimes N_i)_r, carry \leftarrow (k_1 \otimes N_i)_c$
Step 3	For j=2 to n
Step 3a	$t_j \leftarrow ((k_j \otimes N_i)_r \oplus carry)_r$
Step 3b	$carry \leftarrow ((k_j \otimes N_i)_c \oplus ((k_j \otimes i)_r \oplus carry)_c)_r$
End For	
Step 4	$t_{n+1} \leftarrow carry$
Step 5	Output $K_{Ni} = t_{n+1} t_n ... t_2 t_1$
Step 6	$X \leftarrow (K_{m1})'$
Step 6a	$e_i \leftarrow (X +_{AIA} K_{mi})_1, X \leftarrow (X +_{AIA} K_{mi})'$
End For	
Step 7	$M_i = e_{n+p} ... e_n ... e_2$, stop

phase; otherwise, it probes again.

Mutual authentication and Key updating: Mutual authentication phase consider two cases as the state of the authentication session, authentication terminates normally or not.

We illustrate the normal operation procedure as in Figure 2:

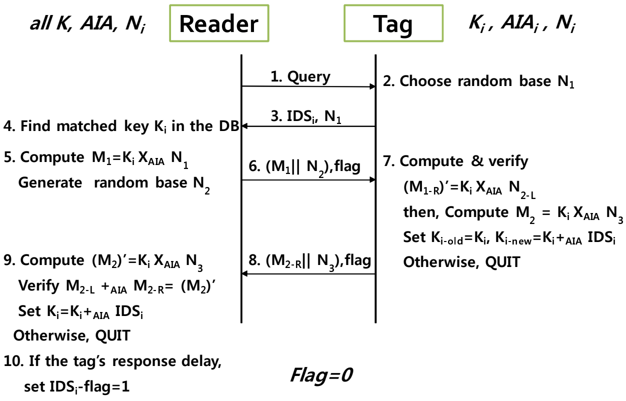
all K, AIA, N_i
Reader
Tag
K_i , AIA_i , N_i
1. Query
2. Choose random base N_1
3. IDS_i, N_1
4. Find matched key K_i in the DB
5. Compute M_1=K_i X_{AIA} N_1
Generate random base N_2
6. (M_1|| N_2),flag
7. Compute & verify
$(M_{1\text{-}R})'$=K_i X_{AIA} $N_{2\text{-}L}$
then, Compute M_2 = K_i X_{AIA} N_3
Set $K_{i\text{-}old}$=K_i, $K_{i\text{-}new}$=K_i+$_{AIA}$ IDS_i
Otherwise, QUIT
8. ($M_{2\text{-}R}$|| N_3),flag
9. Compute $(M_2)'$=K_i X_{AIA} N_3
Verify $M_{2\text{-}L}$ +$_{AIA}$ $M_{2\text{-}R}$= $(M_2)'$
Set K_i=K_i+$_{AIA}$ IDS_i
Otherwise, QUIT
10. If the tag's response delay,
set IDS_i-flag=1
Flag=0

Figure 2: Mutual Authentication I

1. The reader sends M_1, a new random base N_2 and *flag* after computing a message M_1 using AIA_i and the random base N_1 received from the tag.
2. The tag performs the same addition as the reader to verify the reader's message. If so, the tag calculates the next message, M_{2-R}, by randomly choosing N_3. Otherwise, the reader fails to authenticate.
3. Then, the tag updates the current secret key as $K_{i-old} = K_i$ and $K_{i-new} = K_i +_{AIA} IDS_i$, transmitting $(M_{2-R}||N_3)$ and *flag* to the reader.
4. After the tag authenticates the reader, the reader verifies the message M_{2-R} to convince that the tag received the message M_1 correctly. Finally, the reader updates the secret key as $K_i = K_i +_{AIA} IDS_i$.

When the authentication over abnormally, each process looks like Figure 3:

If the last message $(M_{2-R}||N_3)$ in session 9 is interrupted by network disconnection or the adversary, key updating can lead to desynchronization in DB between the tag and the reader, the tag updates the secret key but not the reader. We consider the abnormal situation as follows:

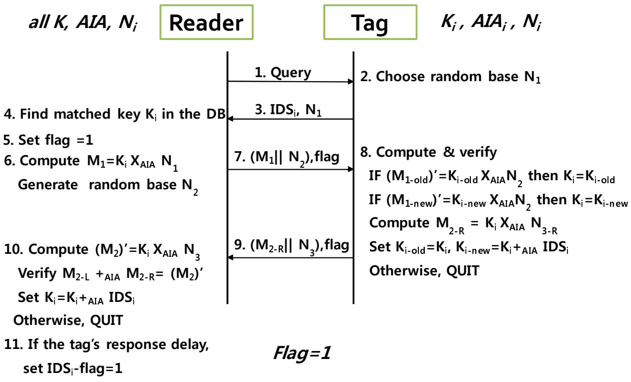
all K, AIA, N_i
Reader
Tag
K_i , AIA_i , N_i
1. Query
2. Choose random base N_1
3. IDS_i, N_1
4. Find matched key K_i in the DB
5. Set flag =1
6. Compute M_1=K_i X_{AIA} N_1
Generate random base N_2
7. (M_1|| N_2),flag
8. Compute & verify
IF $(M_{1\text{-}old})'$=$K_{i\text{-}old}$ $X_{AIA}$$N_2$ then K_i=$K_{i\text{-}old}$
IF $(M_{1\text{-}new})'$=$K_{i\text{-}new}$ $X_{AIA}$$N_2$ then K_i=$K_{i\text{-}new}$
Compute $M_{2\text{-}R}$ = K_i X_{AIA} $N_{3\text{-}R}$
Set $K_{i\text{-}old}$=K_i, $K_{i\text{-}new}$=K_i+$_{AIA}$ IDS_i
Otherwise, QUIT
9. ($M_{2\text{-}R}$|| N_3),flag
10. Compute $(M_2)'$=K_i X_{AIA} N_3
Verify $M_{2\text{-}L}$ +$_{AIA}$ $M_{2\text{-}R}$= $(M_2)'$
Set K_i=K_i+$_{AIA}$ IDS_i
Otherwise, QUIT
11. If the tag's response delay,
set IDS_i-flag=1
Flag=1

Figure 3: Mutual Authentication II

1. The reader initiates *flag* as 1.
2. The reader sends M_1, a new random base N_2 and $flag$ after computing a message M_1 using AIA_i and the random base N_1.
3. When *flag* is 1, the tag performs different additions using old key and new key to reset the secret key K_i. The tag checks whether received message M_1 corresponds with M_{1-old} or M_{1-new}.
4. The tag initiates the next step by randomly choosing N_3, calculating right half message M_{2-R}. Then, the tag updates the current secret key as $K_{i-old} = K_i$ and $K_{i-new} = K_i +_{AIA} IDS_i$, transmitting message $(M_{2-R}||N_3)$ to the reader.
5. After the tag authenticates the reader, the reader combines the received message M_{2-R} and their computing message M_{2-L} to verify that the tag received the message M_1 correctly and they are sharing same secret K_i and AIA. Finally, the reader updates the secret key as $K_i = K_i +_{AIA} IDS_i$.

5 Security and Performance Analysis

5.1 Security Analysis

Every tag is designed with a unique set of logic gates to perform the authentication. In this instance the attacker does not know any portion of (K, AIA, N). Brute force would then require uncovering all K_i and N_i as well as the table values for each of AIA_i. The attacker should try b^{nb} guesses for K_i, $3b^p$ guesses for N_1, N_2, N_3 and $((4b^2(b!)^b)^n$ guesses for all AIA for a total of $b^{nb} + ((4b^2(b!)^b)^n + 3b^p$ to uncover all secret. We believe that this hard problem is as difficult as uncovering (K, AIA, N). In so doing the following security properties appear to be satisfied.

- Man in the middle attack prevention : Even if the adversary sends flipped message $(M_i'||N_i')$, both parties should verify the messages with their unique AIA_i so that each round of the protocol prevents a man in the middle attack.
- Resistance to Cloning Attacks : Even if the secret string AIA is lifted from the tag, the attacker would

need to read the logic gate configuration on the tag and produce new tags with this same logic gate configuration in order to imitate the original tag.

• Forward Security : As the secret string K is stored in memory, periodically, once authentication is successful the tags secret string could be updated.

• Replay attack prevention : Storing all messages from communication between the tag and the reader, and replaying them to the appropriate device will not work because both parties newly generate the message M_i' with their AIA_i.

• Synchronization: Setting up the session state, $flag$, to 0 or 1 as the condition of the authentication session. When the authentication terminates abnormally, the tag resets the secret key, as K was before.

The comparisons with other light-weight schemes are summarized in Table 4 .

Table 4: Security Comparisons

Item	[15]	[18]	[14]	[19]	Our Protocol
Privacy	O	O	O	O	O
Anonymity	O	O	X	O	O
Resist to replay attack	O	X	X	O	O
Resistance to man in the middle attack	O	X	X	X	O
Resistance to Cloning	X	X	X	X	O
Synchronization	X	X	X	X	O

O : Provided , X : Not provided

5.2 Performance Analysis

We compare our protocol with Stephane *et al.*' scheme in terms of the storage, computation and communication requirements from the reader and the tag sides. Table 5 gives the storage and the memory required in the reader and the tag.

The most severe restrictions of the passive tag are the small number of logic gates(200-2000) which can be devoted to security algorithms, and the volatile memory available(32-128 bit) to store intermediate calculations. The implementation of the standard private key cryptosystem, AES (Advanced Encryption Standard), currently requires approximately 4000 logic gates. EPC Class-1 Gen-2 sample tag allows only 128-512 bit of ROM, 32-128 bit of RAM and 1000-10000 gates.

Our protocol requires $2(n+p)\log_2(b)+2$ bit for K_i,N_i, M_i and *flag*, and $(4b^2-b)\log_2(b!)$ bit of ROM to store AIA for the tag where each K_i is n digits long, N_i is p digits long and random base is b. The reader is required to store the tag's all AIA_i consisting of $4b^2-b$ additive carry bit and the $b!$ possible permutations so that the reader side needs $N(4b^2-b)\log_2(b!)$ bit and $N(2(n+p)\log_2(b)+2)$ bit, AIA and K, N respectively. For example if we choose $b=4$, $n=10$, $p=10$ the tag will require 20 bit of ROM, 82 bit of RAM, and 300-400 logic to store (AIA, K, N_i). Our protocol seems efficient enough to satisfy the EPC Class-1 Gen-2 specification.

In order to compare the computational cost of the two protocols, we take into account the AIA algorithm that involves bit-wise multiplication and addition in each authentication session. Table 5 shows our protocol needs the small number the bit-wise multiplication($\frac{3}{2}(np)$) than Stephane *et al.*'s one [19](nr).

Moreover, while Stephane *et al.*'s protocol repeats at least 40 times of authentication round to guarantee reasonable security, our protocol only need 3 times of authentication session. Thus our protocol has practical performance advantages over the Stephane *et al.*'s scheme, while also providing the privacy and security properties.

6 Conclusion

In this paper, we have reviewed the security flaws of the previous light-weight protocol based on bitwise operations or CRC reported by other researchers as well as new ones. Then, as if the Abstraction Integer Arithmetic(AIA), key pool with a unique subset of the remainders and carries of the integer for each tag, proposed by Stephane *et al.*[19]. we enhance efficiency as well as security of Stephane *et al.*'s protocol. While requiring only 82 bit of RAM, 20 bit of ROM and 300-400 logic gates, our protocol can satisfy security requirements(*e.g.*, synchronization, protection to replay, cloning and impersonation)for RFID system. Our protocol may be scaled to provide a high level of security, using relatively little computational resources and be an alternative of the previous schemes based on bitwise operation.

Acknowledgement

The work presented in this paper was supported in part by IT R&D Program of Ministry of Information and Communication(MIC)/ Institute for Information Technology Advancement(IITA)2005-S-106-02, "Development of Sensor Tag and Sensor Node Technologies for RFID/USN"

References

[1] A. Juels, D. Molnar, and D. Wagner, Security and privacy issues in e-passports, IEEE/Create Net Secure Commun., 2005.

[2] S. Kinoshita, M. Ohkubo, F. Hoshino, G. Morohashi, O. Shionoiri, and A.Kanai, Privacy Enhanced Active RFID tag, International Workshop on Exploiting Context Histories in Smart Environments, May 2005.

[3] D. Molnar and D. Wagner,Privacy and security in library RFID: Issues, practices and architectures, Conference on Computer and Communications Security-CCS'04, pp. 210..219, 2004.

[4] M. Ohkubo, K. Suzki and S. Kinoshita,Cryptographic Approach to 'privacy-

Table 5: Performance Comparisons

Feature		Stephane [19]		Our Protocol	
		Tag	Reader	Tag	Reader
Capacity (bit)	RAM	$(n+1)\log_2(b)$	$n((4b^2-b)b\log_2(b!))$	$2(n+p)\log_2(b)+2$	$N(2(n+p)\log_2(b)+2)$
	ROM	$b(n+1)\log_2(b)+2$	$b(n+1)\log_2(b)+2$	$(4b^2-b)\log_2(b!)$	$N(4b^2-b)\log_2(b!)$
Computation (times)	$\oplus$	$(n+1)r$	$(n+1)r$	$(n+1)(p-1)+\frac{np}{2}$	$2(n+1)(p-1)$
	$\otimes$	nr	nr	$\frac{3}{2}(np)$	$2np$
Communication		$2r$		3	

N:the number of tags, b:random base, n:the bit-length of secret key K_i
r:the number of authentication session round, p: the bit-length of a random base string N_i

Friendly' Tags,in RFID Privacy Workshop, 2003.

[5] A. Juels. Minimalist cryptography for low-cost RFID tags, In C. Blundo and S. Cimato, editors, Security in Communication Networks (SCN 04), pages 149.164. Springer-Verlag, 2004. LNCS no. 3352.

[6] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda, "EMAP: An Efficient Mutual Authentication Protocol for Low-Cost RFID Tags, Proc. OTM Federated Conf. and Workshop: IS Workshop,Nov. 2006.

[7] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A.Ribagorda,LMAP: A Real Lightweight Mutual Authentication Protocol Low-cost RFID tags, in: Proc. of 2nd Workshop on RFID Security, July 2006.

[8] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A.Ribagorda, M2AP: A Minimalist Mutual-Authentication Protocol for Low-cost RFID Tagsin,in: Proc. of International Conference on Ubiquitous Intelligence and Computing UIC'06, LNCS 4159, pp. 912-923, Springer, 2006.

[9] H.Y. Chien. SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity, IEEE Transactions on Dependable and Secure Computing 4(4):337-340. Oct.-Dec. 2007.

[10] E.Y. Choi, S.M. Lee, and D.H. Lee,Efficient RFID authentication protocol for ubiquitous computing environment, In Proc. of SECUBIQ05, LNCS, 2005.

[11] I. Vajda and L. Buttyan, Lightweight authentication protocols for low-cost RFID tags, in Proc. 2nd Workshop on Security in Ubiquitous Comput., 2003.

[12] G. Avoine, E. Dysli, P. Oechslin, Reducing time complexity in RFID systems, The 12th Annual Workshop on Selected Areas in Cryptography (SAC), 2005.

[13] A. Juels, Strengthening EPC Tag against Cloning, in ACM Workshop on Wireless Security (WiSe), pp.67-76. 2005.

[14] S. Karthikeyan and M. Nesterenko, RFID security without extensive cryptography, in Proceedings of the 3rd ACM workshop on Security of ad hoc and sensor networks, pp. 63-67, 2005

[15] H.Y. Chien and C.H. Chen, Mutual Authentication Protocol for RFID Conforming to EPC Class 1 Generation 2 Standards, in Computers Standards and Interfaces 29(2), pp 254-259, 2007.

[16] N. J. Hopper and M. Blum, Secure Human Identification Protocols, in Proc. Seventh Int Conf. Theory and Application of Cryptology and Information Security, pp. 52-66, 2001.

[17] A. Juels and S.A. Weis, Authenticating Pervasive Devices with Human Protocols, in Proc of CRYPTO '05, pp. 293-308, 2005.

[18] J. Munilla and A. Peinado, HB-MP: A further step in the HB-family of lightweight authentication protocols, Computer Networks, 51(9):2262-2267, 2007.

[19] L. Stephane and T. L. Adrian, Clone resistant mutual authentication for low-cost RFID technology, IACR Eprint, 2007.

[20] EPCglobal, http://www.epcglobalinc.org/.

[21] T. Li and R. H. Deng, Vulnerability Analysis of EMAP-An Efficient RFID Mutual Authentication Protocol, The Second International Conference on Availability, Reliability and Security (AReS 2007), Vienna, 2007.

[22] T. Li and G. Wang, Security Analysis of Two Bultra-lightweight RFID Authentication Protocols, IFIP SEC 2007, May 2007