Security Analysis and Enhancement of One-Way Hash based Low-Cost Authentication Protocol(OHLCAP) *

JeaCheol Ha[1], SangJae Moon[2], Juan Manuel Gonzalez Nieto[3], and Colin Boyd[3]

[1] Dept. of Information Security, Hoseo Univ., 336-795, Korea
jcha@hoseo.edu
[2] School of Electrical Eng. and Computer Science, Kyungpook National Univ., 702-701, Korea
{short98, sjmoon}@ee.knu.ac.kr
[3] Information Security Institute, Queensland Univ. of Technology, GPO Box 2434, Brisbane, QLD, 4001, Australia
{juamma, boyd}@isrc.qut.edu.au

Abstract. Choi *et al.* recently proposed an efficient RFID authentication protocol for a ubiquitous computing environment, OHLCAP(One-Way Hash based Low-Cost Authentication Protocol). However, this paper reveals that the protocol has several security weaknesses : 1) traceability based on the leakage of counter information, 2) vulnerability to an impersonation attack by maliciously updating a random number, and 3) traceability based on a physically-attacked tag. Finally, a security enhanced group-based authentication protocol is presented.

Keywords: RFID system, group-based authentication, indistinguishability, traceability.

1 Introduction

Radio Frequency Identification(RFID) systems, consisting of RFID tags, an RFID reader, and back-end database, are expected to replace optical bar codes due to several advantages, such as their low cost, small size, quick identification, and embedded implementation into objects. However, communication using the RF signal between a tag and a reader can create new threats to the security and privacy of a RFID tag, including the leakage of privacy, location tracing, and tag or reader impersonation.

Various attempts have already been made to protect the privacy of a tag using physical technology, such as the 'Kill command' [12], 'Active jamming' [5], and 'Blocker tag' [5] approaches. However, none have been successful. As a cryptographic solution, Weis *et al.* [10–12] proposed a hash-lock protocol and randomized hash-lock protocol. Yet, with the hash-lock protocol, since the *metaID*

* This research was supported by the MIC of Korea, under the ITRC support program supervised by the IITA(IITA-2006-C1090-0603-0026).

is unique for each tag, location privacy is compromised due to the fixed *metaID*. Meanwhile, with the randomized hash-lock protocol, the identity of a tag, ID_k is transmitted from the reader to the tag, making the system vulnerable to a replay attack, spoofing attack, and location tracing. Henrici and Müller [4] proposed an ID variation protocol, that is secure against a replay attack, yet location privacy is compromised as the tag's response remains constant until the next authentication session when desynchronization occurs [8]. Ohkubo *et al.* [7] proposed a hash chain-based authentication protocol in which the reader sends a query using two different hash functions, however this scheme is still vulnerable to a replay attack and spoofing attack. In 2005, Lee *et al.* [6] proposed a low-cost RFID authentication scheme in which a tag and the back-end database only perform two one-way hash operations, yet this scheme is still vulnerable to a spoofing attack and location-tracing attack when desynchronization occurs. More recently, Choi *et al.* [1] proposed an efficient RFID authentication protocol for a ubiquitous computing environment, where the tag's ID is static. In [1], the authors claim that their protocol guarantees location privacy due to the use of fresh values in every session, plus an adversary cannot trace the target tag using a physical attack, even when certain secret values are obtained.

However, this paper shows that the protocol developed by Choi *et al.* still has security weaknesses. First, an adversary can trace a tag using leaked counter information. Second, an adversary can impersonate a reader by maliciously updating the random number obtained from the previous session. Finally, in the case of a physically attacked tag, an adversary can easily trace a target tag. Therefore, a low-cost authentication protocol that enhances OHLCAP is proposed to protect against the above attacks.

2 Security Threats to RFID system

An RFID system usually consists of three parts: RFID tags(transponders), the RFID reader(transceiver), and back-end database(Back-end server). An RFID tag includes a microchip for computing and a coupling element, such as an antenna, for communication with the RFID reader. The RFID reader interrogates the tags using an RF signal, then transmits the collected data to the back-end database. However, the channel between the reader and a tag is insecure, as it is based on wireless communication. After the back-end database receives the data from the reader, it transmits certain information to a authenticated tag. The channel between the reader and the database is considered as secure. In this paper, it is assumed that an adversary has the following capabilities:

- **Eavesdropping:** An attacker has a capacity to eavesdrop messages between the reader and the tags due to an insecure channel, then uses the intermediate information or useful responses to try certain enhanced attacks, such as location tracing or a spoofing attack. Therefore, an RFID system should at least protect against information leakage in an insecure channel.
- **Transmitting a malicious message or replaying:** It is assumed that an adversary has the capability to transmit certain malicious messages to the

tag or the reader. By transmitting these messages, the attacker can perform a spoofing attack or replay attack.

- **Interrupting a message:** The communication messages between the tags and the reader can be blocked by an attacker. As a result, a message interrupt attack can bring into desynchronization state between the tag and the reader, due to an abnormal closing of a session, malicious blocking of messages, or different updating of ID between the tag and the database. Furthermore, several successive message interrupts can be used by an attacker in location tracing a target tag.

Since the communication between the reader and the tag is performed using an wireless RF interface, the communicated data can easily be tapped by an attacker. The various security threats that can occur with an insecure channel are categorized as follows:

- **Information leakage:** One RFID privacy problem is information leakage about a user's belongings. For example, a user may not want certain information known by others, such as ownership of expensive products, identification of personal medicine, and so on.
- **Impersonation attack:** After an adversary sends a malicious query to a target tag, they collect the responses emitted by the tag. The attacker can then impersonate the reader using the messages collected from the tag. Conversely, an adversary can replay the reader's query to impersonate the target tag. An attacker can also impersonate a legal tag or reader by replaying certain useful messages.
- **Desynchronization attack:** If the current ID for a tag is different to the one in the database, this is referred to as a state of desynchronization. An adversary can block certain transmitted messages between the tag and the reader, creating a dysynchronization state. This state can occur in an ID-renewable RFID system. If the ID of a tag is desynchronized, the tag can be easily traced, as one of emitted values from the tag will be constant, thereby compromising the location privacy.
- **Location tracing attack:** Here, the adversary can seek some useful information on a tag's location trace. This attack is essentially applied to a rigid RFID system in which certain communication messages between the tag and the database are identical to those used in the previous session.

3 Review of OHLCAP

This section briefly reviews Choi *et al.*'s One-way Hash based Low-Cost Authentication Protocol(OHLCAP).

Database	Reader	Tag
Database field $[GI_i][K][ID_{ij}][S_{ij}]$		Tag field $[ID][GI][K][S][c]$
	$\xrightarrow{Query,\ r}$	$A^1 = K \oplus c$ $A^2 = ID + (GI \oplus r \oplus c)$ $B = H(ID\|(S \oplus GI)\|(r \oplus c))$
Computes c	$\xleftarrow{A^1, A^2, B_R, r}$	$\xleftarrow{A^1, A^2, B_R}$ $B = B_L\|B_R$
Computes IDs for every GI_i Finds ID and checks GI $B = H(ID\|(S \oplus GI)\|(r \oplus c))$		
Checks B_R	$\xrightarrow{B_L}$	$\xrightarrow{B_L}$ Checks B_L

Fig. 1. OHLCAP: One-way Hash-based Low-Cost Authentication Protocol.

3.1 Notations

$H()$: one-way hash function,$H : \{0,1\}^* \rightarrow \{0,1\}^l$
ID_{ij} : identity of jth tag in ith group, l bits
S_{ij} : secret key for jth tag in ith group, l bits
GI_i : ith group index, l bits
K : common secret key in DB and all tags, l bits
r : random number generated by reader,
t : random number generated by tag, l bits
c : counter stored in tag, l bits
$Query$: request generated by reader
B_R : right half of message B
B_L : left half of message B
x_p : value of x in previous session
x_c : value of x in current session
$+$: modular addition by $mod\ (2^l - 1)$
$\oplus$: exclusive-or(xor) operation
$||$: concatenation of two inputs

3.2 Description of OHLCAP

OHLCAP consists of a set-up and mutual authentication phase, as described in Fig. 1.

1) Set-up phase:

- Back-end database: Divides all the tags into n groups, which include m tag identities. The data field of the back-end database is $GI_i||K||ID_{ij}||S_{ij}$.
- Tag: A tag is initialized by a data field, including $ID||GI||K||S$ and a counter c, where K is the same in all tags and GI is the same within a group.

2) Authentication phase:

- Step 1. The reader sends a $Query$ and r to a tag.
- Step 2. The tag computes A^1, A^2 and B, then sends them to the reader. The tag increases the counter c whenever it receives a query from the reader.
- Step 3. Upon receiving A^1, A^2, and B_R from the tag, the reader forwards them with r to the back-end database.
- Step 4. The back-end database computes $c' = A^1 \oplus K$ and $ID'_i = A^2 - (GI'_i \oplus r \oplus c')$ using all the group indices $GI'_i, i \in \{1, \ldots n\}$. If one of the computed ID'_i matches one of the stored IDs, the back-end database checks if one of the computed ID'_is matches one of the stored IDs, the back-end database then checks whether the GI'_i contains the ID'_i matching that for the true GI_i group. The back-end database authenticates the tag by checking that the computed B_R equals the received one, then sends the B_L to the reader.
- Step 5. The reader forwards the B_L to the tag
- Step 6. The tag authenticates the reader by checking the B_L.

4 Security Analysis of OHLCAP

This section analyzes the security weaknesses of OHLCAP and provides attack details.

4.1 Traceability using Counter Information

With OHLCAP, when responding to a query from the reader, the tag computes $A^1 = K \oplus c$ using a counter c. At this point, an adversary can trace a tag if the tag's messages are caught in two successive sessions. The following explains how OHLCAP is vulnerable to location tracing.

- Assumption: It is supposed that an adversary knows certain tag's responses from two successive sessions, ${A^1}_p = K \oplus c_p$ and ${A^1}_c = K \oplus c_c$. Here, the relationship between the two counters is $c_c = c_p + 1$.
- Attack: The adversary computes $A = {A^1}_p \oplus {A^1}_c = c_p \oplus c_c$. As a result, the secret key K is removed from the equation. The value A always has a distinguishable sign, a 1's-run value from the LSB. Now, the adversary can trace a tag by observing successive 1-runs from the LSB of A.

For example, if the previous counter value is $c_p = 1011010111$ and the current one is $c_c = 1011011000$, then $A = c_p \oplus c_c = 0000001111$, which has four 1-runs. As such, it is easy to determine that A is always one when the LSB of the first counter c_p is zero, *i.e.* if $c_p = 1011010110$ and the current $c_c = 1011010111$, then $A = 0000000001$. Thus, an adversary can trace a tag by observing two successive responses, A^1_p and A^1_c. Furthermore, if the counter is a l-bit string, then the possibility that the target tag and a random tag(with a random counter) cannot be distinguished is $l/2^l$, which is negligible as a function of l.

Table 1. Possibility of impersonation attack by updating LSB of random number unknown bit that can differ from

LSB of r_p (Known bit)	LSB of r_c (Update)	LSB of c_p (Guessing bit)	LSB of c_c (Current session)	$r_p \oplus c_p$	$r_c \oplus c_c$	Success $B_c = B_p$
0	1	0	1	*bb..bbb*0	*bb..bbb*0	O
0	1	1	0 with carry	*bb..bbb*1	*xx..xxx*1	X

b : unknown bit x : unknown bit, it can be different with b

4.2 Impersonation by Maliciously Updating Random Number

Choi *et al.* claim that reader impersonation is impossible, due to the authentication process between the reader and a tag, making it impossible for an adversary to send a correct last B_L message to the tag. However, an adversary can impersonate a legal reader using a random number from a previous session as follows:

- Assumption: It is supposed that an adversary catches two messages r_p and B_L from the previous session. Also, for the sake of simplicity, it is assumed that the LSB of r_p is zero.
- Attack: The adversary generates a malicious random number r_c, such as $r_c = r_p + 1$, in the attack session, *i.e.*, the LSB of r_p is just changed to one. After sending a *Query* and the malicious r_c to the tag and receiving some responses from the tag, the adversary then sends the same B_L as used in the previous session to the tag as the last message. Since the value B is computed by $H(ID \| (S \oplus GI) \| (r \oplus c))$, if $r_p \oplus c_p$ equals $r_c \oplus c_c$, then B_c is equal to B_p. In this case, the impersonation attack as a legal reader is successful. When $r_p \oplus c_p$ is not $r_c \oplus c_c$, the attack fails.

The following provides more detail on the above impersonation attack. The tag computes A^1, A^2, and B in response to the query from the adversary. If $r_p \oplus c_p$ is equal to $r_c \oplus c_c$, ${A^2}_c$ and B_c will be the same as ${A^2}_p$ and B_p, respectively, computed in the previous session. From the relationship of $c_c = c_p + 1$ and $r_c = r_p + 1$, if the LSB of r_p is zero, the LSB of c_p is also zero, then this attack will be absolutely successful, as $r_p \oplus c_p$ is equal to $r_c \oplus c_c$. However, if the LSB of c_p is one, such an attack is impossible, as $r_1 \oplus c_1$ is not equal to $r_2 \oplus c_c$. Therefore, an adversary can impersonate the reader by sending the random number updated by one and the last B_L message used in the previous session. The possibility of success is $1/2$ when the adversary chooses a previous random number, r_p, where the LSB is a zero bit. Table 1 outlines the possibility of an impersonation attack based on maliciously updating the LSB of a random number.

4.3 Physical Attack on Tag

When considering the case of an adversary obtaining the secret key K and group key GI_i by physically attacking a tag, Choi *et al.* claim that the adversary cannot

trace the target tag, as the secret value S is unknown, however, tracing is possible without considering S as follows:

- Assumption: It is assumed that an adversary can eavesdrop on A^1, A^2, B_L and B_R between the reader and the target tag. Furthermore, the secret key K and group key GI_i are known through a physical attack.
- Attack 1: The adversary extracts counter c from A^1 using the value K.
  - $c = A^1 \oplus K$

  Then, even though the secret value S is unknown, the tag counter can be compared with the previous one. Thus, all tags can be traced by checking the counter increment by one. Furthermore, if the adversary knows the counter number for the previous session, a malicious random number r_c can be computed, such as $r_c = (r_p \oplus c_p) \oplus c_c$ and $c_c = c_p + 1$. Since the A^2 and B in the attack session are the same as the previous session values, the adversary can easily impersonate the reader, as described in section 3.2.
- Attack 2: Since an adversary can compute the counter c, as shown above, the ID can also be extracted from the eavesdropped messages A^2 and r.
  - $ID = A^2 - (GI_i \oplus r \oplus c)$

  Then, even though the adversary does not know the tag's secret key S, the ID can be extracted from every session related to the ith group. Thus, the adversary can trace a tag by computing the ID for a group that includes a tag corrupted by a physical attack.

The above attack means that the RFID system is compromised with regard to traceability, so the RFID tag can no longer be used.

5 Security Enhancement of OHLCAP

To prevent traceability in the case of using counter information, it is recommended that a random number be used in a tag instead of a counter. However, this requires a random number generator in a tag. Alternatively, a hashed value of a stored number can be used that is changed in each session. If a random number or hashed value is used instead of a counter, the first tracing attack and impersonation attack by maliciously updating the number r_c become impossible. However, if the secret key K and GI_i are compromised, then OHLCAP cannot prevent a tracing attack, as an adversary can compute the counter value from A^1 and fixed ID for a group corrupted by a physical attack.

5.1 Group-based Low-Cost Authentication Protocol

Accordingly, a new authentication protocol is proposed that is based on a group key. In contrast to OHLCAP, the proposed protocol removes the data fields for the secret key S_{ij} and counter c due to their uselessness. To protect the RFID system from the case of K and GI_i being compromised, the proposed protocol computes three messages: $A^1 = K \oplus t$, $A^2 = GI + (r \oplus t)$, and $B =$

Database	Reader	Tag
Database field $[GI_i][K][ID_{ij}]$		Tag field $[ID][GI][K]$
	$\xrightarrow{Query,\ r}$	random number t $A^1 = K \oplus t$ $A^2 = GI + (r \oplus t)$ $B = H(ID\|GI\|r\|t))$
Computes $t = A^1 \oplus K$ Computes $GI = A^2 - (r \oplus t)$ Finds ID in GI by checking B_R	$\xleftarrow{A^1, A^2, B_R, r}$	$\xleftarrow{A^1, A^2, B_R}$ $B = B_L\|B_R$
$B = H(ID\|GI\|r\|t))$	$\xrightarrow{B_L}$	$\xrightarrow{B_L}$ Checks B_L

Fig. 2. Proposed Group based Authentication Protocol.

$H(ID\|GI\|r\|t))$. As a result, even though K and GI_i are compromised, the attacker can not extract a tag's ID from B due to the one-way property of the hash function. Therefore, the proposed group-based protocol is at least secure from above three attacks. Fig. 2 shows the process of the proposed group-based authentication protocol, and the following gives a detailed description of each step.

1) Set-up phase:

- Back-end database: Divides all the tags into n groups. The data field is $GI_i||K||ID_{ij}$.
- Tag: A tag is initialized by a data field, including $ID||GI||K$.

2) Authentication phase:

- Step 1. The reader sends a $Query$ and r to a tag.
- Step 2. The tag generates a random number t and computes $A^1 = K \oplus t$, $A^2 = GI + (r \oplus t)$, and $B = H(ID\|GI\|r\|t))$, then sends A^1, A^2, and B_R to the reader.
- Step 3. The reader forwards A^1, A^2, and B_R with r to the back-end database.
- Step 4. The back-end database computes $t = A^1 \oplus K$ and $GI = A^2 - (r \oplus t)$, then finds the ID in the GI by checking the B_R. The back-end database authenticates the tag by checking that the computed B_R equals the received one, then sends the B_L to the reader.
- Step 5. The reader forwards the B_L to the tag
- Step 6. The tag authenticates the reader by checking whether the received B_L equals the one computed in Step 2.

5.2 Security and Efficiency Analysis

The security of the proposed protocol was evaluated against the threats described in Section 2: 1) information leakage, 2)impersonation attack, 3) desynchronization attack, and 4) location tracing attack. To obtain secret information from a

tag, an adversary must be able to guess the ID. However, an adversary cannot compute the ID from the A^1, A^2, B, and r, due to the security property of a one-way hash function.

Even when an adversary collects a tag's responses, then tries to impersonate a legitimate tag, they cannot compute the hashed messages A^1, A^2, and B without knowing the K, GI, and ID values. Meanwhile, to impersonate the reader, an adversary must send the correct B_L. This is also impossible, as it cannot be computed without knowing the ID value.

In a desynchronization attack, assuming that an adversary blocks the response messages transmitted from a tag, *i.e.*, step 2 in Fig. 2, even though the tag receives the same random number r as in the previous session, the tag sends A^1, A^2, and B in the next session as a response to a query. Therefore, the proposed protocol can protect against a desynchronization attack, as the tag does not emit any useful messages for enhanced attacks, such as location tracing.

In the case of location tracing, the proposed protocol guarantees location privacy by sending different random messages for each session. After the authentication is finished in the previous session, the tag sends A^1, A^2, and B in response to a query in the current session, that is, the same response is not emitted by the tag in the subsequent session. Thus, location privacy is satisfied as A^1, A^2, and B are already refreshed in each session using two random numbers.

When evaluating the storage costs and computational load for the DB and tag, the proposed protocol makes an improvement in the storage costs for the DB as removing the secret key S_{ij} and the counter for each tag. With the proposed protocol, the storage size of the DB is $3l \cdot m$, where l is the length of an ID_{ij}, K, or group index GI_i and m is the number of IDs. Plus, a tag requires $3l$ bits of memory to store an ID, K, and the GI value. The total length of the messages transmitted from a tag to the reader is $2.5l$, while that from the reader to a tag is $1.5l$, except for a *Query*.

The computational cost in the tag and the DB can be slightly reduced compared to the original OHLCAP. The main processing in a tag is hash operation like SHA-1[9] which is the most widely used secure hash function. By high design techniques, SHA-1 needs only 405 clock cycles to compute the hash of 512 bits of data in the work of Kaps *et al.* [2], and SHA-256 requires 1,128 cycles in [3]. Therefore, the proposed protocol is also suitable for a lightweight RFID system with limited memory space and low computational power.

6 Conclusion

This paper revealed several security weaknesses of OHLCAP. Thus, to guarantee security against various threats, it is recommended that a random number be used in a tag instead of a counter. If a random number is generated in a tag, a tracing attack and impersonation attack then become impossible. Furthermore, if the secret key K and GI_i are compromised by a physical attack, OHLCAP cannot prevent a tracing attack. Thus, a group-based low-cost authentication protocol is proposed as a more secure version of OHLCAP. The proposed protocol is

robust to most threats, such as information leakage, an impersonation attack, desynchronization attack, and location tracing attack.

References

1. E. Choi, S. Lee, and D. Lee. Efficient RFID Authentication Protocol for Ubiquitous Computing Environment, In *EUC Workshops 2005*, LNCS 3823, pp. 945-954, Springer-Verlag, 2005
2. J. P. Kaps and B. Sunar. Energy Comparison of AES and SHA-1 for Ubiquitous Computing, In *EUC Workshops 2006*, LNCS 4097, pp. 372-381, Springer-Verlag, 2006
3. M. Feldhofer and C. Rechberger R. A Case Against Currently Used Hash Functions in RFID Protocols, *IS'06*, LNCS 4277, pp. 372-381,Springer-Verlag, 2006.
4. D. Henrici and P. Müller. Hash-based Enhancement of Location Privacy for Radio Frequency Identification Devices using Varying Identifiers, In *proceeding of the Second IEEE Annual Conference on Pervasive Computing and Communications Workshops*, pp. 149-162, IEEE, 2004
5. A. Juels, R. L. Rivest and M. Szydlo. The Blocker Tag: Selective Blocking of RFID Tags for consumer Privacy. In *Proceeding of 10th ACM Conference on Computer and Communications Security'03*, pp. 103-111, 2003.
6. S. Lee, Y. Hwang, D. Lee and J. Lim. Efficient Authentication for Low-cost RFID Systems. *ICCSA'05*, LNCS 3480, pp. 619-627, Springer-Verlag, 2005
7. M. Ohkubo, K. Suzuki and S. Kinoshita. Hash-Chain Based Forward-Secure Privacy Protection Scheme for Low-Cost RFID. In *proceedings of the SCIS'04*, pp. 719-724, 2004.
8. K. Rhee, J. Kwak, S. Kim and D. Won. Challenge-Response Based on RFID Authentication Protocol for Distributed Database Environment. *SPC'05*, LNCS 3450, Springer-Verlag, 2005.
9. National Institute of Standards and Technilogy(NIST) FIPS-180-2: Secure Hash Standard(SHS), 2002.
10. S. E. Sarma, S. A. Weis and D. W. Engels. Radio-Frequency Identification: Security Risks and Challenges. *RSA Laboratories*, Volume 6, No. 1, Spring, 2003.
11. S. A. Weis. Security and Privacy in Radio-Frequency Identification Devices. MS Thesis, MIT, 2003
12. S. A. Weis, S. E. Sarma, R. L. Rivest and D. W. Engles. Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. *Security in Pervasive Computing'03*, LNCS 2802, Springer-Verlag, 2004.