A Lightweight RFID Protocol Using Substring

Hung-Yu Chien and Chen-Wei Huang

Dept. of Information Management, National Chi Nan University, Taiwan, R.O.C.
{hychien, s95213508}@ncnu.edu.tw

Abstract. As low-cost RFIDs with limited resources will dominate most of the RFID market, it is imperative to design lightweight RFID authentication protocols for these low-cost RFIDs. However, most of existing RFID authentication protocols either suffer from some security weaknesses or require costly operations that are not available on low-cost tags. In this paper, we analyze the security vulnerabilities of a lightweight authentication protocol recently proposed by Li et al. [4], and then propose a new lightweight protocol to improve the security.

Keywords: RFID, authentication, low-cost cryptography, tracing, reader, DOS attack.

1 Introduction

A Radio Frequency Identification (RFID) system mainly consists of three components: radio frequency tags, readers, and a backend server/database (or a set of distributed databases) which maintains information on the tagged objects. Generally, the tag consists of a microchip with some data storage and an antenna. A reader queries tags to obtain tag contents though wireless communications.

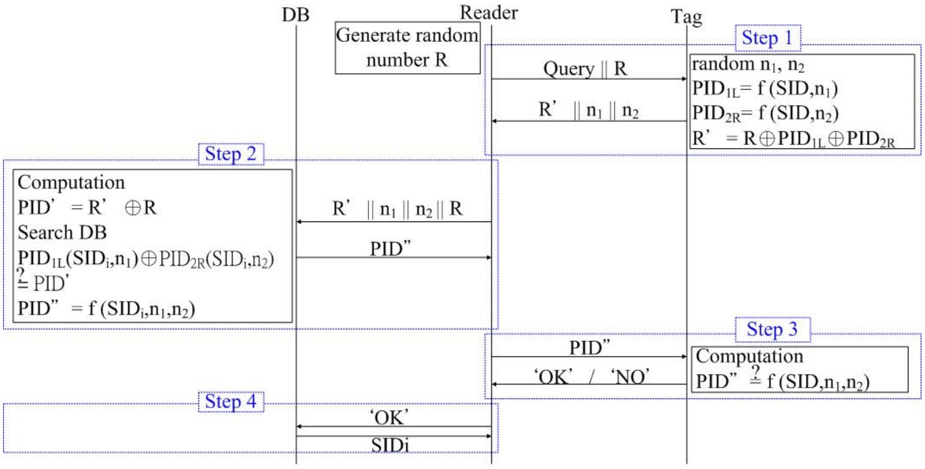
Recently, the wide deployment of RFID systems in a variety of applications has raised many concerns about the privacy and the security. An RFID tag can be attached to a product, an animal, or a person for the purpose of identification using radio waves. For any possible reasons, an adversary may perform various attacks such as eavesdropping, traffic analysis, spoofing, disabling the service, or disclosing sensitive information of tags, and hence infringes people’s privacy and security.

Even though RFID tags with full-fledged capacity are available, to attain great market penetration, RFID tags should be low-cost, which limit the computation power, the storage space, the communication capacity and the gates count. As studied by the previous work like [2], a low-cost RFID tag has approximately 4,000 logic gates. Although there have been many works devoted to design security mechanisms for low-cost RFIDs, most of these works require the tags to be equipped with costly operations such as one-way hashing functions [1, 3, 5], which are still un-available on low-cost tags. Contrary to these works, the schemes [4, 6, 8, 9] do not require the support of hashing functions on tags. However, the schemes [6, 8, 9] have been reported to show some security weaknesses [6, 7]. Recently, Li et al. [4], based on only bitwise XOR ($\oplus$), the Partial ID concept and pseudo random numbers, proposed a lightweight RFID authentication protocol for low-cost RFIDs. Different from most of existing solutions like [1, 3, 5] which used conventional cryptographic primitives (encryptions, hashing, etc), this protocol only used simple operations like XOR and substring. Unfortunately, we find that Li et al.'s scheme has several security weaknesses. In this paper, we shall analyze the security weaknesses of Li et al.'s RFID authentication protocol. To heal the weaknesses while preserving the lightweight feature, we propose a new RFID authentication protocol.

The rest of this paper is organized as follows. Section 2 reviews Li et al. lightweight RFID authentication protocol. Section 3 analyzes the vulnerabilities of Li et al.'s scheme. Section 4 proposes a new RFID authentication protocol that heals the security weaknesses while preserving the lightweight feature for low-cost RFID tags. Section 5 analyzes the security of our proposed protocol. Finally, conclusion remarks and future work are drawn in Section 6.

2 Review of Li et al.'s scheme

The tags in Li et al.'s RFID authentication protocol [4] use only bitwise XOR (⊕), the partial ID concept and pseudo random numbers. Costly operations such as multiplications and hash functions are eliminated in the design. In Li et al.'s scheme, each tag and the backend server share an l-bit secret information, *SID* (the secure ID). During the authentication the tag generates two random numbers n_1 and n_2 such that $2l \geqq n_1+n_2 \geqq l/2$. The two random numbers are used in the substring function f to extract the partial *ID*s, PID_{1L} and PID_{2R}, where PID_{1L} denotes the left substring of *SID* and PID_{2R} denotes the right substring of *SID*. That is, let $f(SID, i, j)$ denotes the substring of *SID* starting from position i to position j, then $PID_{1L} = f(SID, 1, n_1)$ and $PID_{2R} = f(SID, n_2, l)$. Li et al.'s scheme is depicted in figure 1.



SID: Secure ID
PID: Partial ID
f : substring function

DB: Backend Database/server
⊕: XOR operation

Fig. 1. Li et al.'s scheme

The scheme consists of four stages: the *PID* generating stage, the *SID* searching and tag authentication stage, the reader authentication stage and the result returning stage.

- ***PID generating stage*****:** The reader generates a random number *R*, and then sends it to the tag. Upon receiving the probe from the reader, the tag uses two random numbers n_1, n_2 and the substring function *f* to compute $PID_{IL}= f(SID, 1, n_1)$, $PID_{2R} = f(SID, n_2, l)$ and $R' = R \oplus PID_{IL} \oplus PID_{2R}$. The tag then responds the data *R'*, n_1 and n_2 to the reader.
- ***SID searching and tag authentication*** **stage:** The reader sends *R'*, *R*, n_1 and n_2 to the server. The server computes $PID' = R' \oplus R$, and iteratively picks up one candidate *SID'* from the database to check whether $PID'_{IL} \oplus PID'_{2R} = PID'$, where $PID'_{IL} = f(SID', 1, n_1)$ and $PID'_{2R} = f(SID', n_2, l)$. If a match is found, then the selected *SID*' is the tag's identification; otherwise, it continues the process until a match is found or responds with "failure" if no match could be found in the whole database. If a match is found, it computes $PID'' = f\,(SID', n_1, n_2)$ and then sends it to the reader.
- ***Reader authentication*** **stage:** The reader sends *PID"* to the tag, which then checks whether $f\,(SID, n_1, n_2)$ equals *PID"* to authenticate the reader. After the reader is authenticated successfully, the tag sends 'OK' to the reader; otherwise, it responds with "no find" information.
- ***Result returning*** **stage:** If the reader receives 'OK', and then sends it to the server, which will transmit the *SID* to the reader. Otherwise the reader stops the protocol.

3 Vulnerabilities of Li et al. scheme

In this section, we remark that Li et al.'s scheme is vulnerable to replay attack and is prone to reveal the secret information *SID*.

3.1 The replay attack

An adversary can easily eavesdrop on the communications from a legal tag, modify the data, and then replay the messages to masquerade as the legal tag as follows. The attack consists of two stages- the data deriving stage and the spoofing stage.

- **The data deriving stage**: The adversary records the communication (R, R', n_1, n_2) from a tag (say T_a), and then derives $PID_{IL} \oplus PID_{2\mathrm{R}}$ from $R' \oplus R$.
- **The spoofing stage**: In this stage, the adversary uses the derived data $PID_{IL} \oplus PID_{2\mathrm{R}}$ to masquerade as the tag T_a as follows.
  1. Upon receiving the probe Query$\|\overline{R}$ from the reader, the adversary computes $R' = PID_{1L} \oplus PID_{2R} \oplus \overline{R}$, and responds with $R' \| n_1 \| n_2$ to the reader.

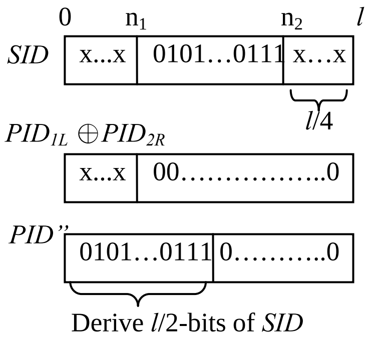
2. It is easy to see that the forged data $R' \| n_1 \| n_2$ will be accepted by the server, and the reader will forward the data PID'' from the server to the adversary.
3. The adversary just records the data PID'' and always responds with “OK” to the reader. We can see that the reader finally accepts this spoofing tag as the genuine tag T_a.

3.2 Disclosing the secret value *SID*

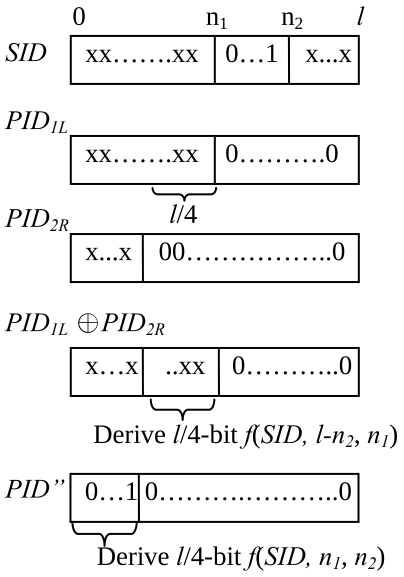
Since an adversary can eavesdrop on the communications and record the data R', R, n_1, n_2 and PID'', and he can compute $R' \oplus R$ to obtain $PID_{1L} \oplus PID_{2R}$. With n_1, n_2, $R' \oplus R$ and PID'', an adversary can derive partial information of *SID*, and can repeatedly run the process many times to fully disclose all the bits of *SID* or derive partial information of *SID* (if most bits of the identification are known, then it is highly possible to guess the rest bits because the identification of a tag- for example, the EPC code- has a pre-defined format). In the following, we describe the single run of our attack process, and examine some cases to point out the vulnerabilities of Li et al.’s scheme.

As the lengths of PID_{1L}, PID_{2R} and PID'' are unequal to l bits, we assume that 0s are padded to them such that each length of them equals l-bit in the following scenario (we can also assume 1s are padded to these strings, and the same attack still works). Based on the values $R' \oplus R = PID_{1L} \oplus PID_{2R}$ and $PID'' = f\,(SID, n_1, n_2)$, an adversary can derive parts of *SID.* The length of the disclosed part of *SID* depends on the values of n_1 and n_2. With $2l \geqq n_1 + n_2 \geqq l/2$ property, the values of n_1 and n_2 generally have four situations. Firstly, if $n_1 = l - n_2$, an adversary can derive $f(SID, n_1, n_2)$. Secondly, if $n_1 > l - n_2$, an adversary can derive $f(SID, n_1, n_2)$ and $f(SID, l-n_2, n_1)$. Thirdly, if $n_1 < l - n_2$, an adversary can derive $f(SID, n_1, n_2)$ and $f(SID, n_1+n_2, l)$. Finally, if $n_1 = l$, $l - n_2 = 0$, an adversary can obtain all of *SID*. Some example cases are discussed as follows.

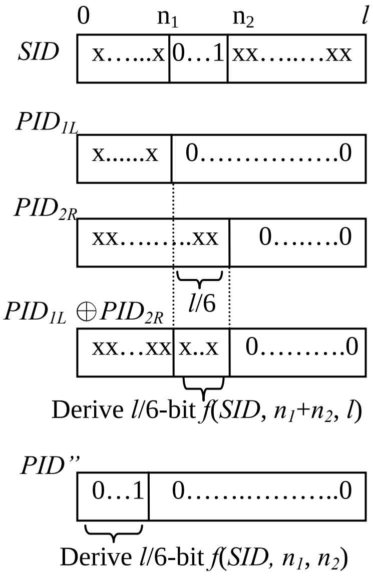
Example 1: Deriving parts of *SID*. Assume $n_1 = l/4$ and $n_2 = 3/4\, l$, an adversary can directly derive the $l/2$-bits $f(SID, n_1, n_2)$ from PID''.



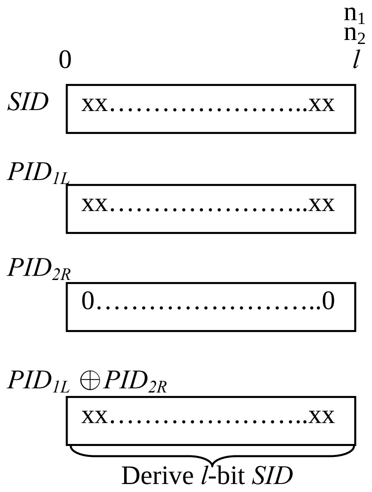
Example 2: Deriving parts of *PID_{1L}*. Assume n_1 = l/2 and n_2 = 3/4 l, an adversary can derive l/4-bit $f(SID, l\text{-}n_2, n_1)$ from $PID_{1L} \oplus PID_{2R}$, and derive l/4-bit $f(SID, n_1, n_2)$ from *PID"* as follows.



Example 3: Deriving parts of *PID_{2R}*. Assume n_1 =1/3 l and n_2 = l/2, an adversary derives l/6-bit $f(SID, n_1{+}n_2, l)$ from $PID_{1L} \oplus PID_{2R}$, and derive l/6-bit $f(SID, n_1, n_2)$ from *PID"* as follows.



Example 4: Deriving all the bits of *SID*. Assume $n_1 = l$ and $n_2 = l$, an adversary can obtain all the bits of *SID* as follows.



In a single run of the above attack, an adversary can derive partial information of *SID*, and he can launch the above attack several times to aggregate the partial information of *SID* or even derive all the bits of *SID*. Although it is possible that part of the *SID* can not be directly derived in the above process, one might guess the rest bits, because the identifications of tags are usually with fixed format.

4 A new lightweight RFID authentication protocol

In this section, we propose a new protocol to improve the security while preserving the lightweight property. Our proposed protocol is depicted in figure 2 and described as follows.

We assume that each tag and the database share an *l*-bit secret key x, $x = x_0x_1.....x_{l-1}x_l$. The reader generates a random number R_1, and the tag generates a random number R_2. Some notations are introduced as follows.

$g(z)$: $g()$ is a random number generator, and z is an input number.
$\tilde{g}$: the random output of $g(z)$.
rotate(p, w): *rotate* denotes the bitwise left rotation operator, and the operand p is rotated w positions.
Left(s): the left half of s.
Right(s): the right half of s.

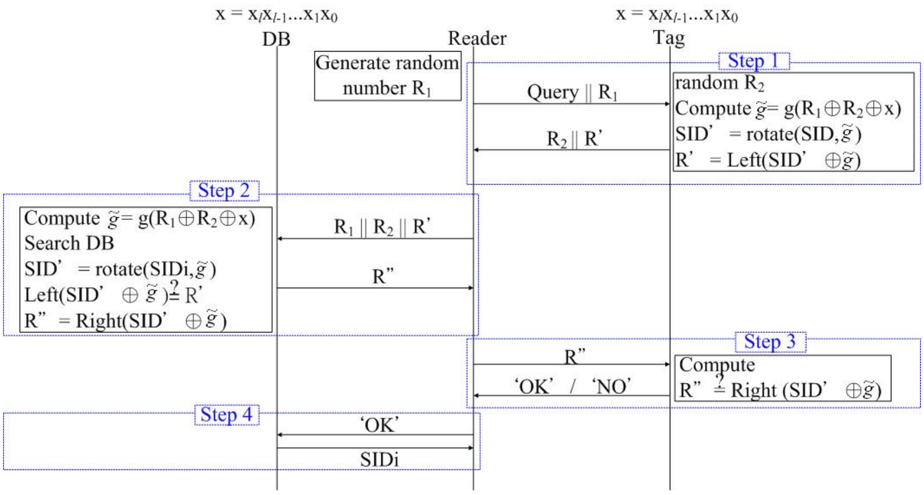


Fig. 2. Our proposed protoco1

Step1: The reader generates a random number R_1, and then sends it to the tag. Upon receiving the probe from the reader, the tag generates another random number R_2, computes $\tilde{g} = g(R_1 \oplus R_2 \oplus x)$ and rotates its *SID* to obtain *SID’* = *rotate*(*SID*, $\tilde{g}$). It calculates *R’*=*Left*(*SID’* $\oplus \tilde{g}$), and responds the data *R’* and R_2 to the reader.

Step2: The reader forwards R_1, R_2, and *R’* to the server. The server iteratively picks up one candidate *SID* from the database, computes $\tilde{g} = g(R_1 \oplus R_2 \oplus x)$ and *SID’* = *rotate*(*SID*, $\tilde{g}$) and checks whether *Left*(*SID’* $\oplus \tilde{g}$) = *R’*. If a match is found, then the selected *SID* is taken as the tag identification; otherwise, it continues the process until a match is found or responds with “failure’” if it cannot find a match in the whole database. If a match is found, it computes *R”*=*Right*(*SID’* $\oplus \tilde{g}$) and then sends it to the reader.

Step3: The reader sends *R”* to the tag, which then checks whether *Right*(*SID’* $\oplus \tilde{g}$) equals *R”* to authenticate the reader. After the reader is authenticated successfully, the tag sends ’OK’ to the reader; otherwise, it responds with “no find” information.

Step4: If the reader receives ‘OK’, and then sends it to the server, which then transmits the *SID* to the reader. Otherwise the reader stops the protocol.

During singulation, if multiple tags respond simultaneously to a query, they will interfere with each other. Therefore, we suggest that an anti-collision algorithm like the binary tree-walking [3] could be used in our proposed protocol to solve the problem of collisions.

5 Analysis

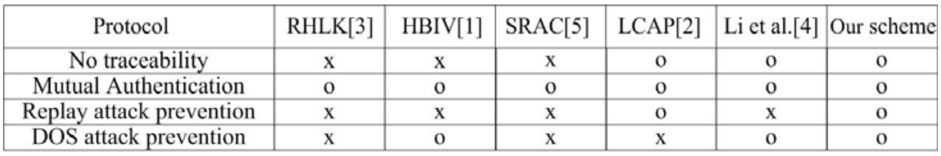
5.1 Security analysis

We now analyze the security of the proposed scheme as follows.

- **No traceability.** During each authentication instance, an adversary can only observe the values (R_1, R_2, R', R''), where R_1, R_2 are random numbers and R' / R'' are respectively the left/right half bits of the random string $SID' \oplus \tilde{g}$. No identity-related information can be derived from these values, and these values are distinct and look random to an adversary. So, an adversary cannot trace the tags.
- **Mutual authentication.** The server authenticates the tag by verifying the substring *Left*(*SID'* $\oplus \tilde{g}$), and the tag authenticates the server by verifying the substring *Right*(*SID'* $\oplus \tilde{g}$). Since only the genuine tag and the server who have the secret key *x* can generate and verify the values, the scheme provides mutual authentication.
- **Replay attack prevention.** An adversary could eavesdrop on the communications between the reader and the tag. However, the substring *Left*(*SID'* $\oplus \tilde{g}$) and the substring *Right*(*SID'* $\oplus \tilde{g}$) should depend on the random challenges R_1, R_2 , and replay messages cannot satisfy the verification either by the reader or by the tag.
- **DOS attack prevention.** In some previous schemes, the technique of varying pseudonyms is used to resist tracing, and these schemes need to synchronize the pseudonyms between the server and the tags; otherwise, they are unable to authenticate each other. In our scheme, there is no requirement of state synchronization. Therefore, it can resist the DOS attack.

In *Table 1*, we show a comparison of the security with previous mentioned schemes [1, 2, 3, 5].

Table 1. Comparison between schemes



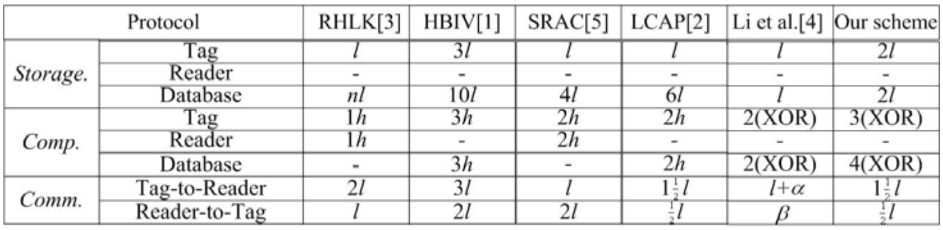
Protocol	RHLK[3]	HBIV[1]	SRAC[5]	LCAP[2]	Li et al.[4]	Our scheme
No traceability	x	x	x	o	o	o
Mutual Authentication	o	o	o	o	o	o
Replay attack prevention	x	x	x	o	x	o
DOS attack prevention	x	o	x	x	o	o

5.2 Performance analysis

It is important to minimize the storage cost, the computational cost and the communication cost of low-cost tags. In *Table 2*, we examine the performance of our scheme in terms of storage space, computational cost and communication cost, and compare it with the previous schemes [1, 2, 3, 5].

- **Storage space**: In our scheme, the tag has to store its tag *ID* of length *l* and an *l*-bit secret key. For identifying the tag, the database has also to store related information. Therefore, implementation of our scheme, the tag and the database both only require 2*l* bits of memory, which is suitable to low-cost tags.
- **Computational cost**: In Henrici-Müller’s scheme [1], Lee et al.’s scheme [2], Weis et al.’s scheme [3] and Lee-Verbauwhede’s scheme [5], the tag has to be equipped with hash functions, which are still un-available on low-cost tags. On the contrary, in our scheme, the tag only needs random number generation, XOR, shifting, and substring function. The computations are very efficient and lightweight.
- **Communication cost**: In our scheme, messages of tag-to-reader communication are R_2 and *R’* with a total of $1\frac{1}{2}l$ bits and a message of reader-to-tag communication is *R”* with $\frac{1}{2}l$ bits. Compared to Henrici-Müller’s scheme [1], Weis et al.’s scheme [3] and Lee-Verbauwhede’s scheme [5], the communication performance of our scheme is more efficient.

Table 2. Performance analysis



Protocol		RHLK[3]	HBIV[1]	SRAC[5]	LCAP[2]	Li et al.[4]	Our scheme
Storage.	Tag	l	$3l$	l	l	l	$2l$
	Reader	-	-	-	-	-	-
	Database	nl	$10l$	$4l$	$6l$	l	$2l$
Comp.	Tag	$1h$	$3h$	$2h$	$2h$	2(XOR)	3(XOR)
	Reader	$1h$	-	$2h$	-	-	-
	Database	-	$3h$	-	$2h$	2(XOR)	4(XOR)
Comm.	Tag-to-Reader	$2l$	$3l$	l	$1\frac{1}{2}l$	$l+\alpha$	$1\frac{1}{2}l$
	Reader-to-Tag	l	$2l$	$2l$	$\frac{1}{2}l$	β	$\frac{1}{2}l$

Notations of Table: *l* – size of required memory, *h* – the cost of a hash function operation, *α* – between *l* /2 and 2*l*, *β* – less than *l*

6 Conclusion and future work

This paper has shown the replay attack and the secret disclosure problem of Li et al.’s scheme. In the attack, an adversary can easily derive partial information of the secret *SID* or even all the bits of the *SID*. We also have proposed a new lightweight RFID authentication protocol, which improves the security, the communication performance and the computational performance. And taking into account that low-cost tags are highly resource-constrained, the tags only need to store tag’s *ID* of length *l* and an *l*-

bit secret key. So it can easily be implemented on those low-cost RFIDs like EPC generation 2 RFID.

In Our proposed protocol, it doesn't offer forward secrecy since the key updating is not fulfilled after the mutual authentication. But, we find that the previous re-keying protocols like [1, 2, 5, 8, 9] all suffer from DOS attacks. Furthermore, designing a protocol that simultaneously ensures forward secrecy and DOS attack resistance is our future work.

Acknowledgments. This research is partially supported by National Science Council with project number NSC95-2221-E-260-050-MY2.

References

1. Henrici, D., Müller, P.: Hash-based Enhancement of Location Privacy for Radio-Frequency Identification Devices using Varying Identifiers. PerSec04 at IEEE PerCom (2004)
2. Lee, S.M., Hwang, Y.J., Lee, D.H., Lim, J.I.: Efficient Authentication for Low-Cost RFID Systems. International Conference on Computational Science and its Applications - ICCSA 2005, May (2005)
3. Weis, S.A., Sarma, S.E., Rivest, R.L., Engels, D.W.: Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. International Conference on Security in Pervasive Computing, March (2003)
4. Li, Y.Z., Cho, Y.B., Um, N.K., Lee, S.H.: Security and Privacy on Authentication Protocol for Low-cost RFID. IEEE International Conference on Computational Intelligence and Security, Volume 2, pp.1101--1104, Nov. (2006)
5. Lee, Y.K., Verbauwhede, I.: Secure and Low-cost RFID Authentication Protocols. Adaptive Wireless Networks－AWiN, November (2005)
6. Chien, H.Y., Chen, C.H.: Mutual Authentication Protocol for RFID Conforming to EPC Class 1 Generation 2 Standards. Computers Standards and Interfaces, Vol. 29/2, pp.254--259 (2007)
7. Lin, C.-L., Chang, G.-G.: Cryptanalysis of EPC Class 1 Generation 2 RFID authentication. Information Security Conference 2007, ChiaYi, Taiwan (2007)
8. Duc, D.N., Park, J., Lee, H., Kim, K.: Enhancing Security of EPCglobal Gen-2 RFID Tag against Traceability and Cloning. The 2006 Symposium on Cryptography and Information Security (2006)
9. Karthikeyan, S., Nesterenko, M.: RFID security without extensive cryptography. In Workshop on Security of ad hoc and sensor networks, pp.63--67. ACM Press, Alexandria, Virginia, USA (2005)