Efficient RFID Authentication Protocol for Ubiquitous Computing Environment*

Eun Young Choi, Su Mi Lee, and Dong Hoon Lee

Center for Information Security Technologies(CIST),
Korea University, 1, 5-Ka, Anam-dong, Sungbuk-ku, Seoul, 136-701, Korea
{bluecey, smlee}@cist.korea.ac.kr, donghlee@korea.ac.kr

Abstract. Radio Frequency identificiation (RFID) will become an important technology in remotely object identification systems. However, the use of RFID tags may create new threats to the security and privacy of individuals holding RFID tags. These threats bring several problems which are information leakage of a tag, location trace of individuals and impersonation of a tag. Low-cost RFID systems have much restrictions such as the limited computing power, passive power mechanism and low storage space. Therefore, the cost of tag's computation should be considered as an important factor in low-cost RFID systems. We propose an authentication protocol, OHLCAP which requires only *one* one-way hash function operation and hence is very efficient. Furthermore, our protocol is suitable to ubiquitous computing environment.

1 Introduction

A Radio Frequency Identification (RFID) tag is a microchip that is capable of transmitting a unique serial number and other additional data through RF(radio frequency) signals. The goal of a RFID system is to identify objects remotely by embedding tags into the objects. For example, goods in shops can be tagged in order to provide automatic theft-detection, or to manage the goods inventory by using wireless scanning without any handwork. RFID tags are useful tools in manufacturing, supply chain management, inventory control, etc.

A RFID system is composed of three components; tag, reader and Back-end database. The characteristics of each component are as follows.

– *RFID tag* carries an object identifying data. When a tag receives a query from a reader, the tag transmits information to the reader using RF signals.
– *RFID reader* reads and re-writes the stored data in a tag. After a reader queries to a tag and receives information from the tag, the reader forwards the information to a Back-end database.
– *Back-end database* is powerful in computational capacity and manages lots of information related to each tag. Generally we assume that an adversary can

* This research was supported by grant No.R01 – 2004 – 000 – 10704 – 0 from the Korea Science & Engineering Foundation.

T. Enokido et al. (Eds.): EUC Workshops 2005, LNCS 3823, pp. 945–954, 2005.

monitor all messages transmitted in wireless communication between a reader and a tag. However in wired communication between a reader and a Back-end database, we assume that the reader can establish secure connection with the Back-end database.

In RFID systems, a RFID tag transmits information to a nearby reader using RF signals. The RF communication used in RFID systems makes it vulnerable to various attacks such as eavesdropping, traffic analysis, message interception and impersonation (e.g., spoofing and replay). Among the various attacks, the impersonation attack permits an adversary to fool RFID systems. For example, (1) In case of a spoofing attack, an adversary can replace a tag of an expensive item with a bogus tag which transmits data obtained from a cheaper item in response to a query from a nearby reader. The tag of an expensive item is attached to some one in shop. When the expensive item passes the checkout counter, a price of the cheaper item is charged for the expensive item and the expensive item is still perceived as existent one in shop. (2) In case of a replay attack, an adversary can impersonate the tag by retransmitting previously transmitted message between a tag and a reader. Therefore, these attacks allow an adversary to fool RFID systems. To prevent these attacks, RFID systems should provide mutual authentication between a reader and a tag to assure that no adversaries can make valid message.

In this paper, we study mutual authentication scheme as an efficient method to resolve these problems, especially for low-cost systems. A low-cost RFID tag is limited in computing power, communication mechanism and storage space since a RFID chip with approximately 4,000 gates is considered to be low-cost. This implies that previously classical authentication schemes are not suitable. Therefore, it is important to construct an efficient authentication scheme for low-cost RFID systems. Furthermore, we will face up to ubiquitous computing environment in the near future. It is also important to construct protocol which is well suitable to ubiquitous computing environment. In this paper, we propose mutual authentication protocol which is suitable to ubiquitous computing environment.

1.1 Related Work

Researchers have recognized the privacy problem of RFID tags [8] and are continuing to devise better approaches to protect a user privacy. We describe some of the related studies below. The simplest physical approach for the protection of user privacy is to"kill" RFID tags [10] before they was put in the hands of a user. However, a low-cost RFID tag will be used in numerous applications and many of these applications may require that tags maintain active state in the hands of a user. Therefore, this method is not a useful solution. In addition to "kill" method, other physical methods are Faraday Cage and active jamming [3]. In addition to "kill" method, other physical methods are Faraday Cage and active jamming [3]. However, two methods are also not suitable to protect a user privacy.

Another general approach is using encryption algorithm. In this approach, messages are encrypted using asymmetric public key algorithms [1, 2, 4] which are based on re-encryption method. In [2], Juels *et al.* proposed a scheme to protect a user privacy implications of RFID-tags embedded in banknotes. The resulting ciphertext undergoes periodic re-encryption period. Recently, Avoine described the privacy issues in RFID banknote protection scheme [1]. In [1, 4], these schemes are based on universal re-encryption used in Mixnets. However, this approach cannot protect a user privacy from a malicious reader. If a malicious reader only receives a response from a tag and do not perform re-encryption operation, then the malicious reader can obtain constant ciphertext. Using this process consecutively, the malicious reader obtains user's location history.

The other approach is to design an authentication protocol using one-way hash function [5, 6, 10]. This approach can prevent an exposure of tag ID using one-wayness property of hash function. However, schemes of papers [5, 6, 10] provide partial solutions to protect a user privacy. In [10], whenever a tag receives a query from a reader, the tag responds with its *metaID* which is fixed. Therefore, an adversary can trace the tag using *metaID.* Ohkubo *et al.* proposed a protocol using a hash chain mechanism [6]. This method uses two different hash functions to protect a user privacy. However, the Back-end database should compute all the hash chains, i.e., it is impractical. However, an adversary can attack these schemes [6] using eavesdropping or impersonation attack. Henrici *et al.* also proposed a simple scheme [5], called hash-based ID variation scheme (HIDV), using one-way hash function and the scheme enhances location privacy by changing traceable identifiers on every session. The proposed scheme is not secure against impersonation attack such as spoofing. Recently, LEE *et al.* [9] proposed LCAP protocol which improved HIDV scheme in both efficiency and security. Also, Rhee *et al.* proposed challenge-response based RFID authentication protocol (CRAP) which is suitable to ubiquitous computing environment [7].

1.2 Contribution

We propose an efficient authentication protocol, OHLCAP, for Hash-based low-cost RFID systems, which is suitable to ubiquitous computing environment. In Table 1, we show efficiency analysis with respect to computation cost and security against various threats in LCAP, CRAP, and OHLCAP. Also, we consider whether the schemes are suitable to ubiquitous computing environment or not.

- *Application:* In ubiquitous computing environment, components of the RFID systems can exist in anywhere. As schemes described in papers [5, 9], if a tag's ID should be dynamic value to protect a user privacy, the tag only communicates with a fixed Back-end database since the tag must synchronize the tag's dynamic ID value with the Back-end database. However if a tag's ID is static value such as CRAP [7], then the tag can perform authentication protocol with any Back-end database since the scheme does not need synchronization of the tag's ID between a Back-end database and the tag. Therefore, the tag holding static ID is able to communicate with any reader in ubiquitous computing environment. As shown in Table 1, OHLCAP is suitable to ubiquitous computing environment because of using static ID. Although our protocol uses static ID, it is secure against various attacks.

Table 1. The analysis of efficiency and security

Protocol		LCAP	CRAP	OHLCAP
Memory.	Tag	$1l$	$1l$	$5l$
	Back-end database	$6l$	$1l$	$4l$
Computation.	Tag	$2H$	$3H$	$1H\ (+A)$
	Back-end database	$1H$	$(\frac{N}{2}+1)H$	$1H+\varepsilon$
Communication.	Tag → Reader	$1\frac{1}{2}l$	$2l$	$2\frac{1}{2}l$
	Reader → Tag	$\frac{1}{2}l$	l	$\frac{1}{2}l$
Spoofing		Prevention	Prevention	Prevention
Loss of message		Restoration	–	–
Replay attack		Prevention	Prevention	Prevention
Location privacy		Prevention	Prevention	Prevention
Distributed database environment		Unsuitability	Suitability	Suitability

Notations of Table: l : the output size of a one-way hash function or the length of ID, H : the cost of a one-way hash function operation, N : the number of tags in a Back-end database, A : the cost of additional operations except for hash operation in a tag, ε : the cost of additional operations except for hash operation in a Back-end database, – : No consideration.
– *Memory.* : the storage cost of each entity.
– *Computation.* : the maximum computation cost of each entity during the execution of an authentication protocol.
– *Communication.* : the length of bits that a tag and a reader send during the execution of an authentication protocol.

- *Efficiency* : As shown in Table 1, we consider a storage cost, a communication cost, and a computation cost of each entity. As compared with the previously proposed schemes in Table.1, although OHLCAP stores more secret values than both LCAP and CRAP, OHLCAP requires that a tag only operates *one* one-way hash function operation, and additional operations A which are four xor-operations and one addition operation. Since both xor-operation and addition operation are very simple bits operation, hardware embodiment of these operations is simpler than one-way Hash function. Therefore, OHLCAP is suitable to a low-cost RFID tag.

ORGANIZATION OF THE PAPER. This paper is organized as follows: In Section 2, we describe security and privacy risks in RFID systems. We describe our scheme OHLCAP in Section 3. In Section 4, we analyze our scheme in security. Finally, we conclude in Section 5.

2 Security and Privacy Risks

2.1 Security Risks

In RFID systems, since an adversary can monitor all messages transmitted in wireless communication between a reader and a tag, the adversary can infringe upon a person's privacy using various methods. Therefore, RFID systems must be designed to be secure against attacks such as eavesdropping, traffic analysis, message interception and impersonation (e.g., spoofing and replay) as described below.

Passive attack - Eavesdropping: A passive adversary can eavesdrop on messages between a reader and a tag. By eavesdropping, the adversary may obtain a user's secret information. So, RFID systems should be designed that the eavesdropper cannot get any secret information from the eavesdropped messages.

Active attack - Impersonation: An active adversary can query to a tag and a reader in RFID systems. By this property, the adversary can impersonate the target tag or reader. There are two types of impersonation attack; replay and spoofing. Besides of impersonation attack, an active adversary can try to trace the location of a tag using traffic analysis : distinguishing whether the response is transmitted by the target tag or not. Therefore, RFID systems should be designed that an active adversary cannot impersonate a target tag or reader and distinguish a target tag's response from a random value.

Active attack - Message interception: In this attack, although an adversary cannot obtain any information in RFID systems, message interception makes a target tag unable to operate further. Among the previously proposed schemes, several schemes such as [5, 9] require that a tag should receive some value from a Back-end database and update stored values using the received value. If message interception occurs in these schemes, the Back-end database should be able to restore the messages. In result, RFID systems can normally operate. Therefore, RFID systems should be able to detect message interception except that a tag does not need to receive updating values from a reader for next session.

2.2 Privacy Risks

As mentioned above, an adversary is able to attack RFID systems using various methods. These attacks make a tag able to disclose sensitive information to an unauthorized reader. If a link between a tag and a user holding the tag is established, his movement can be traced by tracking the tag's ID. This implies that the adversary infringes a user's location privacy. To design secure RFID systems, we should consider these risks in detail below.

−*Information Leakage* : A person is prone to carrying various tagged objects in every life. Some of objects such as expensive products and medicine are quite personal and provide information that the user does not want anyone to know. In RFID systems, the tag emits only distinguishable information in response to a query from a nearby reader. So, various personal information can be leaked without the acknowledgement of the user.
− *Traceability* : When a target tag transmits a response to a nearby reader, an adversary can record the transmitted message and can establish a link between the response and the target tag. Once a link established, the adversary is able to know the user's location history.

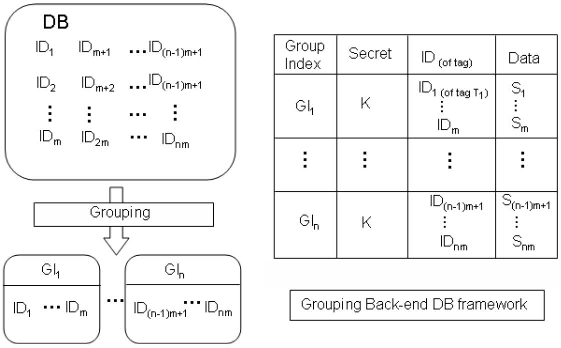
3 Our Hash-Based Low-Cost Authentication Protocol

In this section, we describe our OHLCAP for Hash-based low-Cost RFID systems. OHLCAP consists of set-up and mutual authentication phases.

3.1 System Set-Up

Let $H : \{0,1\}^* \rightarrow \{0,1\}^l$ be a one-way hash function, where a hash value space belongs to $\{0,1\}^l$. ID denotes identity of a tag and is a unique value in $\{0,1\}^l$. In the set-up phase, both a tag and a Back-end database store several secret values and tag's ID. Data fields of a tag and a reader are initialized to the following values:

1. Back-end Database : First, a Back-end database divides identities of tags into several groups. If a number of system's tags are $N(= mn)$, a Back-end database divides it into n groups which include m identities of tags and generates a group index GI in each group, as shown Figure 1. Then, data fields of a Back-end database are initialized to GI, ID, K, S and DATA. The Back-end database needs a one-way hash function to execute hash function operation.
   - GI is a group index of tags with l-bit string. If a tag belongs to i-th group, $\mathtt{GI}_i$ is a group index of the tag.
   - K is a secret value with l-bit string and is stored in all tags. S is a tag's secret value with l-bit string.
   - ID is l-bit string, which is used for identifying. Tag's IDs differs from group indices $\mathtt{GI}_i$, $i \in \{1, ..., n\}$.
   - DATA stores an accessible information about each tag, e.g., a secret value S.
2. Reader : A reader picks uniformly a random value r with $\{0,1\}^l$. A reader does not need to execute any operation. A reader merely forwards a tag's message (or a Back-end database's message) to a Back-end database (or a tag).
3. Tag : The data field of a tag is initialized to its own ID, GI, K and S, c. The tag stores ID, GI, K, S, and a counter c. The counter c is initialized


DB
ID_1 ID_{m+1} ... $\mathrm{ID}_{(n-1)m+1}$
ID_2 ID_{m+2} ... $\mathrm{ID}_{(n-1)m+1}$
⋮ ⋮ ... ⋮
ID_m ID_{2m} ... ID_{nm}
Grouping
GI_1
ID_1 ... ID_m
...
GI_n
$\mathrm{ID}_{(n-1)m+1}$ ... ID_{nm}

Group Index	Secret	ID (of tag)	Data
GI_1	K	ID_1 (of tag T_1) ⋮ ID_m	S_1 ⋮ S_m
⋮	⋮	⋮	⋮
GI_n	K	$\mathrm{ID}_{(n-1)m+1}$ ⋮ ID_{nm}	$S_{(n-1)m+1}$ ⋮ S_{nm}

Grouping Back-end DB framework

Fig. 1. Back-end database framework

by an arbitrary value, which is l-bit string. Whenever a tag receives a query from a nearby reader, the tag increase a counter c. To execute a one-way hash function operation, the tag needs a one-way hash function.

3.2 Mutual Authentication - OHLCAP

When a reader queries to a tag, the tag and the reader authenticate each other as shown in Figure 2. To help to understand OHLCAP protocol, we assume that the tag belongs to i-th group.

NOTATIONS. The addition operation of bits is denoted by + and the exclusive-or (xor) operation of bits is denoted by $\oplus$. $\mathtt{m}||\mathtt{w}$ denotes the concatenation of two messages, m and w.

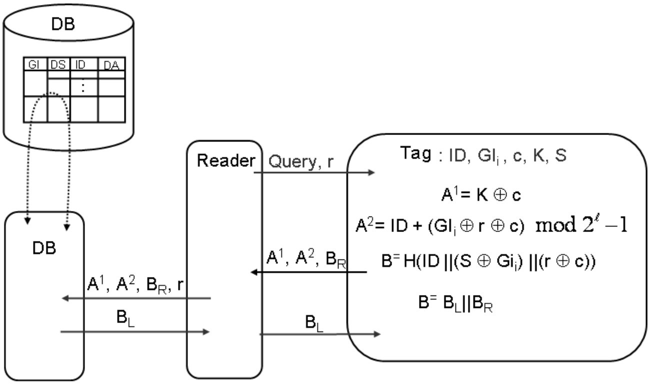


Fig. 2. OHLCAP protocl

Step 1. A reader picks a random value r and sends Query and r to a nearby tag.

Step 2. To respond to the query of the reader, the tag checks a random value r whether it is all zero value or not.

1. If r value is all zero, the tag sends "stop" message to the reader and halts the protocol.
2. Otherwise, the tag performs processes as follows.
   - The tag computes $\mathtt{A}^1 = \mathtt{K} \oplus \mathtt{c}$, $\mathtt{A}^2 = \mathtt{ID} + (\mathtt{GI}_i \oplus \mathtt{r} \oplus \mathtt{c})\ mod\ (2^l - 1)$ using r, c and its own ID, $\mathtt{GI}_i$ and K.
   - Also, the tag computes $\mathtt{B}=\mathrm{H}(\mathtt{ID}||(\mathtt{S}\oplus\mathtt{GI}_i)||(\mathtt{r}\oplus\mathtt{c}))$ using ID, c, r, $\mathtt{GI}_i$ and S, and sends $\mathtt{A}^1$, $\mathtt{A}^2$ and $\mathtt{B_R}$ to the reader, where $\mathtt{B_R}$ is a right half of B, so $\mathtt{B_R}$ has the length of $\frac{1}{2}l$ bit.
   - Then, the tag increases the counter c which should not exceed 2^l-1. If the counter c exceeds 2^l-1, it is initialized by initial c.

Step 3. Upon receiving $\mathtt{A}^1$, $\mathtt{A}^2$ and $\mathtt{B_R}$ from the tag,

1. The reader forwards $\mathtt{A}^1$, $\mathtt{A}^2$, $\mathtt{B_R}$ and $\mathtt{r}$ to the Back-end database.
2. The Back-end database computes $\mathtt{c}'= \mathtt{A}^1 \oplus \mathtt{K}$ and $\mathtt{ID}'_j = \mathtt{A}^2 \text{-} (\mathtt{GI}_j \oplus \mathtt{r} \oplus \mathtt{c}')$ *mod* $(2^l\text{-}1)$ using all group indices $\mathtt{GI}_j$, $j \in \{1, ..., n\}$.
3. The Back-end database checks if one of computed $\mathtt{ID}'_{j(\in\{1,...,n\})}$ is matching to one of the stored IDs in the Back-end database. If this process succeeds, the Back-end database check if the $\mathtt{GI}_j$ used to compute $\mathtt{ID}'_j$ is equal to the group index $\mathtt{GI}_i$ that contains the matching $\mathtt{ID}'_j$.
   - If this succeeds, the Back-end database computes $H(\mathtt{ID}||(\mathtt{S} \oplus \mathtt{GI}_i)||(\mathtt{r} \oplus \mathtt{c}))$ using $\mathtt{c}$, $\mathtt{r}$, $\mathtt{GI}_i$, $\mathtt{S}$ and the matched ID.
   - Otherwise, the Back-end database halts this process.
4. Then, the Back-end database authenticates the tag by checking if the right half of the computing value $H(\mathtt{ID}||(\mathtt{S} \oplus \mathtt{GI}_i)||(\mathtt{r} \oplus \mathtt{c}))$ is equal to the received value $\mathtt{B_R}$.
5. The Back-end database sends $\mathtt{B_L}$ to the reader, where $\mathtt{B_L}$ is a left half of B. the reader forwards $\mathtt{B_L}$ to the tag.

Step 4. The tag authenticates the reader by checking if the received value $\mathtt{B_L}$ is equal to the left half of B of step 2.

4 Security Analysis

In this section, we analyze our protocol OHLCAP in security. Considering attack methods in described section 2.2, we analyze the security of our protocol against the threats introduced in section 2.2; *information leakage* and *traceability.*

Information Leakage. In OHLCAP, an adversary must be authenticated to get any sensitive information in a tag. To pass authentication protocol without knowing GI, K, c, S and ID, an adversary only must guess $\mathtt{B}_L$ value after collecting messages $\mathtt{A}^1$, $\mathtt{A}^2$, and $\mathtt{B}_R$. However, because of one-wayness property of hash function H, the adversary cannot get sensitive information of $\mathtt{B}_L$ from $\mathtt{A}^1$, $\mathtt{A}^2$ and $\mathtt{B}_R$. In OHLCAP, since an adversary does not know a secret K, even if the adversary eavesdrops $\mathtt{A}^1$, the adversary cannot get the tag's group index GI. So, the adversary cannot get any information of $\mathtt{B}_L$ from $\mathtt{A}^2$. Therefore, the adversary has to randomly pick a string from $\{0,1\}^{\frac{1}{2}l}$. Also, even if an adversary collects the hash values $\mathtt{B}_L$, $\mathtt{B}_R$, the adversary cannot get information of tag's ID. In order to guess the target tag's ID, the adversary has to randomly select a string from $\{0,1\}^l$ by one-wayness property of hash function H. Therefore, the advantage of the adversary is at most $\frac{1}{2^{(l/2)}} + \frac{1}{2^l}$, which is negligible.

Traceability. Our OHLCAP protocol guarantees location privacy by using refreshed values r,c, where r and c are refreshed by a reader and a tag in each session, respectively. Even if a malicious reader does not refresh a random value r, a tag transmits the refreshed values that are refreshed by a counter c, where the counter c is refreshed by a tag in each session.

– In OHLCAP protocol, an adversary can eavesdrop on $\mathtt{A}^1$, $\mathtt{A}^2$ in between a reader and a target tag. Since the adversary does not know secret K, she is not able to extract the $\mathtt{c}$ value. Thus, the adversary cannot obtain the tag's group index $\mathtt{GI}$ from eavesdropped messages $\mathtt{A}^1$, $\mathtt{A}^2$. Therefore, it is impossible that the adversary obtains the target tag's ID. This means that the adversary cannot trace the target tag.

– In OHLCAP protocol, since all tags in one group uses an identical group index $\mathtt{GI}$ and a secret $\mathtt{K}$, we consider a special attack that an adversary obtains secret value $\mathtt{K}$ and some $\mathtt{GI}_j$ $(j \in \{1, ..., n\})$ by only attacking physically some tag. This tag is unable to operate further. The adversary try to attack OHLCAP using obtained values. First, the adversary eavesdrops on $\mathtt{A}^1$, $\mathtt{A}^2$, $\mathtt{B}_L$ and $\mathtt{B}_R$ between a reader and a target tag. Then, the adversary can extract a counter $\mathtt{c}$ from $\mathtt{A}^1$ using the value $\mathtt{K}$ and compute a some $\mathtt{ID}'$ from $\mathtt{A}^2$ by using obtained values $\mathtt{GI}_j$. The adversary does not know whether a computed $\mathtt{ID}'$ is the tag's $\mathtt{ID}$ or not. So, by using one-way hash function, the adversary should check if eavesdropped value $\mathtt{B}$ is equal to $H(\mathtt{ID}' \,||(\mathtt{S}\oplus\mathtt{GI}_j)||(\mathtt{r}\oplus\mathtt{c}))$. However, the adversary does not know a secret value $\mathtt{S}$. Therefore, the adversary is not able to check if a computed $\mathtt{ID}'$ is the target tag's $\mathtt{ID}$, and cannot compute the target tag's $\mathtt{ID}$. Thus, the adversary cannot trace the target tag.

In RFID systems, as mentioned in section 2.1, an adversary can attack various attacks such as eavesdropping, traffic analysis, message interception and impersonation. In order to analyze about a user privacy protection of our protocol, we only consider attacks such as eavesdropping and traffic analysis. Now, we show that our protocol is secure against remaining attacks such as message interception and impersonation(e.g., spoofing and replay).

Impersonation. In our protocol, impersonation attack can be prevented by mutual authentication between a reader and a tag. In OHLCAP, an adversary cannot impersonate a target tag using *a replay attack* since the valid massage is refreshed in each session by a random value $\mathtt{r}$ and a counter $\mathtt{c}$. Also, an adversary queries a target tag by impersonating as a reader, receives messages back from the target tag, then she may try a spoofing attack to impersonate the target tag. However, without knowing $\mathtt{ID}$, $\mathtt{GI}$, $\mathtt{S}$ of the target tag, the adversary is unable to compute a half right $\mathtt{B}_R$ of the $\mathtt{B}$ that can only be generated by the target tag. Therefore, it is impossible to impersonate the target tag by *a spoofing attack* in OHLCAP.

In OHLCAP, a tag does not receive any message from a reader in order to update own ID. Even if loss of message occurs between a tag and a reader, the tag increases a counter $\mathtt{c}$ by itself and computes $\mathtt{A}^1$, $\mathtt{A}^2$, $\mathtt{B}_R$ using $\mathtt{r}$ received from a nearby reader in next session. Therefore, message intercetpion does not need to be considered in OHLCAP.

5 Conclusion

We have proposed an efficient and secure authentication protocol OHLCAP to protect a user privacy, especially for low-cost RFID systems in ubiquitous computing environment. The proposed scheme needs only *one* one-way hash function operation and hence is quite efficient. Leakage of information is prevented in the scheme since a tag emits its information only after authentication. By refreshing a message transmitted from a tag in each session, OHLCAP also provides a location privacy and is secure against many attacks such as eavesdropping, traffic analysis, message interception, spoofing and replay.

References

1. G. Avoine. *Privacy issues in RFID banknote protection schemes.* In international Conference on Smart Card Research and Advanced Applications - CARDIS, Toulouse, pp.22-27, 2004.
2. A. Juels and R. Pappu. *Squealing euros : Privacy protection in RFID-enabled banknotes.* In proceedings of Financial Cryptography -FC'03, vol.2742 LNCS, pp.103 121, Springer-Verlag, 2003.
3. A. Juels, R. L. Rivest and M. Szudlo. *The Blocker Tag: Selective Blocking of RFID tags for Consumer Privacy.* In the 8th ACM Conference on Computer and Communications Security, pp. 103-111, ACM Press, 2003.
4. S. Junichiro, R. Jae-Cheol and S. Kouichi, *Enhancing privacy of Universal Re-encryption scheme for RFID Tags.* EUC 2004, Vol. 3207 LNCS, pp.879-890, Springer-Verlag, 12, 2004
5. D. Henrici and P. Muller. *Hash-based Enhancement of Location Privacy for Radio-Frequency Identification Devices using Varying Identifiers.* PerSec'04 at IEEE PerCom. 2004
6. M. Ohkubo, K. Suxuki and S. Kinoshita. *Efficient Hash-Chain Based RFID Privacy Protection Scheme.* Ubcomp2004 workshop.
7. Keunwoo Rhee, Jin Kwak, Seungjoo Kim and Dongho Won. *Challenge-Response Based RFID Authentication Protocol for Distributed Database Environmnet.*, SPC 2005, LNCS 3450, pp. 70-84, 2005.
8. S. E. Sarma, S. A. Weis and D. W. Engels. *Radio-frequency identification systems.* CHES'02, vol.2523 LNCS, pp.454-469, Springer-Verlag, 2002.
9. L. Su Mi, H. Young Ju, L. Dong Hoon and L. Jong In. *Efficient Authentication for Low-Cost RFID systems.* ICCSA05, vol. 3480 LNCS, pp.619-629, Springer-Verlag, 2005.
10. S. A. Weis, S. E. Sarma, S. A. Weis and D. 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