Mutual Authentication Protocol for Low-cost RFID

Jeongkyu Yang[1], Jaemin Park[2], Hyunrok Lee[2], Kui Ren[3], and Kwangjo Kim[2]

[1] Korea Minting & Security Printing Corporation,
35 Gajeong-dong,Yuseong-gu Daejeon 305-713, Korea.
jkyang@komsco.com

[2] Information and Communication University (ICU),
103-6 Munji-Dong, Yuseong-Gu, Daejeon, 305-714, Korea.
{jaeminpark, tank, kkj}@icu.ac.kr

[3] Worcester Polytechnic Institute (WPI),
100 Institute Road, Worcester, MA 01609, U.S.A.
kren@ece.wpi.edu

Abstract— *Radio frequency identification (RFID) is the latest technology to play an important role for object identification as a ubiquitous infrastructure. However, current low-cost RFID tags are highly resource-constrained and cannot support its long-term security, so they have potential risks and may violate privacy for their bearers. To remove security vulnerabilities, we propose a robust mutual authentication protocol between a tag and a back-end server for low-cost RFID system that guarantees data privacy and location privacy of tag bearers. Our protocol firstly provides reader authentication and prevent active attacks based on the assumption that a reader is no more a trusted third party and the communication channel between the reader and the back-end server is insecure like wireless channel. Also, the proposed protocol exhibits forgery resistant against simple copy, or counterfeiting prevailing RFID tags. As tags only have hash function and exclusive-or operation, our proposed protocol is very feasible for low-cost RFID system compared to the previous works. The formal proof of correctness of the proposed authentication protocol is given based on GNY logic.*

Keywords: RFID, tag, reader, back-end server, authentication protocol

1 Introduction

Radio Frequency Identification (RFID) is currently considered as the next generation technology that is mainly used to identify massive objects and will be a substitution for an optical bar code system in the near future. The typical RFID system consists of Radio Frequency (RF) tags, or transponders, and RF tag readers, or transceivers [8, 12]. A back-end server is usually included in RFID system as an individual component [4, 11, 12, 14]. The micro-chip equipped on a tag has a unique identification information and is applicable for various fields such as animal tracking, supply chain management, inventory control, *etc.*

The existing RFID systems are vulnerable to many security risks and imply potential privacy problems, since the implementation of well-known cryptographic algorithms remains hard due to the restricted computational power and the memory size of a low-cost RFID tag [3, 4, 6, 11, 12, 14]. User privacy issues are considered as a big barrier for the proliferation of RFID system applications since the data of a tag can be transmitted by an illegal interrogation without its bearer's attention.

To remove security vulnerabilities, an authentication protocol for RFID systems can be considered as a security measure. As discussed in [1, 3, 11, 14], one of the important issues to provide the security services under RFID environment is to design an authentication protocol keeping the low computational power of RFID tags in mind. In this paper, we propose a robust mutual authentication protocol that fits the low-cost RFID system environment. Our protocol meets the privacy protection for tag bearers, which requires confidentiality, anonymity, and integrity in the cryptographic point of view. The proposed protocol is robust enough against the active attacks such as the man-in-the-middle attack, and the replay attack as well as the data loss [11, 12, 13]. Our protocol is based on mutual authentication between a tag and a back-end server, and provides authentication for the reader in a special case the reader is no more regarded as the trusted third party (TTP). We consider forgery resistance against the attacker who copies or counterfeits a prevailing RFID tag.

The remainder of the paper is organized as follows: In Section 2, we introduce RFID system primer and its related works, and then propose new authentication scheme in Section 3. We discuss the security proof of our scheme and suggest its security and performance in Section 4 and Section 5, respectively.

Finally, we conclude this paper in section 6.

2 Related Works

A hash function is a powerful and yet computational efficient cryptographic tool. Based on the one-wayness of hash function together with authentication process for low-cost RFID system are currently considered as the proper solution in the aspect of security requirements and hardware implementation for low-cost RFID tags. According to [9], a hash function can be implemented with only about 1.7 K-gate.

Weis *et al.* [14] introduced two hash-based authentication schemes; hash-lock scheme and extended hash-lockscheme. Their schemes mutually authenticate a tag and a back-end server, and try to provide the user privacy protection features such as anonymity on a tag's data. However, their proposed protocols are neither private nor secure against eavesdroppers since the attacker can track $metaID$ and $(r, f_s(r) \oplus ID)$ and impersonate the tag to a legitimate reader. Extended hash-lock scheme also has an implementation issue like a random number generator into each tag.

Recently, Henrici and Müller [4] proposed a simple and efficient authentication protocol for low-cost RFID system. Their protocol is based on a hash function embedding in a tag and a random number generator on a back-end server to protect the user information privacy, the user location privacy, and the replay attack. Their scheme also provides a simple method for the data loss. However, this protocol cannot resist against the man-in-the-middle attack. The attacker can be located between a legitimate tag and a legitimate reader and obtain the information from the tag. Thus, the attacker easily can be authenticated by the legitimate reader before the next session.

In the previous schemes, a reader is generally regarded as a TTP without the loss of security. However, the wireless communication channel between a reader and a back-end server can be considered as the insecure channel. Thus, an adversary can impersonate as a legitimate reader. Previous schemes cannot prevent the man-in-the-middle attack when a reader is no more a TTP. Besides, previous results did not clearly denote the linkage between the authentication information and the tag, so forgery is easily enabled with the passive eavesdropping.

3 Our Proposed Protocol

3.1 Notations

We use the notations as summarized in Table 1 to describe the protocol throughout the paper. Like [4], we adopt the similar database structure and the same mechanism to prevent the data loss.

Table 1: Notations

$\mathcal{T}$	RF tag, or transponder.
$\mathcal{R}$	RF tag reader, or transceiver.
$\mathcal{B}$	Back-end server, it has a database.
D	A database of $\mathcal{B}$.
C	Chip serial number that is embedded into $\mathcal{T}$.
$E_k()$	Symmetric-key encryption function with the key, k.
$D_k()$	Symmetric-key decryption function with the key, k.
$h()$	One-way hash function.
$h_k()$	Keyed hash function with the secret key k.
ID	Temporary identification value of $\mathcal{T}$, it is used to make the shared secret k_2 randomized.
ID'	Temporary value to be used to make the shared secret k_1 randomized.
k	Secret key shared between $\mathcal{R}$ and $\mathcal{B}$.
k_1	Shared random secret between $\mathcal{T}$ and $\mathcal{B}$.
k_2	Shared random secret between $\mathcal{T}$ and $\mathcal{B}$.
RNG	Random Number Generator in $\mathcal{R}$.
r	Random number generated by RNG.
S	Keyed one-way hash value of $h_k(r)$.
$\oplus$	Exclusive-or (XOR) function.
$\overset{?}{=}$	Verification operator to check whether the left side is valid for the right side or not.
$\leftarrow$	Update operator from the right side to the left side.
T_1	A field for the shared random secret, k_1.
T_2	A field for the shared random secret, k_2.
AE	A field for the pointer linking a pair of records.
CN	A field for the chip serial number, C, of $\mathcal{T}$.
$DATA$	A field for all other application related data of $\mathcal{T}$.

3.2 Assumptions and Attacking Model

Our protocol works under the natural assumption that $\mathcal{T}$ has a hash function, XOR gate, and the capability to keep state during a single session. The widely acceptable low-cost RFID tags likely require the usage of passive tags [12, 14]. To design our proposed protocol, we assume the low-cost RFID tag is passive and has a re-writable memory like EEPROM with reasonable size like EPC Class 2 of EPC Global [13]. In Crypto 2004, Biham *et al.* [5, 15] showed that collision of SAH0, MD4, MD5, HAVAL-128, and RIPEMD in a special case is easily found. With this in mind, we expect that the cryptographic hash function used in our protocol has the desirable security like preimage resistance, second preimage resistance, and collision avoidance. In our protocol, we assume $\mathcal{T}$ has a hash function. In [9], a hash function unit with block size of 64-bit can be implemented with only about 1.7 K-gate, so it is also assumed that there will be the practical implementation of hash function for the low-cost RFID tag with the desirable security. Like [4, 11], we assume that $\mathcal{T}$ only has its authentication related information. A tag also has a memory for keeping values of ID, k_1, and k_2 to process mutual authentication. The simple structures for the database record and the tag memory are shown in Figure 1. Other required data of $\mathcal{T}$ for an application are stored in the database of $\mathcal{B}$.

In the previous schemes [4, 14], they assumed that $\mathcal{R}$ is a TTP and the communication channel between $\mathcal{R}$ and $\mathcal{B}$ is secure. However, we assume that $\mathcal{R}$ is not a TTP and the communication channel is insecure like the current wireless network. We also assume that k is the secret key for keyed hash function shared between $\mathcal{R}$ and $\mathcal{B}$, and $\mathcal{R}$ and $\mathcal{B}$ has enough capability to manage the symmetric-key cryptosystem and sufficient computational power for encryption and decryption.

To solve the security risks and privacy issues, the following attacking model must be assumed and prevented [4, 12, 13, 14]. However, in our protocol, we do not consider a physical attack like detaching RFID tag physically from a product because it is hard to carry out in public or on a wide scale without detection. We consider the following attacks:

- *Man-in-the-middle attack*: The attackers can impersonate as a legitimate reader and get the information from $\mathcal{T}$, so he can impersonate as the legitimate $\mathcal{T}$ responding to $\mathcal{R}$. Thus, the attacker easily can be authenticated by the legitimate $\mathcal{R}$ before the next session.

- *Replay attack*: The attackers can eavesdrop the response message from $\mathcal{T}$, and retransmit the message to the legitimate $\mathcal{R}$.

- *Forgery*: The simple copy for the information of $\mathcal{T}$ by eavesdropping is enabled by the adversary.

- *Data loss*: The protocol can be damaged from the denial-of-service(DoS) attack, power interruption, and hijacking.

3.3 Security Requirement

To protect the user privacy, we consider the following requirement in cryptographic point of view [13, 11].

- *Data Confidentiality*: The private information of $\mathcal{T}$ must be kept secure to guarantee user privacy. The information of $\mathcal{T}$ must be meaningless for its bearer even though it is eavesdropped by an unauthorized $\mathcal{R}$.

- *Tag Anonymity*: Although the data of $\mathcal{T}$ is encrypted, the unique identification information of $\mathcal{T}$ is exposed since the encrypted data is constant. An attacker can identify each $\mathcal{T}$ with its constant encrypted data. Therefore, it is important to make the information of $\mathcal{T}$ anonymous.

- *Data Integrity*: If the memory of $\mathcal{T}$ is rewritable, forgery and data modification will happen. Thus, the linkage between the authentication information and $\mathcal{T}$ itself must be given in order to prevent the simple copy for $\mathcal{T}$. On the other hand, the data loss will happen from the DoS attack, power interruption, message hijacking, *etc.* Thus, the authentication information between $\mathcal{T}$ and $\mathcal{B}$ must be delivered without any failure, and the data recovery must be provided.

Besides, we must consider and evaluate the following security feature in the design of RFID authentication protocol.

- *Mutual authentication and reader authentication*: In addition to access control, the mutual authentication between $\mathcal{T}$ and $\mathcal{B}$ must be provided as a measure of trust. By authenticating mutually, the replay attack and the man-in-the-middle attack to both $\mathcal{T}$ and $\mathcal{B}$ is prevented. $\mathcal{B}$ also must authenticate $\mathcal{R}$ to avoid the man-in-the-middle attack by an illegitimate $\mathcal{R}$ over the insecure channel.

3.4 Protocol Design

The overall protocol is shown in Figure 1. The detailed procedures for each step are described.

3.4.1 Initial Setup

1) Each $\mathcal{T}$ is given two fresh random secrets and a database, D, of $\mathcal{B}$ also stores them as the shared secret. The temporary used two shared secrets are k_1 and $k_2 \in_U \{0,1\}^l$. $\mathcal{T}$ has a hash function and a XOR function. $\mathcal{T}$ does not need to have the additional storage for its serial number, C, since C is unique and permanently embedded into each $\mathcal{T}$ [8]. The initial identification data, $h(k_1)$, k_1, and k_2 are initially stored into ID, k_1, and k_2 of each $\mathcal{T}$'s memory, respectively.

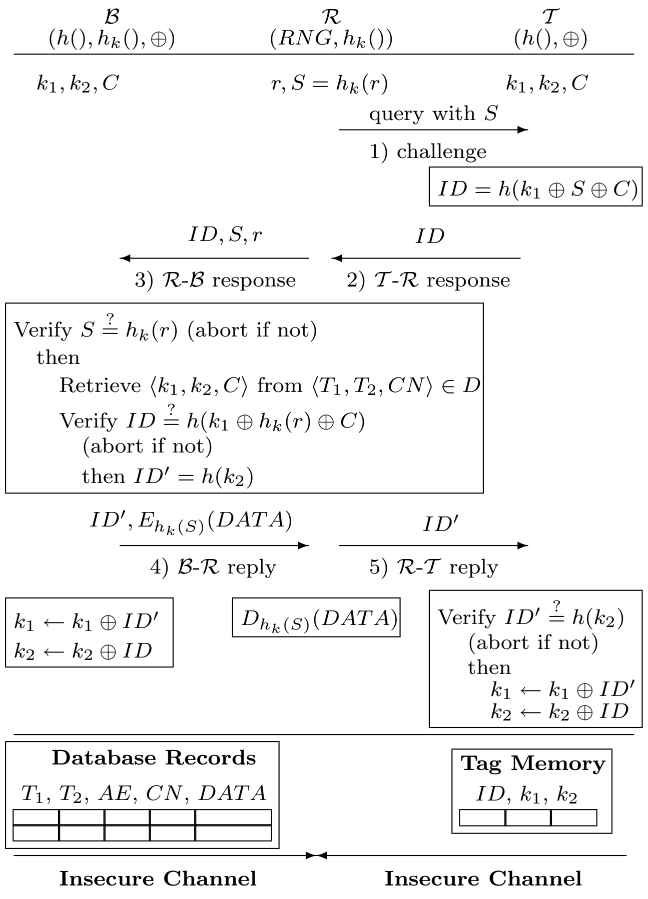
2) $\mathcal{R}$ has a RNG with a keyed hash function, generates a fresh random nonce, $r \in_U \{0,1\}^l$, and calculates $h_k(r)$ for every session. $\mathcal{R}$ and $\mathcal{B}$ manage the secret key k for keyed hash function. We simply denote $h_k(r)$ by S.

3) The database, D, of $\mathcal{B}$ manages a record pair for each tag consisting of $\langle T_1, T_2, AE, CN, DATA \rangle$ like [4]. AE is not set since no associated entry exists initially at this moment. CN, keeps

the unique chip serial number, C, for each $\mathcal{T}$. $\mathcal{B}$ has a hash function and a keyed hash function to verify $\mathcal{T}$ and $\mathcal{R}$, respectively. The pair of records point each other with the pointer field, AE.

3.4.2 Detailed Description

We describe the proposed protocol according to the sequence of message exchange and also discuss the security goals that are achieved during the execution of each protocol message.



$\mathcal{B}$ $(h(), h_k(), \oplus)$
$\mathcal{R}$ $(RNG, h_k())$
$\mathcal{T}$ $(h(), \oplus)$
k_1, k_2, C
$r, S = h_k(r)$
k_1, k_2, C
query with S
1) challenge
$ID = h(k_1 \oplus S \oplus C)$
ID, S, r
ID
3) $\mathcal{R}$-$\mathcal{B}$ response
2) $\mathcal{T}$-$\mathcal{R}$ response
Verify $S \overset{?}{=} h_k(r)$ (abort if not) then Retrieve $\langle k_1, k_2, C\rangle$ from $\langle T_1, T_2, CN\rangle \in D$ Verify $ID \overset{?}{=} h(k_1 \oplus h_k(r) \oplus C)$ (abort if not) then $ID' = h(k_2)$
$ID', E_{h_k(S)}(DATA)$
ID'
4) $\mathcal{B}$-$\mathcal{R}$ reply
5) $\mathcal{R}$-$\mathcal{T}$ reply
$k_1 \leftarrow k_1 \oplus ID'$ $k_2 \leftarrow k_2 \oplus ID$
$D_{h_k(S)}(DATA)$
Verify $ID' \overset{?}{=} h(k_2)$ (abort if not) then $k_1 \leftarrow k_1 \oplus ID'$ $k_2 \leftarrow k_2 \oplus ID$
Database Records
T_1, T_2, AE, CN, $DATA$
Tag Memory
ID, k_1, k_2
Insecure Channel
Insecure Channel

Figure 1: Proposed Authentication Protocol

Step 1 (Challenge) In this step, $\mathcal{R}$ usually applies a collision-avoidance protocol like the secure binary tree walking [2, 13] or the standard protocols of ISO 18000-3 MODE [7] to singularize $\mathcal{T}$ out of many. $\mathcal{R}$ generates a fresh random nonce, r, and randomizes it with the keyed one-way hash function, $S = h_k(r)$. $\mathcal{R}$ sends S to the queried $\mathcal{T}$. The key, k, is shared by $\mathcal{R}$ and $\mathcal{B}$, and S is used to authenticate the validity of $\mathcal{R}$. With S, the man-in-the-middle attack is prevented against an active attacker. It is also used to detect the illegitimate $\mathcal{R}$ by $\mathcal{B}$ after step 3.

Step 2 ($\mathcal{T}$-$\mathcal{R}$ Response) When queried, $\mathcal{T}$ sends ID to $\mathcal{R}$. ID is the output of one-way hash function and used as the identification information. ID has two purposes: One is to verify the legitimate $\mathcal{R}$ with S, and another is to prevent the forgery with C by the passive eavesdropping. ID is randomized with the shared secrets, k_1 and k_2 for every read attempt.

Step 3 ($\mathcal{R}$-$\mathcal{B}$ Response) $\mathcal{R}$ simply forwards ID to $\mathcal{B}$. At the same time, $\mathcal{R}$ also transmits S and r to prevent the man-in-the-middle attack and to detect the illegal $\mathcal{R}$. Within this step, $\mathcal{B}$ authenticates $\mathcal{R}$ and $\mathcal{T}$ consequently with ID.

At first, $\mathcal{B}$ verifies whether the forwarded r is valid or not by comparing S with $h_k(r)$. k is the shared secret key only between $\mathcal{R}$ and $\mathcal{B}$, so $\mathcal{B}$ can detect the illegal $\mathcal{R}$ and discards the forwarded message. So, the man-in the middle attack by the illegitimate $\mathcal{R}$ and a passive eavesdropper can be prevented.

If $\mathcal{R}$ is valid, $\mathcal{B}$ retrieves the records corresponding to ID and get k_1, k_2, and C from T_1, T_2, and CN, respectively. Then, $\mathcal{B}$ authenticates $\mathcal{T}$ with ID. $\mathcal{B}$ calculates $h(k_1 \oplus h_k(r) \oplus C)$ and compares with ID.

Since $\mathcal{B}$ initially stores the chip serial number, C, $\mathcal{B}$ can evaluate the linkage between the forwarded authentication information ID and $\mathcal{T}$ itself in order to prevent forgery. Forgery can be detected and prevented by $\mathcal{B}$ at this moment.

At the same time, $\mathcal{B}$ can detect and prevent the man-in-the-middle attack since S is used as the factor of the man-in-the-middle attack detection. Similarly, the replay attack can be also detected and prevented simultaneously.

If $\mathcal{B}$ successfully finishes the authentication process, $\mathcal{B}$ generates ID' with its one of shared random secrets k_2. ID' will be used to make the shared secret, k_1, anonymous in the remaining steps.

The database of $\mathcal{B}$ generates a new record to consist of a pair of records and updates with the corresponding record. AE has the value to point the pair of records each other. When errors or the data loss in message for the current session happens the database of $\mathcal{B}$ can refer to the record of the previous session pointed by AE of the current session. Thus, the protocol is reliable for the data recovery against the data loss.

Step 4 ($\mathcal{B}$-$\mathcal{R}$ Reply) $\mathcal{B}$ encrypts the $DATA$ using $h_k(S)$, the randomly created shared secret key between $\mathcal{B}$

and $\mathcal{R}$. Then, $\mathcal{B}$ replies ID' and $E_{h_k}(S)(DATA)$. Then, $\mathcal{B}$ makes its shared two keys, k_1 and k_2, randomized simply by Xoring. The same process will be applied to the next step for making the corresponding shared secrets of $\mathcal{T}$ to be anonymous. After this step, the corresponding decryption process, $D_{h_k}(S)(DATA)$, is processed by $\mathcal{R}$ to get $DATA$. Thus, $DATA$ of $\mathcal{T}$ is securely obtained only by the legitimate $\mathcal{R}$ although the adversary eavesdrops the reply messages on the insecure channel.

Step 5 ($\mathcal{R}$-$\mathcal{T}$ Reply) Like step 3, $\mathcal{R}$ forwards ID' to the corresponding $\mathcal{T}$. Then, $\mathcal{T}$ processes the mutual authentication. $\mathcal{T}$ verifies the forwarded ID', calculates $h(k_2)$ and compares it with ID'. If matched, the mutual authentication is finally succeeded, and $\mathcal{T}$, as the last process, updates the shared secrets k_1 and k_2 simply exclusive-ors with ID and ID', respectively. Otherwise, $\mathcal{T}$ will not updates them in a case the replay attack to $\mathcal{T}$ occurs.

4 Correctness

In this section, we prove the correctness of the proposed protocol based on GNY logic [10]. Specifically, the correctness means that after the protocol execution, the communication parties, $\mathcal{T}$ and $\mathcal{B}$, believe that they are sharing two fresh secrets, k_1 and k_2, and ensure that this belief is confirmed by the other side. In addition to this, two entities, $\mathcal{R}$ and $\mathcal{B}$ should believe that they share the secret keys in a case the communication channel between the two entities is insecure.

In the forthcoming description, we use the conventional notations as follows: T, R, and B are entities, $\mathcal{T}$, $\mathcal{R}$, and $\mathcal{B}$, respectively; K_1^i and K_2^i are shared secrets for i-th session between $\mathcal{T}$ and $\mathcal{B}$. $H()$ is a one-way hash function and $H_K()$ is a one-way keyed hash function; N_R is a random nonce generated by $\mathcal{R}$; K is a shared secret for $H_K()$ and K_{RB} is a shared secret for conventional encryption;m is data; other notations like T1, P1, F1, *etc.* follow the logical postulates of GNY logic [10].

4.1 Formalized Protocol

The conventional notations of the generic type of protocol are not convenient for manipulation in a logic. In this section, we, at first, simplify the protocol and describe it as a generic type. Then, we formalize the generic type of the protocol for verification goals as shown in Table 2.

Table 2: Generic Type of Protocol

Protocol Generic Type:
Msg. 1 $R \to T : H_K(N_R)$
Msg. 2 $T \to R : H(K_1^i \oplus H_K(N_R)), H(K_1^i \oplus H_K(N_R) \oplus C)$
Msg. 3 $R \to B :$
$H(K_1^i \oplus H_K(N_R)), H(K_1^i \oplus H_K(N_R) \oplus C), H_K(N_R), N_R$
Msg. 4 $B \to R : H(K_2^i), \{m\}_{K_{RB}}$
Msg. 5 $R \to T : H(K_2^i)$
Formalized Protocol:
Msg. 1 $T \triangleleft \star(H_K(N_R)) \leadsto R \mid\equiv R \overset{K}{\longleftrightarrow} B$
Msg. 2 $R \triangleleft \star(H(K_1^i \oplus H_K(N_R))) \leadsto T \mid\equiv \phi(H(X))$
Msg. 3 $B \triangleleft \star(H(K_1^i \oplus H_K(N_R))) \leadsto B \mid\equiv R \overset{K}{\longleftrightarrow} B$
Msg. 4 $R \triangleleft \star(H(K_2^i), \{R \overset{K_{RB}}{\longleftrightarrow} B\}_{K_{RB}}) \leadsto B \mid\equiv R \overset{K_{RB}}{\longleftrightarrow} B$
Msg. 5 $T \triangleleft \star(H(K_2^i)) \leadsto T \ni K_2^i$

Table 3: Goals of the Correctness Proof

1. $B \mid\equiv T \mid\sim \sharp(H(K_1^i \oplus H_K(N_R)))$	2. $T \mid\equiv B \mid\sim \sharp(H(K_2^i))$
3. $R \mid\equiv R \overset{K}{\longleftrightarrow} B$	4. $B \mid\equiv R \overset{K}{\longleftrightarrow} B$
5. $R \mid\equiv R \overset{K_{RB}}{\longleftrightarrow} B$	6. $B \mid\equiv R \overset{K_{RB}}{\longleftrightarrow} B$

4.2 Proof Goals and Assumptions

The proof goals of correctness are shown in Table 3. The first two goals, (1) and (2), are for the shared secrets. Those beliefs are to state that two entities shared secrets each other exchange fresh messages. The goals (3-6) are about shared keys between two entities. (3) and (4) are for a keyed hash function to guarantee the validity of reader, and (5) and (6) are for message encryption and decryption based on the symmetric key cryptosystem.

Table 4 shows the initial assumptions for our protocol. Assumptions (1-4) state that $\mathcal{T}$ has a hash function, $\mathcal{B}$ has a hash functions and a keyed hash function, $\mathcal{R}$ has a RNG and a keyed hash function, and the random nonce N_R of $\mathcal{R}$ and the keyed hash value $H_K(N_R)$ are fresh. The next six assumptions (3-8) are for two fresh shared secrets, K_1 and K_2, between $\mathcal{T}$ and $\mathcal{B}$. Assumptions (9) and (10) are based on the assumptions (1-8) and $\mathcal{R}$ must be a trusted entity in the viewpoint of $\mathcal{B}$ since the authentication messages from $\mathcal{T}$ are transmitted via $\mathcal{R}$. The abilities for verifying the hashed authentication message transmitted from $\mathcal{T}$ by $\mathcal{B}$ and from $\mathcal{B}$ by $\mathcal{T}$ respectively are based on assumptions (11-14). Assumptions (15-20) mean that both entities, $\mathcal{R}$ and $\mathcal{B}$, trust each other with those keys, K and K_{RB}.

4.3 Verification

In this section, the formal proof of our protocol is stated. The proof based on GNY logic is processed with the assumptions of Table 4. We strictly follow

Table 4: Initial Assumptions for Proof

1. $T \ni H(X)$	2. $R \ni H_K(X)$
3. $B \ni (H(X), H_K(X))$	4. $T \models \sharp(N_R)$
5. $T \ni (K_1^i, K_2^i)$	6. $B \ni (K_1^i, K_2^i)$
7. $T \models \sharp(K_1^i, K_2^i)$	8. $B \models \sharp(K_1^i, K_2^i)$
9. $T \models T \overset{K_1^i, K_2^i}{\leftrightharpoons} B$	10. $B \models T \overset{K_1^i, K_2^i}{\leftrightharpoons} B$
11. $T \models B \ni (K_1^i, K_2^i, C)$	12. $B \models T \ni (K_1^i, K_2^i, C)$
13. $T \models B \Mapsto T \overset{K_1^i}{\rightleftharpoons} B$	14. $T \models \sharp(H(K_2^i))$
15. $T \models R \Mapsto B \mid\!\sim H(K_2^i)$	16. $B \models T \Mapsto T \overset{K_2^i}{\rightleftharpoons} B$
17. $R \ni (K, K_{RB})$	18. $R \models R \xleftrightarrow{K, K_{RB}} B$
19. $B \ni (K, K_{RB})$	20. $B \models R \xleftrightarrow{K, K_{RB}} B$
21. $B \models R \Mapsto R \xleftrightarrow{K, K_{RB}} B$	22. $R \models B \Mapsto R \xleftrightarrow{K, K_{RB}} B$

the logical postulates of [10]. We refer n is the number of list and denote the list of proof goals of Table 3 by Gn, the list of assumptions of Table 4 by An, and the verification steps by (n). The extensions to messages are the precondition and are valid since they hold when messages are sent as are evident from the initial assumptions.

Message 1 $T \triangleleft \star(H_K(N_R)) \rightsquigarrow R \models R \xleftrightarrow{K} B$

1. $T \triangleleft H_K(N_R)$ /*By T1*/
2. $T \ni H_K(N_R)$ /*By P1*/
3. $T \models \sharp(H(N_R))$ /*By F1*/
4. $T \models \sharp(H_K(N_R))$ /*By (2),F10*/

Message 2 $R \triangleleft \star(H(K_1^i \oplus H_K(N_R))) \rightsquigarrow T \models \phi(H(X))$

5. $R \triangleleft H(K_1^i \oplus H_K(N_R))$ /*By T1*/
6. $R \ni H(K_1^i \oplus H_K(N_R))$ /*By P1*/
7. $R \models \sharp(H(K_1^i \oplus H_K(N_R)))$ /*By F10*/
8. $R \models \phi(H(K_1^i \oplus H_K(N_R)))$ /*R6*/
9. $R \models \sharp(H(K_1^i \oplus H_K(N_R)))$ /*For (7), by A18,(5),(6),(8),I1*/
10. $R \models R \xleftrightarrow{K} B$ /*By A20,A22,J1*/

Message 3 $B \triangleleft \star(H(K_1^i \oplus H_K(N_R))) \rightsquigarrow B \models R \xleftrightarrow{K} B$

11. $B \triangleleft H(K_1^i \oplus H_K(N_R))$ /*By T1*/
12. $B \ni H(K_1^i \oplus H_K(N_R))$ /*By P1*/
13. $B \models \sharp(H(K_1^i \oplus H_K(N_R)))$ /*By A3,A6,F10*/
14. $B \models \phi(H(K_1^i \oplus H_K(N_R)))$ /*For (12), by R6*/
15. $B \models R \mid\!\sim H(K_1^i \oplus H_K(N_R))$
    /*For (13), by A3,A6,A20,(11),(13),(14),I1*/
16. $B \models R \xleftrightarrow{K} B$ /*For A18,A21, by J1*/
17. $B \models T \mid\!\sim H(K_1^i \oplus H_K(N_R))$
    /*For (13), by A3,A6,A10,(11),(13),I3*/
18. $B \models T \mid\!\sim \sharp(H(K_1^i \oplus H_K(N_R)))$ /*For (17), by (13),F1*/

Message 4 $R \triangleleft \star(H(K_2^i), \{R \xleftrightarrow{K_{RB}} B\}_{K_{RB}}) \rightsquigarrow B \models R \xleftrightarrow{K_{RB}} B$

19. $R \triangleleft (H(K_2^i), \{R \xleftrightarrow{K_{RB}} B\}_{K_{RB}})$ /*By T1*/
20. $R \triangleleft H(K_2^i)$ /*By T2*/
21. $R \ni H(K_2^i)$ /*By P1*/
22. $R \models \sharp(H(K_2^i))$ /*By P1*/
23. $R \ni (H(K_2^i), \{R \xleftrightarrow{K_{RB}}\}_{K_{RB}})$ /*For (19), by P1*/
24. $R \models B \mid\!\sim (H(K_2^i), R \xleftrightarrow{K_{RB}} B)$ /*For (19), applying A18,I1*/
25. $R \models B \mid\!\sim R \xleftrightarrow{K_{RB}} B$ /* By I7*/
26. $R \models R \xleftrightarrow{K_{RB}} B$ /*By A20,J1*/

Message 5 $T \triangleleft \star(H(K_2^i)) \rightsquigarrow T \ni K_2^i$

27. $T \triangleleft H(K_2^i)$ /*By T1*/
28. $T \ni H(K_2^i)$ /*By P1*/
29. $T \models \sharp(H(K_2^i))$ /*By A7,F10*/
30. $T \models B \mid\!\sim H(K_2^i)$ /*By A5,A9,(27),I3*/
31. $T \models B \mid\!\sim \sharp(H(K_2^i))$ /*By (29),F1*/

As shown above, the proof goals of Table 3 are accomplished by verification steps (10) for G3, (16) for G4, (18) for G1, and (26) for G5, respectively. We omit the proof for G6 since, for the encrypted message with the key, K_{RB}, there is no further message exchange after this step. That is, the encrypted message of the entity, B, is replied to R and decrypted by R.

5 Evaluation

5.1 Security Analysis

We evaluate our protocol in the view point of the security requirement.

Our protocol guarantees the secure mutual authentication only with the hashed messages, $ID = h(k_1 \oplus S \oplus C)$, $ID' = h(k_2)$, and $S = h_k(r)$, and $\mathcal{T}$ does not store user privacy information. Thus, data confidentiality of tag owners is guaranteed and the user privacy on data is strongly protected. In every session, we use the fresh random nonce as the keys between entities. These keys are randomized and anonymous since they are updated for every read attempt. Thus, tag anonymity is guaranteed and the location privacy of a tag owner is not compromised, either. Based on the mutual authentication, our protocol guarantees the data integrity between $\mathcal{T}$ and $\mathcal{B}$. By using the pair of database records and managing AE as we described in the authentication step 3, our protocol provides the data recovery against the data loss during the authentication processes.

To give the forgery resistance feature, we exclusive-or the embedded chip serial number, C, of $\mathcal{T}$ to the authentication information, ID. C is initially embedded during the chip manufacturing. Whenever $\mathcal{T}$ generates ID, it refers to C, so we can come up with the linkage between ID and $\mathcal{T}$ itself. $\mathcal{B}$ keeps each tag's chip serial number initially and authenticates the ownership of the authentication information for $\mathcal{T}$.

Through the authentication step 1 to step 3, $\mathcal{R}$ sends S to $\mathcal{T}$ and S, r to $\mathcal{B}$ for preventing the man-in-the-middle attack. $\mathcal{B}$ can verify S with the calculation of the keyed hashed value of r transmitted from $\mathcal{R}$. Also, the man-in-the-middle attack by $\mathcal{R}$ as an illegitimate

Table 5: Comparison between Protocols

Protocol	HLS [14]	EHLS [14]	HBVI [4]	Our Scheme
User data confidentiality	×	△	△	○
Tag anonymity	×	△	△	○
Data integrity	△	△	○	○
Mutual authentication	△	△	△	○
Reader authentication	×	×	×	○
Man-in-the-middle attack prevention	△	△	×	○
Replay attack prevention	△	△	○	○
Forgery Resistance	×	×	×	○
Data Recovery	×	×	○	○

†† Notation
○ satisfied △ partially satisfied
× not satisfied

Table 6: Computational Loads and Required Memory

Protocol	Entity	HLS [14]	EHLS [14]	HBVI [4]	Our Scheme
No. of Hash Operation	$\mathcal{T}$	1	2	3	2
	$\mathcal{B}$	¬	n	3	2n
No. of Keyed Hash Operation	$\mathcal{R}$	¬	¬	¬	1
	$\mathcal{B}$	¬	¬	¬	1
No. of RNG Operation	$\mathcal{T}$	¬	1	¬	¬
	$\mathcal{R}$	¬	¬	¬	1
	$\mathcal{B}$	¬	¬	1	¬
No. of Encryption	$\mathcal{B}$	¬	¬	¬	1
No. of Decryption	$\mathcal{R}$	¬	¬	¬	1
Number of Authentication Steps		6	5	5	5
Required Memory Size	$\mathcal{T}$	$1\frac{1}{2}L$	$1L$	$3L$	$2\frac{1}{2}L$
	$\mathcal{R}$	¬	¬	¬	$1\frac{1}{2}L$
	$\mathcal{B}$	$2\frac{1}{2}L$	$1\frac{1}{2}L$	$9L$	$8L$

†† Notation
¬ not required n number of tags
L size of required memory

reader is detected and prevented on the insecure channel between $\mathcal{R}$ and $\mathcal{B}$. The $DATA$ of the corresponding $\mathcal{T}$ is not compromised since it is encrypted by $\mathcal{B}$ and decrypted by $\mathcal{R}$ with the randomly generated secret key, $h_k(S)$, from S of $\mathcal{R}$. The key freshness is also guaranteed for each session. The replay attack for $\mathcal{T}$ and $\mathcal{B}$ is detected and prohibited through the step 3 for $\mathcal{B}$ and the step 5 for $\mathcal{T}$. Table 5 shows the comparison of the security requirements and the possible attacks.

5.2 Performance Analysis

We analyze the performance of the proposed scheme in forms of the following overheads: 1) computation, 2) storage, 3) communication, and 4) cost.

• **Computational Overhead.** $\mathcal{T}$ requires only a hash calculation and a XOR operation and needs three hash calculation. However, the cost of hash calculation at the server side is $2n$, where n is the number of tags. Compared to [4], the cost of our protocol has overheads for $\mathcal{B}$. Meanwhile, in [4], the anonymity of tag is guaranteed only after the authentication is successfully completed. Therefore, the location privacy of tag bearers is compromised until the next session is successfully started. To make the output of $\mathcal{T}$ anonymous for the current session, $\mathcal{B}$ should check for every records of D to authenticate each tag like EHLS [14]. However, note that the reduction of this cost should be needed for the admirable performance.

On the other side, our protocol seems to have encryption and decryption overheads for $\mathcal{R}$ and $\mathcal{B}$. However, those cryptographic tools are needed to secure $DATA$ on the insecure channel. We assume that $\mathcal{R}$ and $\mathcal{B}$ have enough computational power to process encryption and decryption based on the symmetric-key cryptosystem.

• **Storage Overhead.** To compare with the previous protocols, we assume the sizes of all components are L bits, and a RNG and a hash function are $h, h_k : \{0,1\}^* \rightarrow \{0,1\}^{\frac{1}{2}L}$ and $r \in_U \{0,1\}^L$, respectively. In our protocol, $\mathcal{T}$ only has a hash function and XOR function, and the size of the memory is $2\frac{1}{2}L$. Thus, the proposed protocol is light-weight and practical. We exclude the comparison for the application-specified data, $DATA$ since the size of $DATA$ depends on applied applications.

• **Communication Overhead.** The proposed protocol accomplishes mutual authentication between $\mathcal{T}$ and $\mathcal{B}$ requiring five rounds. As we denote in the previous section, some protocols [14, 11] requires three or six rounds. However, their protocol have synchronization problem on authentication data between $\mathcal{T}$ and $\mathcal{B}$. Five rounds is mostly acceptable for a minimum number of mutual authentication in RFID environment. Therefore, the proposed protocol is feasible in the sense of communication overheads.

• **Cost Overhead.** [11, 13] claimed that the number of gates available for security generally cannot exceed 2.5-5 K-gate. In our protocol, only one hash function unit and the storage for XOR operation are needed. If we assume the gates for XOR operation needs several tens of gates, the number of expected gates is less than 2 K-gate. Therefore, the proposed protocol is feasible and practical for low-cost RFID environment.

Table 6 shows the comparison of the computational loads and the required memory size for a single session with previous results [4, 14].

6 Concluding Remarks

In this paper, we proposed a robust RFID mutual authentication protocol for the low-cost RFID environment that is computationally light-weight and anonymously interact between entities. The proposed protocol basically fits the low-cost RFID system environment. The tag only has a hash function with the shared two fresh random secrets of small memory size. With this minimal cryptographic primitive, our protocol provides the mutual authentication between the tag and the back-end server and anonymously interacts. Our protocol is robust enough since it protects the replay attack and man-in-the-middle even when the reader is not a trusted third party and the communication channel is insecure. We add the linkage feature between the tag and its authentication data, so forgery is prohibited. All authentication messages are randomized and the tag only has its unique identification data, so the user data privacy and the location privacy is guaranteed. The formal proof of correctness for the proposed protocol was discussed based on GNY logic.

Acknowledgement

I am grateful to Prof. Adi Shamir of the Weizmann Institute in Israel for his precious comments and kind advice on our protocol during his visit to ICU during Asiacryt2004.

References

[1] Auto-ID Center, "860MHz-960MHz Class I Radio Frequency Identification Tag Radio Frequency & Logical communication Interface Specification Proposed Recommendation Version 1.0.0", Technical Report MIT-AUTOID-TR-007, Nov. 2002.

[2] A. Juels, R.L. Rivest, and M. Szydlo, "The Blocker Tag: Selective Blocking of RFID Tags for Consumer Privacy", *Proc. of 10th ACM Conference on Computer and Communications Security*, pp.103-111, Oct. 2003.

[3] A. Juels and R. Pappu, "Squealing euros: Privacy protection in RFID-enabled banknotes", In Rebecca N. Wright, editor, *Financial Cryptography FC'03*, LNCS, vol.2742, pp.103-121, Le Gosier, Gaudeloupe, French West Indies, Jan. 2003.

[4] D. Henrici and P. Müller, "Hash-based Enhancement of Location Privacy for Radio-Frequency Identification Devices using Varying Identifiers", *PerSec'04 at IEEE PerCom*, pp.149-153, Mar. 2004.

[5] E. Biham and R. Chen, "New results on SHA-0 and SHA-1", Presented at the rump session of *Crypto 2004*.

[6] G. Avoine, "Privacy issues in RFID banknotes protection schemes", *In Sixth Smart Card Research and Advanced Application IFIP Conference* - CARDIS, Toulouse, France, Aug. 2004.

[7] ISO/IEC JTC 1/SC 31/WG 4. "Information technology AIDC techniques - RFID for item management Air interface, Part3: Parameters for air interface communications at 13.56 MHz", Version N681R, Apr. 2004.

[8] K. Finkenzeller, "RFID Handbook Second Edition", Wiley & Sons, 2002.

[9] K. Yüksel, "Universal Hashing for Ultra-Low-Power Cryptographic Hardware Applications", Master's Thesis, Dept. of Electronical Engineering, WPI, 2004.

[10] L. Gong, R. Needham, and R. Yahalom, "Reasoning about Belief in Cryptographic Protocols", *1990 IEEE Computer Scociety Synopsis on Research in Security and Privacy*, pp.234-248, 1990.

[11] M. Ohkubo, K. Suzuki, and S. Kinoshita, "Cryptographic Approach to Privacy-Friendly Tags", *RFID Privacy Workshop 2003*, MIT, MA, USA, Nov. 2003.

[12] S. Sarma, S. Weis, and D. Engels, "RFID Systems and Security and Privacy Implication", Auto-ID Center, 2002.

[13] S. Weis, "Security and Privacy in Radio-Frequency Identification Devices", Master's thesis, MIT, 2003.

[14] S. Weis, S. Sarma, R. Rivest, and D. Engels, "Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems", *Proc. of the 1st Security in Pervasive Computing*, LNCS, vol.2802, pp.201-212, 2004.

[15] X. Wang, X. Lai, D. Feng, and H. Yu, "Collisions for hash functions MD4, MD5, HAVAL-128 and RIPEMD", Presented at the rump session of *Crypto 2004*.