Low-Cost and Strong-Security RFID Authentication Protocol*

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
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. This paper proposes a low-cost and strong-security RFID protocol to reduce the computational load on both the back-end database and the tags in an RFID system. When desynchronization occurs as a result of a communication failure or malicious attack, the proposed protocol can recover synchronization between the database and the tag in the following session. Furthermore, the proposed protocol also satisfies most security requirements, including the strong privacy property defined by Juels and Weis, plus robustness against replay and spoofing attacks and forward security.

Keywords: RFID system, authentication, indistinguishability, traceability, strong-privacy.

1 Introduction

Radio Frequency Identification (RFID) systems are expected to replace optical barcodes due to many important advantages, such as their low cost, small size, fast identification, and invisible implementation within objects. An RFID system consists of three parts: RFID tags, an RFID reader, and back-end database. Yet, since the RFID reader communicates with the tags using RF interfaces, this insecure channel leaves an RFID system vulnerable to various attacks, such as eavesdropping, spoofing, replay attacks, traceability, and message interrupt attacks. Although a lot of research has already focused on solving the security problems of RFID systems, some existing RFID protocols still suffer from various security weaknesses, including authentication, location privacy, and resynchronization between two entities.

One solution to protect tags from these threats is secure authentication between the tag and the reader. However, due to tag's computational power and

* This research was supported by the MIC of Korea, under the ITRC support program supervised by the IITA(IITA-2007-C1090-0701-0026).

M. Denko et al. (Eds.): EUC Workshops 2007, LNCS 4809, pp. 795–807, 2007.
© IFIP International Federation for Information Processing 2007

storage space, a low-cost authentication protocol is needed that takes account of the back-end server's capacity and tag's implementation limitations.

Initial attempts to resolve the RFID authentication problem between the tag and the reader involved physical technologies and included the 'Kill command' [11], 'Active jamming' [5], and 'Blocker tag' [5] approaches. Thereafter, Weis *et al.* [9, 10, 11] proposed a hash-lock protocol and randomized hash-lock protocol as cryptographic solutions. However, in the randomized hash-lock protocol, the identity of a tag, ID_k, is transmitted from the reader to the tag through an insecure channel in the final step of authentication, making it vulnerable to a replay attack, spoofing attack, and location tracing. Meanwhile, Henrici and Müller [2] proposed an ID variation protocol based on a hash function, making it secure against a replay attack, as the identity of a tag is refreshed in each session, yet location tracing is still compromised, as the tag's response remains constant until the next authentication session when desynchronization occurs [8]. Dimitriou [1] also 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. More recently, Juels and Weis [4] suggested improvements to their hash-lock protocol and presented a simple, formal definition of strong privacy. While their scheme is now robust against several attacks, the computational load on the back-end database is heavy when authenticating a tag. In 2006, Lee *et al.* [6] proposed an RFID mutual authentication scheme that introduced forward security(or forward traceability) to an RFID system, then proved that their scheme was perfectly indistinguishable and almost forward secure. However, the computational load on the back-end database is still heavy when finding a specific tag's ID. The Advanced Semi-Randomized Access Control(A-SRAC) proposed by Lee and Verbauwhede [7] resolves the location tracing problem, forward security, replay attacks, and so on. Yet, this protocol is vulnerable to location tracing due to the constant response of a tag in the case of successive desynchronization attacks.

Accordingly, this paper proposes a low-cost and strong-security mutual authentication protocol for an RFID system. In the case of desynchronization between the back-end database and a tag, the proposed protocol is able to recover the synchronization and maintain a robust security. As the correct ID can be found based on just comparing the transmitted hash message and the hashed values in the database, the computational load on the back-end system is efficient. The proposed protocol is also secure against spoofing attacks, replay attacks, and desynchronization attacks, while also satisfying the strong privacy property recently defined by Juels and Weis [4].

The remainder of this paper is structured as follows. Section 2 explains the security properties of an RFID system. Section 3 then analyzes several existing RFID systems as regards their security and implementation efficiency. The proposed low-cost and strong-security mutual authentication protocol for a secure RFID system is presented in section 4, and its security and efficiency examined in section 5. Some final conclusions are then given in section 6.

2 Security Properties of RFID System

An RFID system usually consists of three elements: RFID tags, the RFID reader, and back-end database. The RFID reader communicates with the tags using an RF signal, then sends the collected message to the back-end database. Unfortunately, the channel between the reader and a tag is insecure, as it is based on wireless communication while the channel between the reader and the database is considered as secure.

2.1 Security Problems

Since the communication between the reader and the tag is performed using a 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 as follows.

- **Information leakage:** A user may not want certain information known by attackers, such as ownership of expensive products, identification of personal medicine, and so on. Therefore, information leakage is a fundamental RFID privacy problem.
- **Spoofing and replay attack:** The attacker can impersonate a legal tag or reader using the messages collected from the tag or replaying certain useful messages.
- **Desynchronization attack:** An adversary can create a dysynchronization state between the tag and the reader by blocking certain transmitted messages. This abnormal state can occurr in an ID-renewable RFID system. If one of emitted values from the tag in dysynchronization state is constant, the tag can be easily traced, thereby compromising the location privacy.
- **Location tracing attack:** 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 emitted from a tag in the current session are identical to those used in the previous session.

2.2 Security Requirements

Various security requirements are needed for secure RFID authentication, as identified in previous literature [3,6,9]. The information leakage problem can be easily solved by using an anonymous ID for each product, then checking whether or not it is in the database. If a tag's ID is always fixed, then it is suitable for a ubiquitous environment, as many separate databases can be used. Conversely, if a tag's ID is renewed in each session, then it is suitable for a single database system due to the ID updating.

To prevent a spoofing attack or replay attack, the protocol should satisfy an authentication requirement. Plus, in the case an adversary has the ability to impersonate a tag or a reader, a mutual authentication protocol is needed. If a

tag's response does not depend on any reader input, as shown in [11], the tag's messages can be used in a replay attack.

One of the aims of a desynchronization attack is to spoil a tag by disturbing the ID search in the database. The other powerful threat is location tracing by successive desynchronization. If an adversary continuously blocks certain legal messages in a wireless channel, a historical trace can be identified. After blocking a message from a tag in a previous session, an adversary can trace a tag by comparing the messages in the current and previous sessions.

Even though an adversary does not know a tag's ID, a target tag can still be traced if some specific tag message patterns are found, *e.g.*, the transmitted data is increased by one in every session, as for a counter. 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.

3 Analysis of Several RFID Authentication Schemes

This section analyzes the problems of existing RFID authentication protocols: (1) protocol developed by Juels and Weis [4], (2) protocol developed by Lee *et al.* based on synchronized secret information [6], (3) lightweight challenge-response RFID authentication protocol(LCRP) of Dimitriou [1], and (4) advanced semi-randomized access control(A-SRAC) scheme of Lee and Verbauwhede [7].

3.1 Randomized Hash-Locks

Juels and Weis [4] recently proposed a simple, formal definition of strong privacy and suggested improvements to their hash-lock protocol. In the improved randomized hash-lock scheme, a reader sends a random number r_R then a tag transmits the value $r_T||h(r_R||r_T||ID)$, where r_T is a random number generated by the tag. The authors insist that their protocol provides strong privacy and can protect against a replay attack. Rhee *et al.* [8] independently proposed a challenge-response authentication protocol based on a hash function that is almost the same as the improved randomized hash-locks scheme. Their scheme is also robust against a spoofing attack, replay attack, and location tracing attack. Nonetheless, the scheme is still vulnerable to forward security, as the ID does not change every session. Plus, their protocol is inefficient in terms of the computational load, as the back-end database is required to perform on average $m/2$ hash operations for an ID search, where m is the number of IDs.

3.2 Scheme Based on Synchronized Secret Information

Lee *et al.* [6] proposed an RFID mutual authentication scheme that utilizes a hash function and synchronized secret information. This scheme offers the most

enhanced security properties with respect to user privacy, including resistance against tag cloning, allowing an additional hash operation. In particular, they introduced forward security(or forward traceability) to an RFID system, then proved that their scheme is perfectly indistinguishable and almost forward secure. However, the back-end database is required to perform about m hash operations to find the specific ID related to a tag.

3.3 Lightweight Challenge-Response Protocol: LCRP

Dimitriou [1] proposed a lightweight challenge-response RFID authentication protocol(LCRP) that guarantees user privacy and protects against cloning. This protocol is based on the use of a secret shared between a tag and the back-end database that is renewed to avoid tag tracing. However, since an attacker can block the final message transmitted from the reader to the tag, it can result in a state of desynchronization. The tag and back-end database update using different keys, as the back-end database renews the secret key, while the tag keeps the old value, which allows an attacker to make the target tag useless. In addition, an attacker can trace a tag by successively sending a query from the reader in a desynchronized state. As the 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. Therefore, this protocol is vulnerable to a location tracing attack.

3.4 Advanced Semi-Randomized Access Control: A-SRAC

Lee and Verbauwhede [7] proposed advanced semi-randomized access control, called A-SRAC, where the tag sends $H(ID)$, r_T, and $H(ID||r_R)$ as a response to the reader. The authors insist that A-SRAC resolves most security properties, such as location tracing, forward security, and replay attacks based on the use of a random number generator in the tags. However, 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 the reader is not received due to a message interrupt, where key in their original paper is the same notation as ID. Therefore, this protocol is vulnerable to location tracing due to the constant response of a tag in the case of successive desynchronization attacks in a second or third pass. Furthermore, if an attacker sends a constant r_R, then a tag will transmit a constant $H(ID||r_R)$, which is used to distinguish it from other tags and trace the tag's location.

3.5 Privacy Vulnerability in LCRP and A-SRAC

Juels and Weis recently proposed a simple, formal definition of strong privacy that is useful for a fundamental analysis of RFID systems [4]. As such, this section applies the definition to check the vulnerabilities of previous protocols. The goal of the adversary in their experiment was to distinguish between two different tags. In other words, if an RFID system does not satisfy strong privacy,

an adversary can distinguish two different tags. They parameterize the number of READERINIT messages sent by an attacker using r, the number of computational steps performed by s, and the number of TAGINIT messages sent by t. In addition, the parameter k is a security parameter, such as the length of ID or a random number. More details are given in [4].

As now explained, the LCRP [1] and A-SRAC [7] protocols are both unfortunately vulnerable to attack as regards strong privacy, as an adversary can send a TAGINIT message and block certain messages in the 2nd or 3rd pass between the tag and the reader. The aim of this blocking is to interrupt the ID updating of the tag. After certain messages are blocked, the target tag can not update its ID value, thus the tag's message, such as $H(ID)$, in next session will be the same as the one generated in the previous session. As a result, an adversary can distinguish the target tag by comparing the messages emitted in the previous and current sessions. The simple adversarial algorithm in Fig. 1 demonstrates that neither of the above two schemes can achieve (r, s, t)-privacy for $t \geq 2$, $s \geq 2$ and $r \geq 1$.

LCRP/A-SRAC Adversarial Algorithm
1. In Phase 1(Learning Phase), adversary selects a pair of distinct tags T_i and T_j uniformly at random.
2. Adversary sends a query together random number to T_i(sends a TAGINIT). Adversary stores some messages and interrupts for ID updating in tag T_i.
3. Adversary submits T_i and T_j as its challenge candidates.
4. In Phase 2(Challenge Phase), adversary initializes a protocol between T_b^* and reader.
5. If adversary can receive a same message with stored one from a tag, he guesses $b = 0$, i.e. $T^* = T_i$. Else he guesses $b = 1$, i.e. $T^* = T_j$.

Fig. 1. Adversarial algorithm for LCRP and A-SRAC

4 Low-Cost and Strong-Security Mutual Authentication Protocol

This section describes the proposed low-cost and strong-security mutual authentication protocol for an RFID system. It is usually assumed that the communication channel between R and DB is secure, while the communication channel between R and T is insecure, as it is based on an air interface.

4.1 Notations

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

Notation	Meaning
T	RFID tag or transponder
R	RFID reader or transceiver
DB	back-end database or back-end server
ID	identity of tag, k bits
HID	hashed value of ID, k bits
PID	previous identity of tag used in previous session, k 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 updating ID simultaneously or not, 1 bit
$H()$	one-way hash function, $H : \{0,1\}^* \rightarrow \{0,1\}^l$
$\|\|$	concatenation of two inputs
$\stackrel{?}{=}$	comparison of two inputs

4.2 Protocol Description

The back-end database DB manages the ID, hashed values HID, and PID for each T in the database field. According to the state of the tag's previous session, the DB finds the ID for the current session or PID used for the previous session by comparing the received P with the HID and PID. After authenticating T, the DB updates the tag's ID and transmits a message of authentication.

An RFID tag T emits $P = H(ID)$ or $P = H(ID\|r_T\|r_R)$ according to the state of $SYNC$ in response to a query from the R. If the T does not receive the last message from the R due to a communication malfunction or the verification procedure fails, the $SYNC$ state is set as 1 and the T responds with $P = H(ID\|r_T\|r_R)$ to the R in the next session. In the case the protocol finishes normally, the $SYNC$ state becomes 0 and the T transmits $P = H(ID)$ in the next session.

The RFID reader R broadcasts a query to a T with a random number r_R and receives information related to authentication from the T, such as hashed values and a random number r_T. The message received from the T is then forwarded to the DB. After the DB authenticates the T, the R transmits the message received from the DB to the T. Fig. 2 shows the process of the proposed low-cost and strong-security protocol, and the following is a detailed description of each step:

1. The R generates a random number r_R and broadcasts it to a T using a $Query$.
2. The T chooses a random number r_T and computes P differently according to the state of $SYNC$. That is, if the $SYNC$ state is 0, then the T computes $P = H(ID)$, otherwise $P = H(ID\|r_T\|r_R)$ using r_T and r_R, and then sets the $SYNC$ state as 1. The T transmits P and r_T to the R, which then forwards the P and r_T messages to the DB together with r_R generated by itself in step 1.

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

Fig. 2. The proposed low-cost and strong-security authentication protocol

3. The DB searches for the specific tag via the received P. First, the DB compares the received $P = H(ID)$ with the HID values saved in the database. If the values match, the DB regards the ID as the identity of the T requesting authentication. This is the general case when the previous session is closed normally. If the DB cannot find the HID in the first search, it computes a $H(ID\|r_T\|r_R)$ value for all the ID and compares it with the P. Thus, if the tag's response messages were blocked in the previous session, that is, the $SYNC$ state is 1 and the IDs in the DB and tag have not been updated, then the DB will find a match with the ID of the T in the second search. However, if the DB cannot find the ID of the tag in the above two cases, it computes a $H(PID\|r_T\|r_R)$ value for all the PID and compares it with the P. Thus, the DB will find a match with the PID of the T if the reader's last messages were blocked in the previous session, that is, the $SYNC$ state is 1 and the DB updated the ID, yet the tag's ID was not updated. If the DB is still unable to find the tag's ID using the above three cases, it halts the search for the ID and orders the R to query again. If the DB does find the ID or PID using one of the three search cases, it authenticates a tag by checking of the existence of an ID. The DB computes $Q = H(PID\|r_T)$[1] and transmits it to the R, then computes $ID = H(PID\|r_R)$ and updates $HID = H(ID)$ for the next session. The R then forwards the message Q to the T.
4. To verify the correctness of Q received from the DB, the T checks the following equation:

$$Q \stackrel{?}{=} H(ID\|r_T). \tag{1}$$

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

If equation (1) is correct, the T updates its ID as $ID = H(ID\|r_R)$, then sets the $SYNC$ state at 0.

5 Security and Efficiency

5.1 Basic Security Analysis

The basic security of the proposed protocol was analyzed against the attacks described in Section 2. To obtain secret information in a tag, an adversary must be able to compute the ID. However, any adversary cannot extract the ID value from $H(ID)$, $H(ID||r_T)$, or $H(ID||r_T||r_R)$ due to the one-way property of a hash function.

An adversary collects a tag's messages, then tries a spoofing attack based on impersonating a legitimate tag. However, an adversary cannot compute the transmitting message P without knowing the ID. On the other hand, to impersonate a reader, an adversary must send the correct Q to the tag. This is also impossible, because an adversary cannot compute it without knowing the ID value. A replay attack also cannot compromise the proposed protocol, as $H(ID)$ or $H(ID||r_T||r_R)$ is refreshed by updating the ID or including random numbers r_T and r_R in each session.

In the case of a desynchronization attack, where message loss occurs due to an adversary, the proposed protocol allows the tag and reader to recover synchronization. In the first case, if the adversary blocks the response messages transmitted from a tag, *i.e.*, step 2 in Fig. 2, plus, if the tag does not receive any correct response from the reader, the $SYNC$ state is set at 1, so the tag will transmit $H(ID||r_T||r_R)$ in the next session. Nonetheless, the two entities can recover their synchronization by searching the correct ID in the back-end database, as the DB stores the ID value. In the second case, if the adversary blocks the message Q which is transmitted from the reader, the DB has already updated the ID, yet the $SYNC$ state is set at 1. Therefore, when the tag transmits $H(ID||r_T||r_R)$ as the response in the next session, the T and DB can still recover synchronization based on finding the PID in the back-end database. Therefore, the proposed protocol can protect against a desynchronization attack.

For location tracing, the proposed protocol also guarantees location privacy based on renewing the ID for each session. After the authentication is completely closed in the previous session, the tag sends $H(ID)$ in response to a query in the current session. Thus, indistinguishability is satisfied as the ID in the previous session has been refreshed using a one-way hash function. In contrast, if the previous session is finished abnormally, the value P transmitted from the tag is $H(ID\|r_T||r_R)$, thus the same response is not emitted by the tag in the subsequent session. Next subsection provides a formal proof of this indistinguishability, which is included in the strong-privacy definition presented by Juels and Weis [4].

In the case of forward security, it is assumed that an attacker obtains a tag's correct ID at some time. However, any previous ID cannot be extracted due

to the security property of the one-way hash function used to update the ID. Consequently, it is impossible for an attacker to trace the location of a tag backwards. However, it is harder to satisfy forward security during a state of successive desynchronization, during which an adversary collects all communication messages until obtaining the target secret ID. In this case, the adversary can trace the past history of the T, as the ID of the tag has not been changed. Nonetheless, the proposed protocol is able to guarantee forward security from the setup time to the latest point of successful ID updating.

5.2 Formal Proof of Strong Privacy(Indistinguishability)

The proposed protocol is able to guarantee strong privacy and is also resistant to other attacks. In particular, the proposed improvement of the LCRP and A-SRAC schemes is powerful for location tracing. Therefore, it is concluded that tags should not emit the same message as used in the previous session. Next, a formal proof is provided for the strong privacy [4] of the proposed protocol.

Theorem 1. ((r, s, t)-Private) *The proposed protocol is (r, s, t)-private in random oracle model, for any polynomially bounded adversary, i. e. any r, s, t polynomial in k.*

Proof: The simulator **Sim** is specified for T_b^* in the privacy experiment Exp^{priv}. Recall that the adversary chooses two challenge tags T_i and T_j. The adversary can collect the message list of P and r_T for a given random number r_R during the learning phase(Phase 1). Let L be the full list of pairs $\{(P, r_T)\}$ output by T_i and T_j. Let $O(\ )$ represent the random oracle for $H(\ )$ in this experiment.

During the challenge phase, **Sim** simulates the result of a TAGINIT call to T_b^* by generating a random number r_T of messages $\{(P, r_T)\}$ and appending then to a list L'. In order for the adversary to distinguish between the simulated challenge phase and a real challenge phase, the adversary should identify a pair $\{(P, r_T)\}$. It is assumed that the two tag IDs have fixed values, to allow them to be distinguished from each other. Consequently, one of the following three cases must occur at some time point during the experiment:

(1) To distinguish **Sim** from T_b^* in the message $P = H(ID)$, the adversary successfully submits to $O(\ )$ a query in the form of ID_i or ID_j, where $O(\ )$ is a random oracle. As the length of the IDs is k-bits, the corresponding space is 2^k. Yet, since the outputs do not reveal any information, the possibility that an adversary can successfully submit a query to $O(\ )$ is at most $2s/2^k$, where s is the number of computational steps for a random oracle.

(2) For a message $P = H(ID||r_T||r_R)$ that is transmitted in a state of desynchronization, the adversary has a success possibility of at most $(r + 2s + t^2)/2^k$ in which the space of P is also 2^k. It is why the random number r_R can be considered as fixed information in the experiment. In fact, the r_R can be intensively determined and transmitted to the tag by an adversary. The proof of possibility is given at [4] in the case of a randomized hash-lock protocol.

Thus, an adversary can distinguish **Sim** from T_b^* with a probability of at most $(r + 4s + t^2)/2^k$, which is negligible for a polynomially bounded adversary. Furthermore, for a successful replay attack, an adversary should guess the reader's random number r_R. Therefore, the success possibility is $1/2^k$, which is also negligible. □

5.3 Comparison of Security

A security comparison with existing authentication protocols is described in Table 1. Most protocols are designed to protect against information leakage, spoofing attacks, and replay attacks. However, the LCRP [1] and A-SRAC schemes [7] do not satisfy the indistinguishability property in the case of a desynchronization attack that interrupts the updating of a tag's ID. This means that these schemes are unable to satisfy the strong privacy defined in [4], as shown in Fig. 1. Meanwhile, the Juels-Weis scheme *et al.* [4] and challenge-response-based protocol [8] do not satisfy forward security, as they use a fixed ID. In existing literature, this security weakness is present in most fixed ID RFID systems. Furthermore, Dimitrio's protocol [1] does not support resynchronization when a desynchronization attack occurs, which is a critical weakness in practical RFID systems. In a desynchronized state, a tag is useless and can not guarantee indistinguishability when a query by a malicious reader is repeatedly generated. In contrast, the proposed protocol is secure against most attacks presented up to now, including replay attacks, spoofing attacks, desynchronization attacks, and location tracing attacks. The proposed protocol also satisfies the strong privacy defined in [4].

Table 1. Comparison of security

Protocol	LCRP [1]	Juels *et al.* [4] [8]	Lee *et al.* [6]	A-SRAC [7]	Proposed
Information leakage	O	O	O	O	O
Spoofing attack	O	O	O	O	O
Replay attack	O	O	O	O	O
Indistinguishability *	×	O	O	×	O
Forward security	△	×	△	△	△
Resynchronization	×	O	O	O	O

O : secure or support △ : partially secure × : insecure or not support.
* : Strong privacy defined in [4].

5.4 Efficiency

When evaluating the computational cost for the two entities, the proposed protocol exhibited a remarkable enhancement for the DB, as shown in Table 2. Even though Lee *et al.* [6] and the challenge-response-based protocol [8] satisfy

most security items, except forward security, their critical disadvantage is that the DB is required to perform $m+3$ or $m/2+2$ hash operations to authenticate a tag, where m is the number of IDs. In contrast, the proposed protocol only requires 3 hash operations by the DB and tag, respectively, and there is no relation with the length of m. In the case of just desynchronization, the correct ID or PID can be found based on an average of $m/2+3$ or $m+m/2+3$ hash operations. As such, the recovery time to a synchronized state is $m+3$ operations on average. However, since desynchronization is a special and abnormal state, a usual synchronized state only requires 3 hash operations, which is a low computational cost compared to other existing protocols. Especially, the first response time of tag is just one hash operation, then the proposed protocol guarantees faster authentication in DB than LCAP or A-SRAC.

With the proposed protocol, since the DB only stores 3 ID-related values for each tag, the storage size of the DB is $3k \cdot m$, where k is the length of an ID or hashed value. Plus, a tag needs $(k+1)$-bits of memory to store its ID and 1-bit $SYNC$ value. Plus, the total amount of messages transmitted from a tag to the reader is $2k$, and that from the reader to a tag is $2k$, except for a $Query$.

Table 2. Comparison of computational and communication efficiency

Protocol	LCRP [1]	Juels *et al.* [4] [8]	Lee *et al.* [6]	A-SRAC [7]	Proposed
Comp.(hash # of DB)	4	$m/2+2$	m+3	4	3^*
Comp.(hash # of tag)	4	2	3	4	3
Storage of DB(bits)	$2k \cdot m$	$k \cdot m$	$3k \cdot m$	$k \cdot m$	$3k \cdot m$
Storage of tag(bits)	k	k	k	k	$k+1$
Communication load	$5k$	$4k^{**}$	$4k$	$6k$	$4k$

m : the number of IDs.
* : $m+3$ to recover the synchronization on average.
** : assuming that the 3-pass mutual authentication in [8] is adopted.

6 Conclusion

A low-cost and strong-security protocol was proposed to protect an RFID system from various existing attacks. The proposed protocol guarantees authentication, robustness against spoofing or replay attacks, and untraceability. Furthermore, even though the protocol can fall into a desynchronized state due to a malicious attacker, in which the database and a tag have different IDs, synchronization can be recovered in the next session. As regards its strong privacy property, a formal proof of the robustness of the proposed protocol is provided. In conclusion, the proposed protocol can be used in low-cost RFID systems that require a small computational load for both the back-end database and the tags.

References

1. Dimitriou, T.: A lightweight RFID protocol to protect against traceability and cloning attacks. In: SecureComm 2005. Security and Privacy for Emerging Areas in Communications Networks-2005, pp. 59–66 (September 2005)
2. Henrici, D., Müller, P.: 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, Los Alamitos (2004)
3. Juels, A.: RFID: Security and Privacy: A Research Survey. RSA Laboratories (2005)
4. Juels, A., Weis, S.A.: Defining strong privacy for RFID, Cryptology ePrint Archieve, Report 2006/137 Referenced (2006), at `http://eprint.iacr.org`
5. Juels, A., Rivest, R.L., Szydlo, M.: The Blocker Tag: Selective Blocking of RFID Tags for consumer Privacy. In: Proceeding of 10th ACM Conference on Computer and Communications Security 2003, pp. 103–111 (2003)
6. Lee, S., Asano, T., Kim, K.: RFID: Mutual Authentication Scheme based on Synchronized Secret Information. In: Proceedings of the SCIS 2006 (2006)
7. Lee, Y.K., Verbauwhede, I.: Secure and Low-cost RFID Authentication Protocols. In: AWiN. 2nd IEEE International Workshop on Adaptive Wireless Networks (November 2005)
8. Rhee, K., Kwak, J., Kim, S., Won, D.: Challenge-Response Based on RFID Authentication Protocol for Distributed Database Envirionment. In: Hutter, D., Ullmann, M. (eds.) SPC 2005. LNCS, vol. 3450, Springer, Heidelberg (2005)
9. Sarma, S.E., Weis, S.A., Engels, D.W.: Radio-Frequency Identification: Security Risks and Challenges. RSA Laboratories 6(1) (Spring 2003)
10. Weis, S.A.: Security and Privacy in Radio-Frequency Identification Devices. MS Thesis, MIT (2003)
11. Weis, S.A., Sarma, S.E., Rivest, R.L., Engles, D.W.: Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems. In: Hutter, D., Müller, G., Stephan, W., Ullmann, M. (eds.) Security in Pervasive Computing. LNCS, vol. 2802, Springer, Heidelberg (2004)