Mobile Information Systems 7 (2011) 201–216
DOI 10.3233/MIS-2011-0118
IOS Press

Design of a secure RFID authentication scheme preceding market transactions

Chin-Ling Chen*
Department of Computer Science and Information Engineering, Chaoyang University of Technology, Taichung, Taiwan

Abstract. In recent years, as RFID reader equipment is ever more widely deployed in handled devices, the importance of security problems among RFID reader, tags and server have obviously gained increased attention. However, there are still many security issues preceding transactions; these issues are well worth discussing. In this paper, we propose a novel authentication scheme, conforming EPC C1G2 standards, at a low implementation cost for market application. In order to achieve mutual authentication, the proposed scheme integrates fingerprint biometrics, related cryptology and a hash function mechanism to ensure the security of the transmitted messages. The proposed scheme also can resist known attacks.

Keywords: Mutual authentication, RFID, security, attack, EPC C1G2 standards

1. Introduction

In recent years, the RFID (Radio Frequency Identification) system has been widely used in many fields. For example, in a market environment a tag is attached to the product. Advances in wireless network technology and the continuously increasing number of users of mobile devices make the latter an ideal channel for offering personalized services [1–4] to mobile users. As RFID reader equipment has been widely deployed in hand-held devices, the importance of successfully dealing with security problems related to RFID readers, tags and servers is inescapable. An RFID system contains three essential components: tag, reader and back-end database [5,6]. When a reader sends a request message to a tag, the tag responds with a message via radio frequency signal. In such an environment, there exists the potential for many latent attacks. Once an RFID reader sends a request signal, the RFID tag will respond to the reader's writing and reading request. In the past, a bar-code system was widely utilized; however, the RFID system has greater accuracy and identifies objects efficiently.

In an RFID system, each RFID tag is assigned a unique identity: the Electronic Product Code (EPC), whereas the bar-code system does not support EPC. For this reason, an RFID system performs well in regard to sales management and accesses the product's information conveniently and efficiently. As a result of the RFID system transmitting messages via radio-frequency, many security and privacy problems arise between the RFID tag and RFID reader. In order to solve the RFID security problems, some scholars [5,7] have proposed a one-way hash function to perform mutual authentication. Due to the limited number of the RFID tag's logic gates, the schemes [7–9] using hash operation and symmetric encryption are infeasible. Other researchers [10,11] have proposed lightweight authentication protocols

*Corresponding author. Tel.: +886 4 23323000; E-mail: clc@mail.cyut.edu.tw.

without extensive cryptography. EPCglobal has issued the EPC class 1 generation 2 standards [12]. The EPC class 1 generation 2 standards restrict the RFID tag to only operate the CRC function, logic operation and to generate random numbers. Some scholars [13,14] have proposed a scheme that conforms to EPC class 1 generation 2 standards for the RFID system. Moreover, mobile RFID readers with CF and SD interface for the industrial Handheld Computer Socket has become a mature technology [15]. Recently, some papers [16,17] also discussed about related security issues. We therefore propose a secure system based on mobile RFID technology [18–20] to design a secure RFID authentication scheme preceding market transactions. A good RFID system must avoid illegal access, defend against known attacks and protect the RFID system. The following security issues are often discussed in relation to RFID systems:

(1) Tag impersonation attack: An attacker impersonates a target tag to interact with a reader and he/she can pass the reader's authentication procedure successfully.
(2) Mobile reader lost attack: A user loses the mobile reader and then a malicious user gets the mobile reader to fake the rule of owner to pass the authentication procedure.
(3) Replay attack: An attacker intercepts the messages between a reader and a tag, and then uses these messages to interact with each other in the next session, thereby allowing him/her to successfully pass the authentication procedure.
(4) Trace attack: An attacker intercepts the messages from a target tag and according to the messages, in the next session knows whether they were sent from the target tag. If the messages intercepted are the same, it means they were sent from the target tag, and the attacker can trace the location of the target tag by the messages.
(5) Forgery attack: An adversary listens to an iteration of communication messages between the reader and the legitimate tag for the attacker to store these messages and the attacker could be able to calculate the next correct communication parameter from the intercepted message and pass the authentication.
(6) Man-in-the-middle attack: The attacker intercepts the communication messages between the tag and the reader. The attacker mimics a transmission role; when the reader wants to query the tag, the attacker will intercept the message from the reader and then transfer it to the tag. When the tag wants to send a response message to the reader, the attacker will intercept the message again and transfer it to the reader. The attacker can hold and modify the messages to transmit them between tags and readers.
(7) Privacy protection: The attacker can reveal the related secret information (such as reader's identity, tag's EPC or other secret values).
(8) Forward secrecy: An attacker traces the messages intercepted from past transactions to infer the target tag's internal secret data.
(9) Mutual authentication: Refers to two parties authenticating each other suitably. In terms of technology, it refers to a client or user authenticating himself/herself to a server and that server authenticating itself to the user in such a way that both parties are assured of the other's identity. When describing online authentication processes, mutual authentication is often referred to as server-to-tag, or server-to-reader authentication.
(10) Anonymity: Anonymity is a result of not having identifying characteristics (such as a name or description of physical appearance) disclosed. For example: the attacker can identify whether the tag is the same tag of the last communication by intercepting the communication messages between tag and reader.

The rest of the paper is organized as follows: We present the preliminary in Section 2. Session 3 shows the detailed procedures of the proposed protocol. We analyze the security for our scheme which

can resist several attacks in Section 4. In Section 5, we also make a mechanism analysis. Finally, we conclude this paper in Section 6.

2. Preliminary

2.1. One-way hash function

A one-way hash function [21] $h : m \rightarrow m'$ is a function $h(m) = m'$ condensing an arbitrary message $m$as an input to a fixed-size message output m' ("e.q. 160 bits"). The hash function is public. We describe its properties as follows:

1. The function h is a one-way property that given m, is easy to compute $h(m) = m'$, but given m', is difficult to compute $h^{-1}(m') = m$.
2. Weak collision resistance. e.g. given m_1, it is infeasible to find m_2 so $h(m_2) = h(m_1)$.
3. Strong collision resistance. e.g. it is not feasible to find any pair (m_1, m_2) so $h(m_1) = h(m_2)$.

2.2. Keyed hash function

Let Kdenote an n-dimensional vector space over GF(2). A keyed-hash function [22] $(h_k : k \in K)$ indexed by a key k so $h_k : m \rightarrow m'$, which maps a key and a second bit string to return a fixed-length string output $h_k(m) = m'$. We describe its properties as follows:

1. Given key k and message m, it is easy to compute $h_k(m)$.
2. Without knowledge of k, it is difficult to find $h_k(m)$ when m is given.
3. Without knowledge of k, it is difficult to find m when $h_k(m)$ is given.
4. Given key k, it is difficult to find two messages m_1 and m_2 so $h_k(m_1) = h_k(m_2)$, but $m_1 \neq m_2$.
5. Given (possibly many) pairs of m and $h_k(m)$, it is difficult to compute k.
6. The function h_k should produce a message digest with at least 128 bits.

3. The proposed scheme

3.1. System framework

Assume that users can use their mobile RFID readers to get the information on tags, and make a desired transaction with cash registers. The membership requirement can be satisfied by the service of the market. There are five parties in our proposed scheme; they are described as follows.

1. Server (S): Stores all the information like mobile reader's ID, privacy information, tags' EPC code, information, etc.
2. Cash Register (CR): A device in the market: when users want to purchase products, they can use their mobile readers to communicate with the server through the cash register for transactions.
3. Mobile Reader (MR): A device which can query tags' information and interact with the cash register and the server to complete a transaction.
4. Tag (T): Attached on the products for users to query the product information; all of the tags conform to EPCglobal class 1 generation 2 standards.
5. User (U): A member of the market who can derive service from the market.

In our scheme, the RFID tags comply with EPCglobal C1G2 standards. Therefore, as mentioned above, the tags can only support CRC function and generate random numbers. We describe the steps as follows:

Step 1: The product's tag, users' mobile reader ID, password, fingerprint are all registered with the server via secure channel.
Step 2: User uses his/her mobile reader to query the tag's information. The server will authenticate the mobile reader and then transmit the tag's information to the mobile reader.
Step 3: User uses his/her mobile reader to update its password and symmetric key with the server.

3.2. Notations

ID_{MRj}:	the j th mobile reader's unique identity
Pw:	the user's password
F_j:	the$j$ th user's fingerprint template
f_j:	the hash value of the user's fingerprint template, where $f_j = h(F_i)$
K_i:	the$i$th tag's unique secret key
PK:	the server's public key
SK:	the server's secret key
SYM_j:	the j th user's mobile reader's symmetric key
EPC_i:	96-bit EPC (Electronic Product Code) of the i th tag
$E_{PK}(m)$:	use the server's public key *PK* to encrypt the message$m$
$D_{SK}(m)$:	use the server's secret key *SK* to decrypt the message m
$E_x[m]$:	use the symmetric key x to encrypt the message m
$D_x[m]$:	use the symmetric key x to decrypt the message m
R_1, R_2, R_3:	the random numbers generated by mobile reader, server and tag, respectively
$h(.)$:	cryptographic one-way hash function [21]
$h_k(.)$:	cryptographic one-way hash function with secret key k [22]
CRC(.):	a Cyclic Redundancy Check (CRC) function
$A? = B$:	comparing whether A is equal to B
$\|\|$:	concatenation operation
$\oplus$:	exclusive-or operation
$+$:	addition operation
$-$:	subtraction operation
$\wedge$:	AND logical operation
Tmark:	the market trademark; the *Tmark* is updated regularly
M:	the hash value of the market trademark, where $M = h(T - mark)$ is known as the server and market tags
Info.:	detailed information of product (includes specifications, price, transaction serial number and coupon, etc.)

Our scheme is divided into the following phases:

3.3. Registration phase

In this phase, users have to use the mobile reader to register with the server for the market service, and the server will store the authentication information into the tags; the scenario of the registration phase is presented in Fig. 1, and the corresponding steps are described as follows:

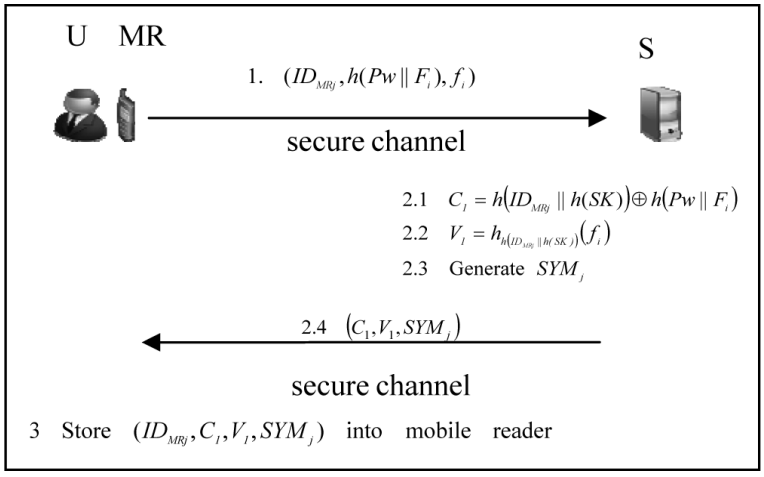

Fig. 1. Flowchart of registration phase.

Step 1: First, the unique Electronic Product CodeEPC_i, unique secret key K_i and the hash value of the market trademark $M = h(Tmark)$ are set to the ith tag and stored into server's database. The jth user uses mobile readers to compute f_j by:

$$f_j = h(F_j) \tag{1}$$

and then transmits the registration information $(ID_{MRj}, h(Pw||F_j), f_j)$ to server via a secured channel.

Step 2: Upon receiving the registration request from the user, the server computes C_1 and V_1 corresponding the user's registration information as follows:

$$C_1 = h(ID_{MRj}||h(SK)) \oplus h(Pw||F_j) \tag{2}$$

$$V_1 = h_{h(ID_{MRj}||h(SK))}(f_j) \tag{3}$$

and generates the symmetric key SYM_j. The server transmits the message (C_1, V_1, SYM_j) to the mobile reader via the secured channel.

Step 3: After receiving the messages (C_1, V_1, SYM_j), the user stores $(ID_{MRj}, C_1, V_1, SYM_j)$ into the mobile reader and the registration phase is finished.

3.4. Query and authentication phase

In this phase, we focus on mobile readers used to query product information, and verify the legality of the mobile user by the mutual authentication between server and tags. We illustrate the flowchart of this phase in Fig. 2, described in the following steps:

Step 1: Before the user wants to query the product information via the mobile reader, the user must be authenticated by the mobile reader. The user has to input the secret password Pw and fingerprint template F_j into the mobile reader; then the mobile reader computes:

$$h(ID_{MRj}||h(SK)) = C_1 \oplus (Pw||F_j) \tag{4}$$

and verifies V_1 as:

$$V_1? = h_{h(ID_{MRj}||h(SK))}(f_j) \tag{5}$$

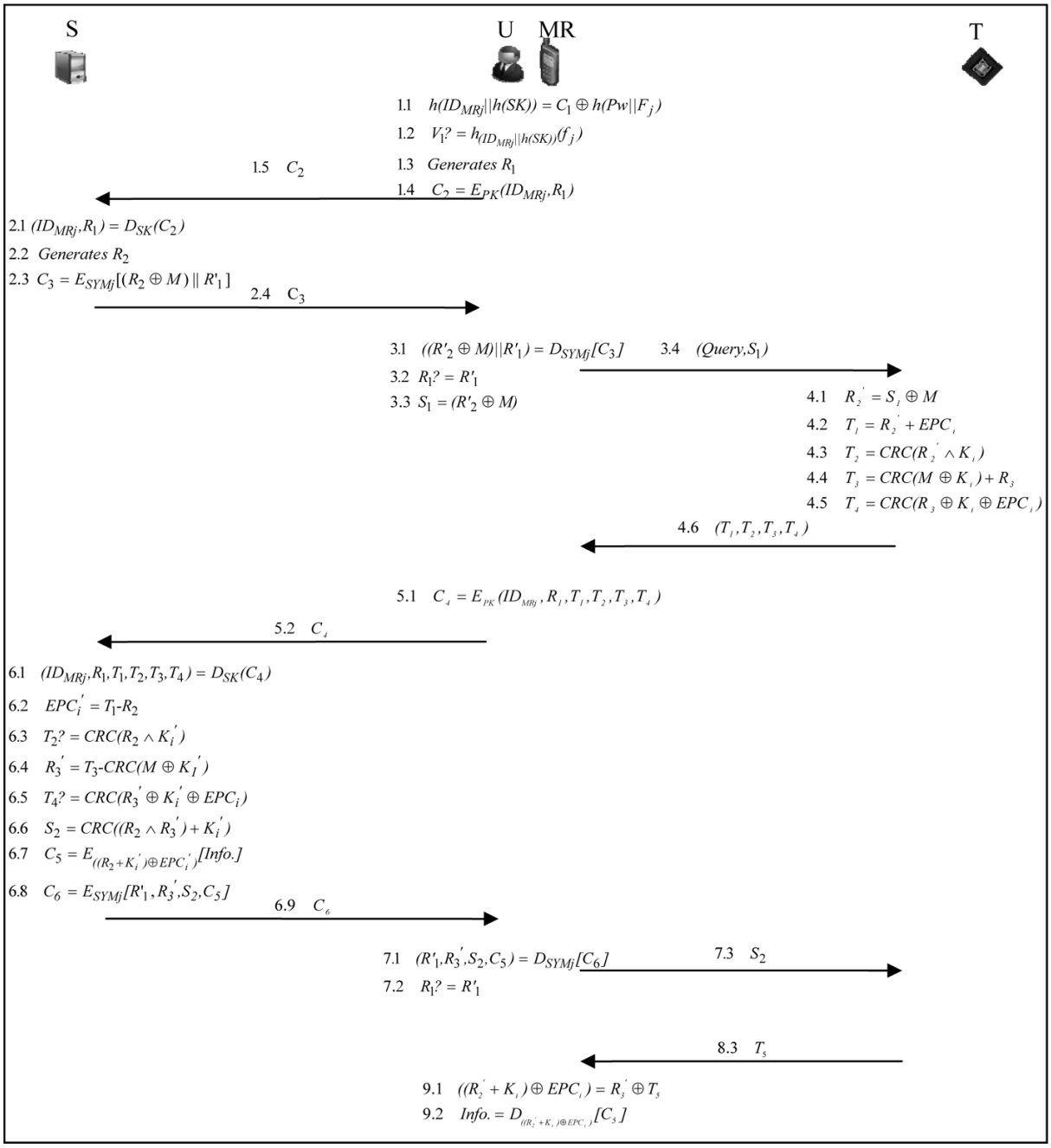

Fig. 2. Flowchart of query and authentication phase.

If the equality holds, the user is legal. Then the user can query the tag's information via the mobile reader; first, the mobile reader generates a random number R_1 and encrypts the mobile reader identity ID_{MRj} and R_1 by the server's public key PK as follows:

$$C_2 = E_{PK}(ID_{MRj}||R_1) \tag{6}$$

and then transmits C_2 to server.

Step 2: Upon receiving the message C_2, the server uses its secret key SK to decrypt the message C_2 by:

$$(ID_{MRj}, R_1') = D_{SK}(C_2) \tag{7}$$

and gets the mobile reader's identity ID_{MRj}. The server uses the mobile reader's identity ID_{MRj} to search the corresponding symmetric key SYM_j, and generates a random number R_2. After that, the server uses the mobile reader's symmetric key SYM_j to encrypt the value as follows:

$$C_3 = E_{SYMj}[((R_2 \oplus M)||R_1')] \tag{8}$$

and sends the message C_3 to the mobile reader.

Step 3: When the mobile reader receives the messages C_3 from the server, the user uses the mobile reader's symmetric key to decrypt the message C_3as follows:

$$((R_2' \oplus M)||R_1') = D_{SYMj}(C_3) \tag{9}$$

The user checks if $R_1? = R_1'$; and computes

$$S_1 = R_2' \oplus M \tag{10}$$

And then sends the query messages $(Query, S_1)$ to the specific tag.

Step 4: Once the messages $(Query, S_1)$ are received from the mobile reader, the tag uses the hash value $M = h(T - mark)$ to get the random number R_2' generated by the server, by the following calculation:

$$R_2' = S_1 \oplus M \tag{11}$$

The tag then uses its Electric Product Code EPC_i, secret key K_i and the random number R_2' to calculate T_1 and T_2 as follows:

$$T_1 = R_2' + EPC_i \tag{12}$$

$$T_2 = CRC(R_2' \wedge K_i) \tag{13}$$

Moreover, the tag generates a random number R_3 to compute T_3 and T_4 as follows:

$$T_3 = CRC(M \oplus K_i) + R_3 \tag{14}$$

$$T_4 = CRC(R_3 \oplus K_i \oplus EPC_i) \tag{15}$$

and sends the messages (T_1, T_2, T_3, T_4) to the mobile reader.

Step 5: After receiving the messages (T_1, T_2, T_3, T_4) from the tag, the user uses his/her mobile reader to encrypt the message (T_1, T_2, T_3, T_4), the mobile reader identity ID_{MRj} and the random number R_1 by the server's public key PK, as follows:

$$C_4 = E_{PK}(ID_{MRj},R_1,T_1,T_2,T_3,T_4) \tag{16}$$

The user then uses the mobile reader to send the message C_4 to the server.

Step 6: When receiving the message C_4, the server uses its secret key SK to decrypt the message C_4 as follows:

$$(ID_{MRj},R_1,T_1,T_2,T_3,T_4) = D_{SK}(C_4) \tag{17}$$

Then the server uses the message T_1 and the random number R_2 generated, to compute by:

$$EPC_i' = T_1 - R_2 \tag{18}$$

and gets the tag's Electronic Product Code $EPC_i{}'$. The server according to the EPC code EPC_i seeks the tag's corresponding secret key$K_i{}'$, and uses the $K_i{}'$ and R_2 to verify T_2 as follows:

$$T_2? = CRC(R_2 \wedge K_i') \tag{19}$$

If the equality holds, it means that the tag is legal. The server then uses the message T_3, the tag's EPC code EPC_i and M to compute $R_3{}'$ by:

$$R_3' = T_3 - CRC(M \oplus K_i{}') \tag{20}$$

The server uses the random number $R_3{}'$, the tag's secret key $K_i{}'$ and the tag's EPC code EPC_i to verify the message T_4 as follows:

$$T_4? = CRC(R_3' \oplus K_i' \oplus EPC_i) \tag{21}$$

If the equality holds, it means that the random number $R_3{}'$ generated by the tag has not been tampered with by the attacker in the delivering process.
Finally, the server uses the random numbers R_2, $R_3{}'$ and the tag's secret key $K_i{}'$ to compute S_2:

$$S_2 = CRC((R_2 \wedge R_3{}') + K_i{}') \tag{22}$$

and encrypts the tag's information $Info.$ by a session key $((R_2 + K_i{}') \oplus EPC_I)$ which is combined with the random number R_2 generated by the server, the tag's secret key K_i and EPC Code EPC_i as follows:

$$C_5 = E_{((R_2+K_i')\oplus EPC_I)}[Info.] \tag{23}$$

The server then uses the mobile reader's symmetric key SYM_j to encrypt the random number $R_3{}'$ generated by the tag and the values S_2, C_5:

$$C_6 = E_{SYMj}(R_1', R_3', S_2, C_5) \tag{24}$$

and sends the message C_6 to the mobile reader.

Step 7: Once the message C_6 is received from the server, the user uses the mobile reader's symmetric key SYM_j to decrypt C_6 as follows:

$$(R_1',R_3',S_2,C_5) = D_{SYMj}[C_6] \tag{25}$$

The user checks if the $R_1? = R_1'$; and uses his/her mobile reader to send the message S_2 to the tag.

Step 8: After receiving the message S_2, the tag uses the random number $R_2{}'$ generated by the server, the random number R_3 generated itself and secret key K_i to verify S_2, as follows:

$$S_2? = CRC((R_2{}' \wedge R_3) + K_i) \tag{26}$$

If the equality holds, it means that the server is authenticated by the tag successfully and it is legal. The tag then uses the values: $R_2{}'$, R_3, K_i and its EPC code EPC_i to calculate T_5 as follows:

$$T_5 = ((R_2{}' + K_i) \oplus EPC_i) \oplus R_3 \tag{27}$$

and sends T_5 to the mobile reader.

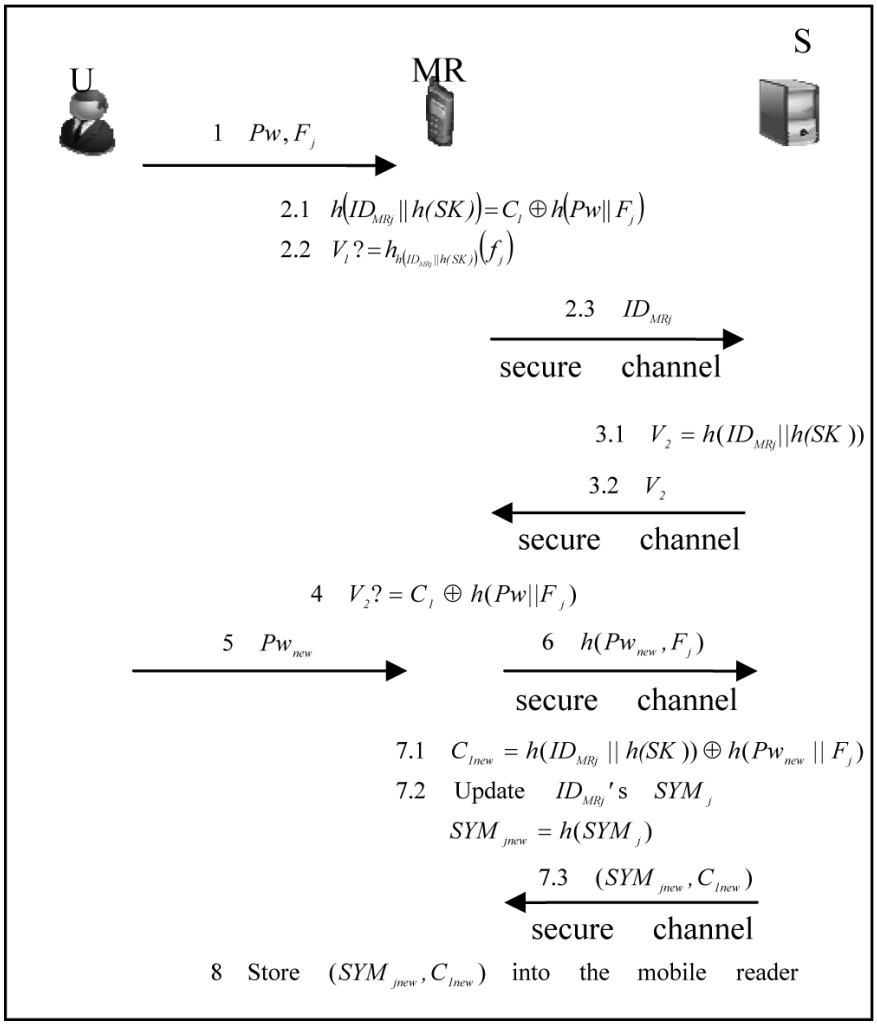

Fig. 3. Flowchart of password and key change phase.

Step 9: When receiving the message T_5, the user uses the mobile reader to compute the session key $((R_2' + K_i) \oplus EPC_i)$ with random number R_3' and T_5, as follows:

$$((R_2' + K_i) \oplus EPC_i) = R_3' \oplus T_5 \tag{28}$$

Finally, the user uses the session key to decrypt the message C_5:

$$Info. = D_{((R_2'+K_i)\oplus EPC_i)}[C_5] \tag{29}$$

and gets the tag's information.

3.5. Password and key change phase

In this phase, users can change their passwords by using mobile readers freely, and the server will update the symmetric key of the user's mobile reader. The flowchart of the password and key change phase is shown in Fig. 3; the descriptions are also provided as follows:

Step 1: The user inputs his/her original password Pw and fingerprint template F_i into the mobile reader.

Step 2: The mobile reader computes the hashed value as:

$$h(ID_{MRj}||h(SK)) = C_1 \oplus h(Pw||F_i) \tag{30}$$

and verifies the user's identify by:

$$V_1? = h_{h(ID_{MRj}||h(SK))}(f_j) \tag{31}$$

If the above equality holds, it means that the user is valid. The user then uses the mobile reader to send its identity ID_{MRj} to the server via the secure channel.

Step 3: When the server receives the mobile reader's identity ID_{MRj}, it computes V_2 as follows:

$$V_2 = h(ID_{MRj}||h(SK)) \tag{32}$$

and then sends V_2 to the mobile reader via secure channel.

Step 4: Once it receives the value V_2, the mobile reader verifies V_2 as follows:

$$V_2? = C_1 \oplus h(Pw||F_j) \tag{33}$$

If the equality of Eq. (32) holds, it means the server is legal.

Step 5: The user then inputs a new password Pw_{new} into the mobile reader.

Step 6: The mobile reader then sends the value $h(Pw_{new}||F_j)$ to the server via a secure channel.

Step 7: Upon receiving the value $h(Pw_{new}||F_j)$ from the mobile reader, the server computes C_{1new} as follows:

$$C_{1new} = h(ID_{MRj}||h(SK)) \oplus h(Pw_{new}||F_j) \tag{34}$$

and updates the mobile reader's symmetric key SYM_j by:

$$SYM_{jnew} = h(SYM_j) \tag{35}$$

The server then sends the messages (SYM_j, C_{1new}) to the mobile reader via a secure channel.

Step 8: After receiving the messages (SYM_j, C_{1new}), the user stores (SYM_j, C_{1new}) into the mobile reader.

4. Security analysis

4.1. Resist tag impersonation attack

In step 6 of the query and authentication phase, the server verifies whether a tag is legal by:

$$T_2? = CRC(R_2 \wedge K_i{}') \tag{36}$$

If an attacker wants to impersonate a tag and passes the server's authentication successfully, he/she must know the server's challenge random number R_2 and the tag's secret key K_i. In our scheme, the server's challenge random number R_2 and tag's secret key K_i are not transmitted directly in plain text; thus, the attacker can not successfully impersonate a tag and pass the server's authentication procedure.

4.2. Resist mobile reader lost attack

If a malicious user steals the mobile reader and uses the mobile reader to query a tag, the malicious user must know the true password Pw and true fingerprint template F_j. But the malicious users can not use the illegal password Pw' and fingerprint template F_j' to compute:

$$h(ID_{MRj}||h(SK))' = C_1 \oplus h(Pw'||F_j{}') \tag{37}$$

So, in the step 1 of the query and authentication phase, the malicious users can not pass the authentication successfully as:

$$V_1 \neq h_{h(ID_{MRj}||h(SK))'}(f_j{}') \tag{38}$$

Thus, our scheme can resist malicious users from using a lost mobile reader to violate the owner's rights.

4.3. Resist replay attack

In step 4 of the query and authentication phase, each tag uses some privacy information: unique secret key K_i and the value M to generate messages T_1 and T_4. If the attacker intercepts the tag's privacy information K_i and the hash value M, he/she can transmit previously obtained messages T_1 and T_4 to pass the mobile reader's authentication procedure. The scenario is described as follows:

The attacker intercepts the 1st communication messages.

Step 4: T $\rightarrow$ MR: T_1, T_2, T_3, T_4

where

$$T_1 = R_2{}' + EPC_i \tag{39}$$

$$T_4 = CRC(R_3 \oplus K_i \oplus EPC_i) \tag{40}$$

The attacker replays previously obtained T_1 and T_4 to pass the mobile reader's authentication procedure, but it will fail. The reason is described as follows:

$$T_1 = R_2{}' + EPC_i \tag{39}$$

$$T_1{}' = R_2{}'' + EPC_i \tag{41}$$

$$T_1 \neq T_1' \tag{42}$$

and

$$T_4 = CRC(R_3 \oplus K_i \oplus EPC_i) \tag{40}$$

$$T_4{}' = CRC(R_3{}' \oplus K_i \oplus EPC_i) \tag{43}$$

$$T_4 \neq T_4' \tag{44}$$

Since the random number R_3 is updated for each transaction, the attacker can successfully not pass the mobile reader's authentication procedure. Thus, our scheme can resist replay attack.

4.4. Resist trace attack

In step 4 of the query and authentication phase, the tag uses the random numbers R_2' and R_3 to make the values (T_1, T_2, T_3, T_4) differ for each transaction. Even though the attacker intercepts the values: (T_1, T_2, T_3, T_4), he/she can not use the values: (T_1, T_2, T_3, T_4) to trace the tag's location. Thus our scheme can resist trace attack.

4.5. Resist forgery attack

The Cyclic Redundancy Check (CRC) is a checksum algorithm which is used to detect data errors during transmission. The CRC checksum is computed as a remainder of the division of the original data by the CRC polynomial. The following theorem is the basis of the CRC linear property [23].

Theorem 1: For any CRC (independent of its generator polynomial), for any values a and $b \in F_2[X]$, it holds:

$$CRC(a) \oplus CRC(b) = CRC(a \oplus b) \tag{45}$$

In step 4 of the query and authentication phase, an attacker can intercept messages T_3 ($T_3 = CRC(M \oplus K_i) + R_3$) and T_4 when the tag sends messages to the reader, and then the attacker can decompose T_4 as follows:

$$\begin{aligned} T_4 &= CRC(R_3 \oplus K_i \oplus EPC_i) \\ &= CRC(R_3 \oplus K_i) \oplus CRC(EPC_i) \\ &= CRC(R_3) \oplus CRC(K_i) \oplus CRC(EPC_i) \end{aligned} \tag{46}$$

Since the random number R_3 is updated for each transaction, and the random number R_3 is protected by mathematics or logic computation in messages T_3 and T_4. The attacker can not decompose the message T_4 to get the i th tag's unique secret key K_i, the EPC EPC_i of the i th tag; thus, our scheme can resist forgery server attack.

4.6. Resist man-in-the-middle attack

When the mobile reader wants to send a request message to the tag, the attacker will intercept the messages from the mobile reader and then transfer the messages to the tag as follows.

Step 1: MR $\rightarrow$ S: C_2
Step 2: S $\rightarrow$ MR: C_3
Step 3: MR $\rightarrow$ T: $Query, S_1$
Step 4: T $\rightarrow$ MR: T_1, T_2, T_3, T_4
Step 5: MR $\rightarrow$ S: C_4
Step 6: S $\rightarrow$ MR: C_6
Step 7: MR $\rightarrow$ T: S_2
Step 8: T $\rightarrow$ MR: T_5

The server calculates S_1 and S_2 as follows:

$$S_1 = R_2 \oplus M \tag{47}$$

$$S_2 = CRC((R_2 \wedge R_3) + K_i) \tag{48}$$

The attacker can intercept and modify the message S_1 and S_2; the message S_1' is computed as follows:

$$S_1' = R_2' \oplus M \tag{49}$$

And the message S_2' is computed as follows:

$$S_2' = CRC((R_2' \wedge R_3) + K_i') \tag{50}$$

Due to the correct random numbers R_3 being protected by related parameters and updated for each transaction, the attacker can not compute the next correct communication values S_1 and S_2 by spoof messages. Therefore, our proposed scheme can resist man-in-the-middle attack.

4.7. Privacy protection

In the query and authentication phase, an attacker can intercept messages C_2 when the reader sends a message to the server. The reader's identity ID_{MR} is protected by random number R_1, and the server's public key *PK,* as follows:

$$C_2 = E_{PK}(ID_{MRj}||R_1) \tag{6}$$

On the other hand, the tag's EPC_i is protected by the tag's secret key K_i and the random number R_2; the hash value of the trademark $M$is protected by the tag's secret key K_i and the random number R_3, as follows:

$$T_1 = R_2' + EPC_i \tag{12}$$

$$T_2 = CRC(R_2' \wedge K_i) \tag{13}$$

$$T_3 = CRC(M \oplus K_i) + R_3 \tag{14}$$

$$T_4 = CRC(R_3 \oplus K_i \oplus EPC_i) \tag{15}$$

So, the attacker can not decompose the messages to ascertain the reader's real identity ID_{MR}, the identity EPC_i of Tag A or Tag B.

4.8. Forward secrecy

Forward secrecy refers to past messages not being compromised even after the long-term secret data are exposed. On the other hand, even if somebody decomposes the transaction message between the reader and the tag, he/she can not analyze the message to decipher the content via pre-transaction. In our scheme, the user's password P_W and the *j th* mobile reader's symmetric key SYM_j are updated as follows:

$$C_{1new} = h(ID_{MRj}||h(SK)) \oplus h(Pw_{new}||F_j) \tag{34}$$

$$SYM_{jnew} = h(SYM_j) \tag{35}$$

The server then sends the messages (SYM_j, C_{1new}) to the mobile reader via secure channel. Thus, our protocol can achieve forward secrecy between the server and the reader.

5. Mechanism analysis

5.1. Mutual authentication

Case 1: Server authenticates tag: In step 4 of the query and authentication phase, the tag uses the value M shared with the server to compute the random number $R_2{}'$ generated by the server, as follows:

$$R_2{}' = S_1 \oplus M \tag{11}$$

The tag then computes:

$$T_2 = CRC(R_2{}' \wedge K_i) \tag{13}$$

and responds T_2 to the server. The server then authenticates whether or not the tag is legal by:

$$T_2? = CRC(R_2 \wedge K_i{}') \tag{19}$$

Case 2: Tag authenticates server: After authenticating the tag, the server then uses the random numbers R_2,$R_3{}'$ and the tag's unique secret key $K_i{}'$ to compute S_2, as follows:

$$S_2 = CRC((R_2 \wedge R_3') + K_i{}') \tag{22}$$

The tag will use the random numbers $R_2{}'$, R_3 and unique secret key K_i to verify whether or not the server is legal by:

$$S_2? = CRC((R_2{}' \wedge R_3) + K_i) \tag{26}$$

Therefore, our proposed scheme achieves mutual authentication.

5.2. Anonymity

In our protocol, ID_{MR} is transmitted via a secure channel. The reader's identity ID_{MR}, user's fingerprint template F_j and tag's EPC are not transmitted in plain text. We camouflage the identity information with the random numbers R_1, R_2, R_3 and compute by the secret values K_i, the jth user's mobile reader's symmetric key SYM_j and the server's secret key SK between the sender and receiver. So, if the attacker interrupts this information to analyze the identity of a sender or receiver, he/she only obtains the protected identity of members in the current transaction; the attackers fail to obtain the real identity information of the members.

5.3. Conform EPCglobal C1G2

In our protocol, tags only use comparison operations, exclusive-or operations, addition operations, CRC operations, and subtraction operations. These operations conform to the EPCglobal C1G2 standards and are low-cost, so they decrease the computational load of tags.

6. Conclusion

In this paper, we present a RFID mutual authentication protocol for market application system. Our scheme has some positive characteristics. For example: unregistered mobile reader can not access the service; the mobile reader integrates the user's fingerprint biometrics and all tags conform to EPCglobal

class 1 generation 2 standards. In order to enhance the user's privacy, we combine and integrate the authentication fingerprint biometrics and password mechanism into the mobile reader. Even though the user lost his/her mobile reader, the person who gets the mobile reader can not pass the mobile reader's authentication procedure.

Furthermore, our scheme is secure against tag impersonation attack, replay attack, trace attack and forgery attack; it maintains privacy protection and achieves mutual authentication, anonymity and forward secrecy. In summary, our scheme can provide a convenient, low-cost and improved security mechanism in market management system.

Acknowledgment

This work is partially supported by the National Science Council, Taiwan, under contract No. NSC 99-2628-E-324-026.

References

[1] V. Conti, C. Militello, F. Sorbello and S. Vitabile, A multimodal technique for an embedded fingerprint recognizer in mobile payment systems, *Mobile Information Systems* **5**(2) (May 2009), 105–124.
[2] P. Fülöp, S.R. Imre, S. Szabó and T. Szálka, Accurate mobility modeling and location prediction based on pattern analysis of handover series in mobile networks, *Mobile Information Systems* **5**(3) (Oct 2009), 255–289.
[3] S. Caballé, F. Xhafa and L. Barolli, Using mobile devices to support online collaborative learning, *Mobile Information Systems* **6**(1) (April 2010), 27–47.
[4] H.H. Hsu and C.C. Chen, RFID-based human behavior modeling and anomaly detection for elderly care, *Mobile Information Systems* **6**(4) (Dec 2010), 341–354.
[5] S.L. Garfinkel, A. Juels and R. Pappu, RFID Privacy: An overview of problems and proposed solutions, *IEEE Security & Privacy Magazine* **3** (May 2005), 34–43.
[6] R. Weinstein, RFID: a technical overview and its application to the enterprise, *IT Professional* **7**(3) (2005), 27–33.
[7] J. Ayoade, Security implications in RFID and authentication processing framework, *Computers & Security* **25**(3) (2006), 207–212.
[8] B. Toiruul and K. Lee, An advanced mutual-authentication algorithm using AES for RFID systems, *in International Journal of Computer Science and Network Security* (*IJCSNS*) **6** (2006), 156–162.
[9] K. Osaka, T. Takagi, K. Yamazaki and O. Takahashi, An efficient and secure RFID security method with ownership transfer, *IEEE International Conference on Computational Intelligence and Security, Ramada Pearl Hotel*, Guangzhou, China, (Nov. 2006), 3–6.
[10] S. Karthikeyan and M. Nesterenko, RFID security without extensive cryptography, *Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks, Hilton Alexandria Mark Center, Alexandria*, VA, USA, (Nov. 2005), 63–67.
[11] H. Gilbert, M. Robshaw and H. Sibert, An active attack against HB+ – a provably secure lightweight protocol, *IEEE Electronic Letters* **41**(21) (2005), 1169–1170.
[12] EPCglobal, Inc., http://www.epcglobalinc.org/, Access available: 16 May 2010.
[13] C.L. Chen and Y.Y. Den, Conformation of EPC Class 1 Generation 2 Standards RFID System with Mutual Authentication and Privacy Protection, *Engineering Applications of Artificial Intelligence* **22**(8) (2009), 1284–1291.
[14] H.Y. Chien and C.H. Chen, Mutual authentication protocol for RFID conforming to EPC class 1 generation 2 standards, *Computer Standards and Interfaces* **22**(8) (2007), 254–259.
[15] Wireless Dynamics Inc., http://www.wdi.ca/products.shtml. Access available: 16 May 2010.
[16] G. Costa, A. Lazouski, F. Martinelli, I. Matteucci, V. Issarny, R. Saadi, N. Dragoni and F. Massacci, Security-by-contract-with-trust for mobile devices, *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications* **1**(4) (2010), 75–91.
[17] J. Hunker and C.W. Probst, Insiders and Insider Threats – An Overview of Definitions and Mitigation Techniques, *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications* **2**(1) (2011), 4–27.
[18] J. Kim and H. Kim, A wireless service for product authentication in mobile RFID environment, *1st International Symposium on Wireless Pervasive Computing*, Phuket, Thailand, (Jan 2006), 16–18.
[19] N. Park, J. Kwak, S. Kim, D. Won and H. Kim, WIPI mobile platform with secure service for mobile RFID network environment, *APWeb Workshops*, Harbin, China, (Jan 2006), 741–748.

[20] W. Zhu, D. Wang and H. Sheng, Mobile RFID technology for improving m-commerce, *IEEE International Conference on e-Business Engineering* (*ICEBE 2005*), Beijing, China, (Oct 2005), 118–125.
[21] NIST FIPS PUB 180-2, Secure Hash Standard, *National Institute of Standard and Technology, U.S Department of Commerce*, DRAFT, (2002).
[22] A. Bhargav-Spantzel, A.C. Squicciarini, S. Modi, M. Young, E. Bertino and S. J. Elliott, Privacy preserving multi-factor authentication with biometrics, *Journal of Computer Security* **15**(5) (2007), 529–560.
[23] P.L. Pedro, H.C. Julio Cesar, M.E.T. Juan and R. Arturo, Cryptanalysis of a novel authentication protocol conforming to EPC-C1G2 standard, *Computer Standards & Interfaces* **31**(2) (2009), 372–380.

Chin-Ling Chen was born in Taiwan in 1961. He received a B.S. in Computer Science and Engineering from Feng Cha University in 1991; he holds both an M.S. and a Ph.D. in Applied Mathematics from the National Chung Hsing University, Taichung, Taiwan, in 1999 and 2005, respectively. He is an active member of the Chinese Association for Information Security. From 1979 to 2005, he was a senior engineer at the Chunghwa Telecom Co., Ltd. He is currently an associate professor within the Department of Computer Science and Information Engineering at the Chaoyang University of Technology. From 2011, he is an Editorial Board member of International Journal of Advances in Internet of Things. His research interests include: cryptography, network security, and electronic commerce.