LRMAP: Lightweight and Resynchronous Mutual Authentication Protocol for RFID System*

JeaCheol Ha[1], JungHoon Ha[2], SangJae Moon[2], 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
boyd@isrc.qut.edu.au

Abstract. Despite various solutions to the security problems in an RFID system, most are unable to fully support all the security requirements. Plus, when designing a viable RFID system, account should also be taken of the computational load on the back-end database and restricted capacity of a tag. Accordingly, an efficient RFID protocol is proposed to reduce the computational load on both the back-end database and the tags, while also guaranteeing most security requirements for RFID wireless communication, including untraceability, authentication, and robustness against replay and spoofing attacks. Plus, in the case of desynchronization resulting from communication failure or malicious attack, the proposed scheme can recover synchronization between the database and the tag.

Keywords: RFID system, Mutual authentication, Privacy, Traceability, Desynchronization attack.

1 Introduction

Radio Frequency Identification (RFID) systems, a new form of automatic identification technology involving the use of small devices called RFID tags, are expected to replace optical barcodes due to several important advantages, including a low cost, small size, quick identification, and invisible implementation within objects. An RFID system consists of RFID tags, an RFID reader, and a back-end database. Yet, since the RFID reader communicates with the tags using RF signals, existing RFID protocols still suffer from various weaknesses, including location privacy, authentication, and resynchronization between two entities. One solution to protect tags from attack is mutual authentication between the

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

F. Stajano et al. (Eds.): ICUCT 2006, LNCS 4412, pp. 80–89, 2007.

tag and the reader. Thus, a lightweight authentication protocol is needed that takes account of the tag's design limitations, restricted implementation cost, and back-end server's capacity.

Several studies have already attempted to resolve the authentication problem between the tag and the reader using physical technologies, including the 'Kill command' [12], 'Active jamming' [5], and 'Blocker tag' [5] approaches. Then, in 2004, Weis *et al.* [10,11,12] proposed a hash-lock protocol and randomized hash-lock protocol as cryptographic solutions. In another approach based on a hash function, Henrici and Müller [3] proposed an ID variation protocol. While this protocol is secure against a replay attack, as the identity of a tag is renewed in each session, location privacy is still compromised, since the tag's response remains constant until the next authentication session when desynchronization occurs [9]. Ohkubo *et al.* [8] also 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, and imposes a heavy burden on the back-end database to authenticate the tag. Rhee *et al.* [9] proposed a challenge-response authentication protocol based on a hash function that is robust against a spoofing attack and replay attack, plus location privacy is also guaranteed. However, the computational load on the back-end database is still heavy when authenticating a tag. The RFID mutual authentication scheme based on synchronized secret information presented by Lee *et al.* [6] also requires many computational operations in the back-end database. Thus, in 2005, Lee *et al.* [7] 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 also vulnerable to a spoofing attack and location-tracing attack when desynchronization occurs. Recently, Dimitriou [2] proposed a lightweight RFID authentication protocol that enforces user privacy and protects against cloning. However, there is no method for recovering synchronization when a state of desynchronization occurs, where one tag blocks any further tag functionality.

Accordingly, this paper proposes a lightweight and resynchronous mutual authentication protocol (LRMAP) for an RFID system. When a desynchronization problem arises between the back-end database and a tag due to communication failure or a malicious attack, the proposed scheme stays robust and recovers the synchronization. In addition, the computational load on the back-end system is efficient, as a different ID searching method is applied according to the state of the previous session. Moreover, the proposed protocol is secure against location tracing, a replay attack, and spoofing attack, plus mutual authentication is guaranteed between the back-end database and an RFID tag.

2 RFID System

2.1 Composition of RFID System

An RFID system typically consists of three elements, such as RFID tags, (*transponders*), the RFID reader (*transceiver*), and back-end database (*Back-end server*).

- **RFID tag.** An RFID tag generally consists of a microchip for computing and a coupling element, such as an antenna, for wireless communication. A passive RFID tag does not possess an on-board power source, but is powered by the electromagnetic waves from the reader. Meanwhile, an active tag contains an on-board power source, such as a battery. In addition, the tags are categorized into several types according to their physical characteristics and application [1].
- **RFID reader.** The RFID reader interrogates the tags through an RF interface, then transmits the collected data to back-end database. The reader can also read and write the tag data. The channel from the reader to a tag, referred to as the *forward* channel, is insecure, as it is based on an air interface. Similarly, the channel from a tag to the reader, known as the *backward* channel, is also insecure.
- **Back-end database.** The back-end database receives data from the reader and provides certain services to a specific tag, such as product and prices information etc. The communication between the reader and the database is considered as a secure channel.

2.2 Security Requirements for RFID System

Since the communication between the reader and a tag is performed using an air interface, the communicated data can easily be tapped by an attacker. Therefore, various requirements are needed for a secure RFID protocol, as identified in previous literature [4,6,10].

- **Eavesdropping.** An attacker can eavesdrop messages between the reader and tags due to wireless communication, then use secret information or useful messages to perform various enhanced attacks, such as a replay attack or spoofing attack. Therefore, an RFID system should be designed to protect against the leakage of secret information.
- **Spoofing.** An adversary sends a malicious query to a targeted tag, then collects the response messages emitted by the tag. Thereafter, the attacker can impersonate the reader using the messages collected from the tag. On the other hand, an adversary can reply to the reader's query by impersonating a tag.
- **Location tracking.** The adversary seeks information on a tag's location track information. Thus, for perfect location privacy, an RFID system should satisfy both indistinguishability and forward security, where the former means that the values emitted by one tag should not be distinguishable from the values emitted by other tags, while the latter means even if an attacker obtains the secret data stored in a tag, the location of the tag can not be traced back using previous known messages, *i.e.*, disclosed data or communication information.
- **Message Interrupt.** The communication messages between the tags and the reader can be interrupted when an attacker tries to block the service. As a result, a message interrupt attack can create a state of desynchronization between the tag and the back-end database, due to an abnormal closing of a session, message blocking, or different ID updating of two entities within one session.

3 Related Work

3.1 *ID* Variation Protocol

Henrici and Müller [3] proposed an ID variation protocol that changes the identity of a tag in each session. Although this protocol is secure against a replay attack, as the ID of a tag is refreshed in each session by a random number, a spoofing attack can be applied, where an attacker impersonates the reader. Meanwhile, for location tracking, the attacker does not transmit the last message of the protocol, then since the tag then thinks that the information is lost, it does not update its ID [9]. As a result, the protocol has a database desynchronization problem. If the ID of a tag is desynchronized, the tag can be easily traced, as one of emitted values of the tag $H(ID)$ will be identical, thereby compromising the location privacy. This is called a ***desynchronization attack*** in which the attacker traces the tag's location using successive desynchronizations.

3.2 Challenge-Response-Based Authentication Protocol

Rhee *et al.* [9] proposed a challenge-response authentication protocol based on a hash function. This scheme is robust against a spoofing attack and replay attack. In addition, location privacy is guaranteed, as the tag transmits a different response in each session using a random number received from the reader. Nonetheless, the scheme is inefficient in terms of the computational load, as the back-end database is required to perform an ID search to find the specific information related to the tag requesting authentication.

3.3 Low-Cost Authentication Protocol: LCAP

Lee *et al.* [7] proposed a low-cost authentication protocol, LCAP, that only involves two one-way hash function operations in a tag, making it quite efficient. Although location privacy is supposedly guaranteed, the scheme is still vulnerable to location tracing, as a tag will respond to the same $H(ID)$ in the case the last message from reader is not received due to a message interrupt. Therefore, this protocol is vulnerable to location tracing using successive desynchronization attacks.

3.4 Lightweight Challenge-Response Protocol

Recently, Dimitriou [2] proposed a lightweight RFID authentication protocol that enforces user privacy and protects against cloning. However, an attacker can still block the final message transmitted from the reader to the tag. In the resulting state of desynchronization, the tag and back-end database update using different keys, thereby blocking any further tag functionality. In addition, an attacker can trace a tag by repeatedly sending a query from the reader. As a tag will respond with the same message $H(ID_i)$ in which ID_i is fixed in a desynchronized session, the tag cannot satisfy indistinguishability.

4 Proposed Authentication Protocol: LRMAP

This section presents the proposed lightweight and resynchronous mutual authentication protocol (LRMAP) for an RFID system.

4.1 Notations

The following notations are used for the entities and computational operations to simplify the description.

T	: RFID tag or transponder
R	: RFID reader or transceiver
DB	: back-end database or back-end server
ID	: identity of a tag, L bits
HID	: hashed value of ID, L bits
PID	: previous identity of a tag used in previous session, L bits
r_R	: random number generated by reader R
r_T	: random number generated by tag T
$Query$	: request generated by R
$SYNC$	: parameter used to check whether both T and DB succeeded in ID updating simultaneously or not, 1 bit
$H()$	: one-way hash function, $H : \{0,1\}^* \rightarrow \{0,1\}^l$
$L(m)$	: left half of input message m
$R(m)$	: right half of input message m
$\|\|$	: concatenation of two inputs
$\stackrel{?}{=}$	: comparison of two inputs

4.2 System Model and Protocol

To define the model of the proposed lightweight and resynchronous mutual authentication protocol (LRMAP), the RFID system consists of three entities, the tag T, reader R, and back-end database DB. T emits $P = H(ID)$ or $P = H(ID\|r_T)$ according to the state of $SYNC$ in response to a query from R. That is, if T does not receive the last message from R due to a communication malfunction or the verification procedure fails due to a malicious attack, the $SYNC$ value is set as 1 and T responds with $P = H(ID\|r_T)$ in the next session. In the case the protocol finishes normally, the $SYNC$ value becomes 0 and T transmits $P = H(ID)$ to R in the next session. DB manages the ID, hashed values HID, and PID for each T in the database field. According to the state of the previous session, *i.e.*, the value P received from T, DB finds ID for the current session or PID used for the previous session by comparing the received P with the HID and PID in the database field. It is assumed that the communication channel between R and DB is secure, while the communication channel between R and T is insecure. Fig. 1 shows the

Database	Reader	Tag
Database field [ID][HID][PID]		Tag field [ID][SYNC]
	$\xrightarrow{Query,\ r_R}$	If($SYNC \stackrel{?}{=} 0$)$P = H(ID)$ else $P = H(ID\|\|r_T)$ $Q = H(ID\|\|r_T\|\|r_R)$
If($P \stackrel{?}{=} HID$)$PID = ID$ else if($P \stackrel{?}{=} H(ID\|\|r_T)$ $PID = ID$ else if($P \stackrel{?}{=} H(PID\|\|r_T)$ $PID = PID$ else halt $Q' = H(PID\|\|r_T\|\|r_R)$	$\xleftarrow{P,L(Q),r_T,r_R}$	$\xleftarrow{P,L(Q),r_T}$ $SYNC = 1$
$L(Q') \stackrel{?}{=} L(Q)$ $ID = H(PID\|\|r_R)$ $HID = H(ID)$	$\xrightarrow{R(Q')}$	$\xrightarrow{R(Q')}$ If($R(Q) \stackrel{?}{=} R(Q')$) { $ID = H(ID\|\|r_R)$ $SYNC = 0$ }

Fig. 1. The proposed lightweight and resynchronous mutual authentication protocol

process of the proposed LRMAP, and the following gives a detailed description of each step:

1. R chooses a random number r_R and broadcasts it to T with a $Query$.
2. T selects a random number r_T and computes P differently according to the state of $SYNC$. That is, if the $SYNC$ value is 0, then $P = H(ID)$, otherwise T computes $P = H(ID\|r_T)$ using r_T generated by itself. It then computes $Q = H(ID\|r_T\|r_R)$ and sets the $SYNC$ field as 1. T transmits $P, L(Q)$ and r_T to R in response to the $Query$, where $L(Q)$ is the left half of Q.
3. R forwards the message $P, L(Q)$ and r_T received from T to DB together with r_R generated by itself in step 1.
4. DB firstly compares the received $P = H(ID)$ with the HID values saved in the database. If the values match, DB regards the ID as the identity of T requesting authentication. This is a general case when the previous session is closed normally. If DB cannot find the HID in the first searching case then it secondly computes $H(ID\|r_T)$ value with the received r_T and compares it with P. If the tag's response messages were blocked in the previous session, that is, the $SYNC$ value will be 1 and two IDs in the DB and tag will not be updated, then the DB finds a match with the ID of T in the second searching case. However, if DB cannot find the ID of tag in above two cases, then it thirdly computes $H(PID\|r_T)$ value and compares it with P. The DB finds a match with the PID of T when the reader's last messages were blocked in the previous session, that is, the $SYNC$ value will be 1 and DB will update the ID, yet the tag's ID will not be updated. Unfortunately, if DB cannot find the identity of

T in above three cases, it halts the searching of ID and can order R to query again in order to restart the process from the first step. If DB finds the ID or PID in three searching cases, then it computes $Q' = H(PID\|r_T\|r_R)$[1] and verifies that the following equation is satisfied:

$$L(Q') \stackrel{?}{=} L(Q). \tag{1}$$

If equation (1) is satisfied, DB computes $R(Q')$, transmits it to R, and updates the HID for the next session. That is, it computes $ID = H(PID\|r_R)$ and updates the $HID = H(ID)$.

5. R delivers the message $R(Q')$ received from DB to T.
6. To verify the correctness of $R(Q')$, T tests the following equation:

$$R(Q) \stackrel{?}{=} R(Q'), \tag{2}$$

where $R(Q)$ is the right half of $Q = H(ID\|r_T\|r_R)$ computed by itself in step 1. If equation (2) is correct, T updates the identity as $ID = H(ID\|r_R)$, then sets the $SYNC$ value at 0.

5 Analysis

5.1 Security

The security of the proposed LRMAP was evaluated against the threats described in Section 2.

- **Eavesdropping.** To obtain secret information from a tag, an adversary must be able to guess the ID after collecting the communication messages. However, an adversary cannot extract the ID from the $H(ID)$ or $H(ID||r_T)$ due to the security property of a one-way hash function. Otherwise, the adversary has to compute a correct string $L(Q)$ from a known r_T and r_R, which is also hard due to their one-way property. A replay attack cannot compromise the proposed protocol, as the $H(ID)$ or $H(ID||r_T)$ is refreshed by updating the ID or including a random number r_T in each session. Therefore, the proposed LRMAP can defeat a replay attack due to the freshness of the communication messages.
- **Spoofing.** Here, an adversary collects a tag's response, then tries a spoofing attack to impersonate a legitimate tag. However, an adversary cannot compute the hashed messages P and $L(Q)$ without knowing the ID value. Meanwhile, to impersonate as the reader, an adversary must transmit the correct $R(Q)$. This is also impossible, because an adversary cannot compute Q without knowing the ID. Thus, it is impossible to impersonate a tag or the reader using a spoofing attack.

[1] Since ID is updated into PID after finding ID from HID, $Q' = H(PID\|r_T\|r_R)$ is computed regardless of PID or ID.

- **Location tracking.** The proposed protocol guarantees location privacy by refreshing the ID in the tag and back-end database for each session. After the successful authentication is finished in the previous session, the $SYNC$ value is set at 0. Thus, indistinguishability is satisfied with a one-way hash function in which the input of the previous session is refreshed. In contrast, if the previous session is not closed normally, the $SYNC$ value is set at 1. Here, indistinguishability is also satisfied using a one-way hash function, as the input is refreshed by a random number r_T. That is, the value P transmitted from the tag is not $H(ID)$ but $H(ID\|r_T)$. As regards forward security, this assumes that an attacker can obtain a tag's correct ID at some point. However, no previous ID can be extracted due to the one-way property of a hash function. That is, it is impossible to recover the ID from $H(ID\|r_R)$, making it impossible for an attacker to trace the location of a tag backwards. Unfortunately, this protocol may be impossible to satisfy forward security while successive desynchronizations are occurred. An adversary can collect the communication messages and continuously make last massage $R(Q')$ invalid up to the time obtaining a target secret ID. After obtaining the secret ID of tag, the adversary may make it possible to trace the some past histories of T while ID of tag was not changed because he knows the previous P and r_T. Therefore, LRMAP perfectly satisfies the forward security property from setup time to the latest point occurred a successful authentication.
- **Message Interrupt.** In the first case, it is assumed that an adversary can block the response messages transmitted from a tag, *i.e.*, step 2 of LRMAP. At this point, as the reader does not know of the tag's existence, the $SYNC$ value for the tag is set at 1, plus, if the tag does not receive a response from the reader within a predefined time, the tag sends $H(ID||r_T)$ as a response to a query from the reader in the next session. Nonetheless, the two entities T and DB can still recover the synchronization by finding the current ID in the back-end database. In the second case, if an attacker blocks the last messages transmitted from the reader, the DB already knows of the tag's existence and updates the ID value, while the $SYNC$ value for the tag is set at 1. Therefore, when a tag sends $H(ID||r_T)$ as the response in the next desynchronized session, the two entities can recover the synchronization based on finding the PID in the back-end database. Therefore, LRMAP can be protected against messages loss due to an attacker in a wireless channel.

A security comparison with previous authentication protocols is shown in Table 1. The proposed LRMAP is secure against most attacks presented up to now, including a replay attack, spoofing attack, location tracing attack, and desynchronization attack.

5.2 Efficiency

When evaluating the computational load and storage cost for the two entities, as shown in Table 1, the LRMAP exhibited a remarkable improvement in the computational cost for the DB. Even though the challenge-response-based protocol

Table 1. Comparison of security and efficiency

Protocol	Henrici[3]	Rhee[9]	Lee[7]	Dimitriou[2]	LRMAP
Replay attack	O	O	O	O	O
Spoofing attack	×	O	×	O	O
Indistinguishability	×	O	×	×	O
Forward security	△	×	×	△	△
Resynchronization	O	O	O	×	O
ID refreshment	O	×	O	O	O
Comp.(hash # of DB)	3	$m/2+2$	2	4	3*
Comp.(hash # of tag)	3	2	2	4	3
Storage of DB(bits)	$8L \cdot m$	$L \cdot m$	$6L \cdot m$	$2L \cdot m$	$3L \cdot m$
Storage of tag(bits)	$3L$	$1L$	$1L$	$1L$	$1L+1$

*: $m+3$ to recover the synchronization on average.
O: secure or support, △: partially secure ×: insecure or not support,
m: the number of IDs.

[9] satisfies most security items, except forward security, its critical disadvantage is that the DB is required to performs $m/2+2$ hash operations to authenticate a tag. In contrast, the proposed protocol only requires 3 hash operations in the DB and tag, respectively, even though m is large. In the case of desynchronization, the correct ID or PID can be found based on an average of $m/2+3$ or $m+m/2+3$ hash operations. So we can say that the recovery time in desynchronization state is $m+3$ operations on average. However, since desynchronization of a tag is a special and unusual state, the normal synchronization state only requires 3 hash operations.

With the proposed protocol, the storage size of the DB is $3L * m$, where L is the length of an ID or hashed value and m is the number of IDs. Plus, a tag needs $(L+1)$-bits of memory to store an ID and the $SYNC$ value. The length of the total message transmitted from a tag to the reader is $2.5L$, while that from the reader to a tag is $1.5L$, except for a $Query$. Therefore, the LRMAP is suitable for an RFID systems with limited memory space and computational power.

6 Conclusion

A lightweight and resynchronous mutual authentication protocol (LRMAP) was proposed to protect an RFID system against existing attacks. The proposed protocol guarantees untraceability, authentication, and robustness against replay and spoofing attacks. Furthermore, even though the protocol can fall into a desynchronized state as a result of a malicious attacker, synchronization between the database and a tag can be recovered in the next session. As the regards the computational cost, the LRMAP is designed to reduce the computational load on both the back-end database and the tags. Consequently, the proposed scheme can be used in low-cost RFID systems, as in a normal state, the correct ID is found using a comparison of the transmitted hash message with the hashed values in the DB.

References

1. Auto-ID Center. Draft Protocol Specification for a Class 0 Radio Frequency Identification Tag, February, 2003.
2. T. Dimitriou, A lightweight RFID protocol to protect against traceability and cloning attacks. Security and Privacy for Emerging Areas in Communications Networks-2005. SecureComm 2005, pp. 59-66, Sept., 2005
3. D. Henrici and P. Müller. Hash-based Enhancement of Loaction Privacy for Radio Frequency Identification Devices using Varing Identifiers, In *proceeding of the Second IEEE Annual Conference on Pervasive Computing and Communiations Workshops*, pp. 149-162, IEEE, 2004
4. A. Juels. RFID Security and Privacy: A Research Survey. *RSA Laboratories*, 2005.
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, T. Asano and K. Kim. RFID Mutual Authentication Scheme based on Synchronized Secret Information. In *proceedings of the SCIS'06*, 2006.
7. S. Lee, Y. Hwang, D. Lee and J. Lim. Efficient Authentication for Low-cost RFID Systems. *ICCSA'05*, NCS 3480, pp. 619-627, Springer-Verlag, 2005
8. 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.
9. K. Rhee, J. Kwak, S. Kim and D. Won. Challenge-Response Based on RFID Authentication Protocol for Distributed Database Envirionment. *SPC'05*, LNCS 3450, Springer-Verlag, 2005.
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.