Efficient Ultralightweight RFID Mutual Authentication

Yu-Chung Huang and Jehn-Ruey Jiang

Department of Computer Science and Information Engineering
National Central University
Jhongli City, Taiwan

***Abstract*—In the RFID (Radio Frequency Identification) system, the communication between the reader and tags is vulnerable to attacks due to the nature of RF signals. Typical attacks include the forged-server, forged-tag, man-in-the-middle (MitM), tracking, replay, forward secrecy and denial of service (DoS) attacks. Some mutual authentication schemes/protocols have been proposed to resist these attacks. Unfortunately, these schemes still have some flaws. For example, some of them cannot resist all the above-mentioned attacks due to the cyclic redundancy check (CRC) security flaw, and others need tags to have more powerful computation ability than a normal passive one. In this paper, we propose a mutual authentication protocol conforming to the popular EPC Class 1 Generation 2 (EPC C1G2) specification to resist all the above-mentioned attacks. The proposed protocol uses only ultrallightweight operations, including CRC, to reduce computation and communication overheads without causing the CRC security flaw. We conduct security analysis for the proposed scheme and compare it with other related ones to demonstrate its superiority in terms of the communication cost, computation cost and security.**

Keywords-Radio Frequency Identification (RFID); Electronic Product Code (EPC); Security; Mutual Authentication; Cyclic Redundancy Check (CRC)

I. INTRODUCTION

In recent years, RFID (Radio Frequency Identification) systems have been widely adopted by many applications, such as manufacture automation, animal tracking, healthcare, etc. An RFID system consists of tags, readers and a backend server [1-4]. A tag with a unique ID is attached to an object and a reader can recognize the object by identifying the attached tag with the identification procedure (or interrogation procedure). With this identified tag ID, the reader can then retrieve the related information of the object from the backend server database.

In the identification procedure, the reader issues RF signals to command tags to respond their IDs. Due to the nature of RF signals, the communication between the reader and tags is vulnerable to attacks. The adversary can intercept (or eavesdrop on) messages, modify them, and/or inject fake messages to launch attacks, such as the forged-server, forged-tag, man-in-the-middle (MitM), tracking, replay, forward secrecy and denial of service (DoS) attacks [5-9].

Several schemes [11, 12] has been proposed, which use only ultralightweight operations, such as the random number generator (RNG), pseudo random number generator (PRNG), cyclic redundancy check (CRC), exclusive-or (XOR) operations, to resist the latent attacks. Since those schemes apply only ultralightweight operations, they can be executed on resource-limited tags conforming to the popular EPCglobal Class 1 Generation 2 (EPC C1G2) standard [10]. Unfortunately, these schemes still suffer from security weaknesses.

Chien and Chen [11] proposed a scheme based on the XOR and CRC operations to achieve mutual authentication between the reader and tags. However, Peris-Lopez et al. [14] pointed out various major security weaknesses due to the CRC security flaw in Chien and Chen's scheme, such as vulnerability to the forged-server and forged-tag, MitM and DoS attacks. Chen and Deng [12] proposed a new EPC C1G2 compliant mutual authentication scheme using the PRNG and CRC operations to resist several attacks. Nevertheless, Peris-Lopez et al. [14] revealed the weaknesses in Chen and Deng's scheme, such as vulnerability to the forged-server, forged-tag, tracking, and DoS attacks. Huang and Jiang [13] further proposed an ultralightweight mutual authentication scheme, which can resist several attacks by using the RNG, PRNG and XOR operations. Without using the CRC operation, the scheme thus avoids the CRC security flaw [14] and can still resist the above-mentioned attacks.

In this paper, we propose an EPC C1G2 compliant mutual authentication scheme using the ultralightweight the XOR operation, the RPNG operation, as well as the CRC operation to reduce computation and communication overheads. Although using the CRC operation, the proposed scheme avoids the CRC security flaw carefully and can resist the forged-server, forged-tag, MitM, tracking, replay, forward secrecy and DoS attacks. We compare the proposed scheme with other related ones to demonstrate its superiority in terms of the communication cost, computation cost and security.

The remainder of this paper is organized as follows. Some mutual authentication schemes that conform to EPC C1G2 standard are introduced in Section II. The proposed scheme is detailed in Section III. Security analyses and

comparisons are presented in Section IV. Finally, some concluding remarks are drawn in Section V.

II. Related Work

The EPC C1G2 scheme [15] was adopted by ISO/IEC as an international standard referred to as ISO/IEC 18000-6C. An EPC C1G2 tag is passive and communicates with a reader on the UHF band (800-960 MHz) at the range from 2 m to 10 m depending on the operating environment. It supports on-chip 16-bit PRNG, 16-bit CRC, and XOR operations

Several schemes [11-14] try to raise the security level of EPC C1G2 RFID systems in which tags are resource-limited. These schemes thus use only ultralightweight operations, such as PRNG, CRC, and XOR, in order to build systems conforming to the EPC C1G2 standard. Below, we describe in detail three of these schemes, Chien and Chen's scheme [11], Chen and Deng's scheme [12] and Huang and Jiang's scheme [13], which are most related to our proposed scheme.

A. *Chien and Chen's scheme* [11]

Chien and Chen's scheme uses only ultralightweight PRNG and CRC operations. Initially, the backend server randomly selects an initial authentication key K_i^0 and an initial access key P_i^0 for tag_i. The two keys are stored on tag_i and will be updated after each successful authentication session. The Electronic Product Code (EPC) EPC_i is also stored on tag_i.

The server database maintains a six-field record (EPC_i , K_i^{old} , P_i^{old} , K_i^{new} , P_i^{new} , $DATA$) for tag_i. In the record, K_i^{old} (K_i^{new}) is the old (new) authentication key for tag_i and it is set to K_i^0 initially; meanwhile, P_i^{old} (P_i^{new}) denotes the old (new) access key and is set to P_i^0 initially; the last one, $DATA$, denotes the full information about the tagged object.

The authentication process is explained as follows.

Step 1: $Reader_j$ sends a random number N_1 as a challenge to tag_i.

Step 2: Tag_i generates a random number N_2 and then calculates $CRC(EPC_i \,||\, N_1 \,||\, N_2) \oplus K_i$ as M_1 . The value M_1 is sent back to $reader_j$ and forwarded to the server by the reader.

Step 3: The server retrieves every data record in the database and checks if I^{old} or I^{new} matches M_1, where $I^{old} = M_1 \oplus K_i^{old}$ and $I^{new} = M_1 \oplus K_i^{new}$. The check is repeated until a match is found or the end of the database is reached. If a match is found, it implies that tag_i has been successful authenticated; otherwise, a failure message is sent to $reader_j$ and the authentication process stops.

The server calculates $M_2 = CRC(EPC_i \,||\, N_2) \oplus P_i^{old}$ or $M_2 = CRC(EPC_i \,||\, N_2) \oplus P_i^{new}$ depending on either of K_i^{old} and K_i^{new} is matched. It also updates authentication key K_i and access key P_i by executing $K_i = PRNG(K_i^{new})$ and $P_i = PRNG(P_i^{new})$.

Step 4: The server sends (M_2, $DATA$) to $reader_j$, where $DATA$ is the information of the object to which tag_i is attached. Then $reader_j$ in turn passes M_2 to tag_i.

Step 5: Upon receiving M_2, tag_i has to verify whether the $M_2 \oplus P_i$ equals $CRC(EPC_i \,||\, N_2)$. If so, it updates its K_i and P_i by executing $K_i = PRNG(K_i^{new})$ and $P_i = PRNG(P_i^{new})$.

In Chien and Chen's scheme, Tag_i shares some private information, such as EPC_i, authentication key K_i and access key P_i with $reader_j$. This information is used to build messages M_1 and M_2 in order to prove its authenticity. Unfortunately, since the communication channel between tag_i and $reader_j$ is insecure, the adversary can monitor and modify the message between them. As shown by Peris-Lopez et al. in [14], Chien and Chen's scheme cannot resist forged-tag, forged-server, DoS, tracking, and forward secrecy attacks.

B. *Chen and Deng's scheme* [12]

Chen and Deng [12] proposed a mutual authentication scheme between tag_i and $reader_j$ based on the use of the PRNG and CRC operations. Originally, tag_i is associated with a unique EPC code EPC_i, and $reader_j$ is associated with a unique identification IDR_j. To register tag_i, the server randomly selects a nonce N_i and an initial authentication key K_i for tag_i and stores EPC_i, N_i and K_i in both tag_i and the server database. To register $reader_j$, the server stores IDR_j in the database. After registration, the reader and the tag can authenticate each other by the following steps.

Step 1: When $reader_j$ wants to access tag_i, it sends a request message (M_{req} , RND_1 , $CRC(N_i \oplus RND_1)$) to tag_i, where RND_1 is a random number generated by RNG.

Step 2: Upon receiving(M_{req} , RND_1 , $CRC(N_i \oplus RND_1)$), tag_i uses the stored N_i to calculate CRC $(N_i \oplus RND_1)$ and check the validation of CRC $(N_i \oplus RND_1)$. If a match is found, it implies that tag_i has been successful authenticated, then tag_i generate a new random number RND_2 and calculate X as $K_i \oplus EPC_i \oplus RND_2$ and Y as $CRC(RND_2 \oplus N_i \oplus X)$ and responds the message (RND_2, X, Y) to $reader_j$.

Step 3: Upon receiving the response message from tag_i, then $reader_j$ computes its local comparison version of Y= CRC $(RND_2 \oplus N_i \oplus X)$. If the local comparison version Y equals the received Y , the server uses K_i to obtain EPC_i of tag_i as $EPC_i = K_i \oplus RND_2 \oplus X$.

Step 4: When $reader_j$ obtains EPC_i from step 3 and confirms the authenticity of tag_i, the $reader_j$ sends a response M_{resp} to tag_i.

Chen and Deng's scheme is vulnerable to the forged attack due to CRC security flaw indicated by Peris-Lopez et al. [14]. One possible forged attack is described below. In Chen and Deng's scheme, RND_2 are updated according to X and Y containing in the message sent by the valid tag_i. Unfortunately, the adversary can eavesdrop on the message and obtain the values of X and Y. The adversary can substitute original X and Y to pass the authentication. Additionally, the scheme cannot resist some attacks either, such as the forged-server and forged-tag, DoS, replay attack and tracing attacks.

C. *Huang and Jiang's scheme* [13]

Huang and Jiang's scheme [9] adopted the RNG, PRNG and XOR operations to achieve mutual authentication between tag_i and $reader_j$. These operations are ultralightweight and can be implemented on EPC C1G2 RFID tags.

Initially, the server sends (EPC_i, N_i, K_i, PID_i) to tag_i and stores (EPC_i, N_i^{old}, K_i^{old}, PID_i^{old}, N_i^{new}, K_i^{new}, PID_i^{new}) in the database to register tag_i, where EPC_i is the EPC code, N_i is the communication key, K_i is the authentication key, and PID_i is the pseudonym identity of tag_i. Note that the server stores two versions of N_i, K_i and PID_i, that is the current version N_i^{new}, K_i^{new} and PID_i^{new}, and the old version N_i^{old}, K_i^{old} and PID_i^{old}. At the beginning, $N_i^{old} = N_i^{new}$, $K_i^{old} = K_i^{new}$, and $PID_i^{old} = PID_i^{new}$. Note that the server sends RID_j to $reader_j$ and stores RID_j in the database to register $reader_j$, where RID_j is called the pseudonym identity of $reader_j$.

Huang and Jiang's scheme is described as follows:

Step 1: Before $reader_j$ queries tag_i, it generates a random number r_1 and sets $V_R = h(RID_j \oplus r_1)$, where h is a hash function. Then $reader_j$ sends a request message (r_1) to tag_i.

Step 2: Upon receiving (r_1), tag_i generates a random number r_2 and uses N_i, K_i and EPC_i to calculate $M_1 = N_i \oplus r_2$ and $M_2 = P_{RNG}(EPC_i || r_1 || r_2) \oplus K_i$. After that, it responds to $reader_j$ with (M_1, M_2, PID_i).

Step 3: After receiving the response message from tag_i, $reader_j$ appends r_1 and V_R to this message to form an authentication request (M_1, M_2, PID_i, r_1, V_R) to send to the server.

Step 4: Upon receiving the authentication request (M_1, M_2, PID_i, r_1, V_R) from $reader_j$, the server authenticates $reader_j$ by verifying V_R=$H(RID_j \oplus R_1)$, where h is a general hash function. If the verification is successful, the server uses PID_i to find (N_i^{old}, N_i^{new}, K_i^{old}, K_i^{new}, EPC_i) in the backend database by checking $PID_i \stackrel{?}{=} PID_i^{old}$ or $PID_i \stackrel{?}{=} PID_i^{new}$. The server then calculates $r_2 = M_1 \oplus N_i^{old}$ and $r_2 = M_1 \oplus N_i^{new}$ to verify the correctness of $M_2 \stackrel{?}{=} P(EPC_i || r_1 || r_2) \oplus K_i^{old}$ and $M_2 \stackrel{?}{=} P(EPC_i || r_1 || r_2) \oplus K_i^{new}$, where P is the *PRNG* operation (or function). If either of the above verifications is correct, the server sets x= *old* (if K_i^{old} passes the verification) or x= *new* (if K_i^{new} passes the verification), and calculates $M_3 = P(EPC_i || r_2 || N_i^x) \oplus K_i^x$, $Info = D_i \oplus RID_j$ and forwards the message (M_3, *Info*) to $reader_j$. Moreover, if $x = new$, the server performs the following updates: $PID_i^{old} = PID_i^{new}$, $PID_i^{new} = P(PID_i \oplus r_2)$, $N_i^{old} = N_i^{new}$, $N_i^{new} = P(N_i \oplus r_2)$, $K_i^{old} = K_i^{new}$, and $K_i^{new} = P(K_i \oplus r_2)$.

Step 5: After receiving the message (M_3), $reader_j$ forwards M_3 to tag_i.

Step 6: Upon receiving M_3 from $reader_j$, tag_i verifies the correctness of $M_3 \stackrel{?}{=} P(EPC_i || r_2 || N_i) \oplus K_i$. If the above verification is correct, tag_i performs the following updates: $PID_i = P(PID_i \oplus r_2)$, $N_i = P(N_i \oplus r_2)$, and $K_i = P(K_i \oplus r_2)$.

In Huang and Jiang's scheme, the pseudonym identification PID_i is updated according to M_3 contained in the message sent by the authenticated $reader_j$. Unfortunately, an adversary may intercept the message to prevent tag_i from updating PID_i. The adversary can track tag_i if tag_i responds to $reader_j$ interrogation queries with the same PID_i continuously. Hence, the scheme is vulnerable to the tracking attack.

III. PROPOSED SCHEME

This section elaborates the proposed mutual authentication scheme, which has two phases: (1) the register phase and (2) the mutual authentication phase. Similar to the schemes mentioned in Section II, the proposed scheme assumes that an adversary is able to monitor and modify the communication messages between tag_i and $reader_j$, but the communication between $reader_j$ and the backend server is secure. Notations used in the proposed scheme are described in Table 1.

Table 1. Notations used in the proposed scheme

P_i	The communication key shared between the tag_i and $reader_j$
K_i	The authentication key shared between the tag_i and $reader_j$
$\oplus$	The exclusive-or operation
r	A random number generator by $reader_j$ or tag_i
$\|\|$	The concatenation operation
$P(\cdot)$	A 16-bit pseudo random number generator
$CRC(\cdot)$	A 16-bit Cyclic Redundancy Check function
EPC_i	The 96-bit EPC (Electronic Product Code) of tag_i
$A \stackrel{?}{=} B$	A comparison function that checks whether A is equal to B

A. Registration phase

Initially, tag_i keeps EPC_i, P_i and K_i, where EPC_i is the Electronic Product Code, P_i is the communication key and K_i is the authentication key. Furthermore, the server stores (EPC_i, P_i^{old}, P_i^{new}, K_i^{old}, K_i^{new}) in the server database to register tag_i, where P_i^{new} and K_i^{new} are the current (or new) version of P_i and K_i, and P_i^{old} and K_i^{old} are the old version of P_i and K_i. Note that at the beginning, $P_i^{old} = P_i^{new}$ and $K_i^{old} = K_i^{new}$.

B. Mutual authentication phase

The procedures of the mutual authentication phase of the proposed scheme is depicted in Fig. 1 and described as follows.

Step 1: Before $reader_j$ begins to query tag_i, it generates a random number r_1 and then sends a message (r_1) as a challenge to tag_i.

Step 2: Upon receiving (r_1), tag_i generates a random number r_2 and uses P_i, K_i and EPC_i to calculate A, B and M_1 according to Eqs. (1)-(3).

$$A = r_1 \oplus r_2 \oplus P_i \tag{1}$$

$$B = (P_i || r_1) \oplus r_2 \tag{2}$$

$$M_1 = CRC(EPC_i || A || B || K_i) \tag{3}$$

Afterwards, tag_i sends (M_1, B) to $reader_j$.

Step 3: After receiving (M_1, B), $reader_j$ appends r_1 to this message as an authentication request and forwards (M_1, B, r_1) to the backend server.

Step 4: Upon receiving the authentication request (M_1, B, r_1) from $reader_j$, the server searches all of (EPC_i, P_i^{old}, P_i^{new}, K_i^{old}, K_i^{new}) in its database to calculate r_2 and A' according to Eqs. (4)-(7).

$$r_2 = B \oplus (P_i^{old} || r_1) \tag{4}$$

$$r_2 = B \oplus (P_i^{new} || r_1) \tag{5}$$

$$A' = r_1 \oplus r_2 \oplus P_i^{old} \tag{6}$$

$$A' = r_1 \oplus r_2 \oplus P_i^{new} \tag{7}$$

The server then executes the following verifications.

$$M_1 \stackrel{?}{=} CRC(EPC_i || A' || B || K_i^{old}) \tag{8}$$

$$M_1 \stackrel{?}{=} CRC(EPC_i || A' || B || K_i^{new}) \tag{9}$$

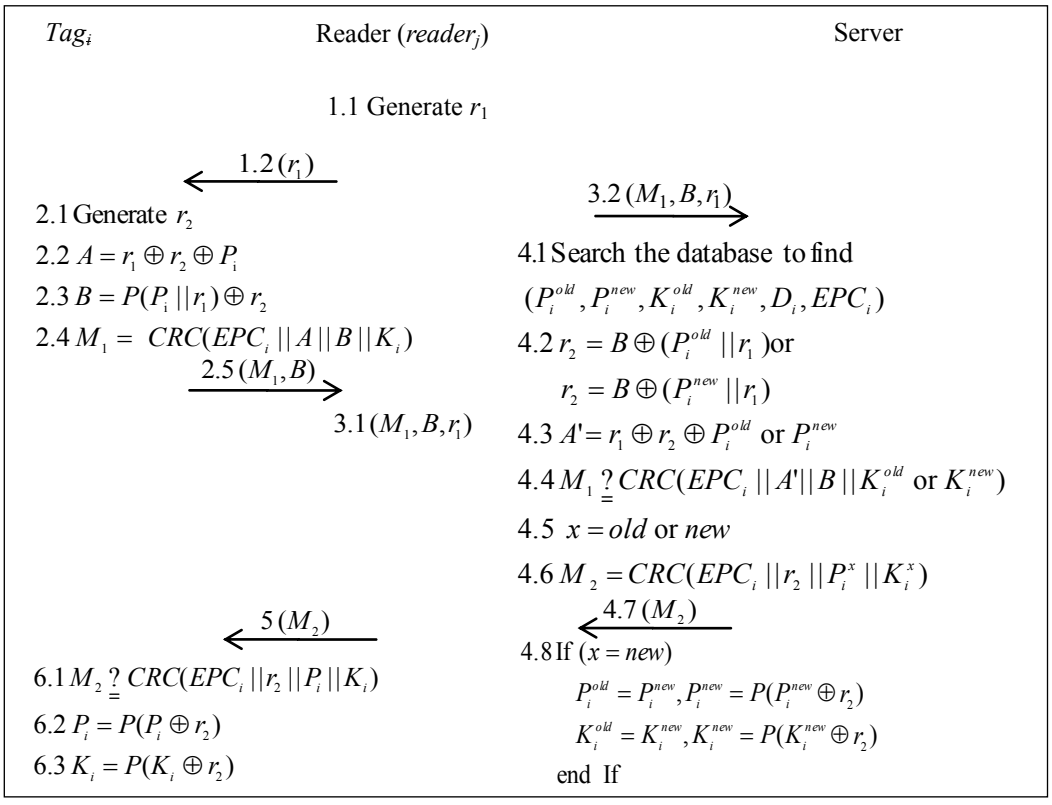

Figure 1. The mutual authentication scheme

If either of the above verifications is correct, tag_i is authenticated. The server sets $x = old$ (resp., new) if K_i^{old} (resp., K_i^{new}) is the one to make the verification correct. The server then calculate M_2 according to Eq. (10)

$$M_2 = CRC(EPC_i \,||\, r_2 \,||\, P_i^x \,||\, K_i^x) \tag{10}$$

After that, the server sends (M_2) to $reader_j$. But if the above verification does not hold, the authentication phase will be aborted. Moreover, if x = *new*, the server updates the database entry (P_i^{old} , P_i^{new}, K_i^{old}, K_i^{new}) of tag_i according to Eq. (11) and Eq. (12)

$$P_i^{old} = P_i^{new}, P_i^{new} = P(P_i^{new} \oplus r_2) \tag{11}$$

$$K_i^{old} = K_i^{new}, K_i^{new} = P(K_i^{new} \oplus r_2) \tag{12}$$

Step 5：After receiving the transmission message (M_2), $reader_j$ sends (M_2) to tag_i.

Step 6：Upon receiving message (M_2) from $reader_j$, tag_i uses the CRC function to do the following verification.

$$M_2 \stackrel{?}{=} CRC(EPC_i \,||\, r_2 \,||\, P_i \,||\, K_i) \tag{13}$$

If the above verification is correct, tag_i also updates P_i and K_i according to Eqs. (14)-(15).

$$P_i = P(P_i \oplus r_2) \tag{14}$$

$$K_i = P(K_i \oplus r_2) \tag{15}$$

IV. SECURITY ANALYSES AND COMPARISONS

In this section, the security of the proposed scheme is analyzed and compared with that of other schemes. Note that T, R, and S respectively represent tag_i, $reader_j$ and the server in the following context.

A. Security analyses

1) *Forged-server attack analysis*

In a forged-server attack, the adversary can pretend to be a legitimate server to pass the authentication $M_2 \stackrel{?}{=} CRC(EPC_i \,||\, r_2 \,||\, P_i \,||\, K_i)$ after eavesdropping on communication messages between tag_i and the server. Below we explain why the proposed scheme can resist the forged-server attack.

The adversary can easily get the information (M_1, M_2, r_1, B) from the communication messages between tag_i and $reader_j$, and between $reader_j$ and the server.

Message 1: R → T: (r_1)

Message 2: T → R: (M_1, B)

Message 3: R → S: (M_1, B, r_1`)

Message 4: R → T: (M_2)

However, when the adversary afterwards tries to obtain, by taking advantage of the CRC security flaw [14], the private information (EPC_i, P_i, K_i) stored in the server or the information (r_2, P_i, K_i) stored in tag_i, the adversary will fail. The reason is that r_2, P_i, K_i are updated after each authentication. Therefore, the adversary cannot calculate the correct communication parameter $M_2$from the intercepted messages to pass the authentication, where $M_2 = CRC(EPC_i \,||\, r_2 \,||\, P_i \,||\, K_i)$.

2) *Forged-tag attack analysis*

The adversary only needs to eavesdrop passively on the messages transmitted between tag_i and $reader_j$ and then initiates the forged-tag attack to try to supplant tag_i. The explanations below show why the proposed scheme can resist the forged-tag attack.

The adversary can easily obtain information (M_1, B, r_1) from the following messages transmitted between tag_i and $reader_j$.

Message 1: R → T: (r_1)

Message 2: T → R: (M_1, B)

Because the private information (EPC_i, P_i, K_i) shared between tag_i and $reader_j$ and random number r_2 stored in tag_i are not transmitted, the adversary has no way to uncover them. Therefore, the adversary cannot calculate the correct M_1 through the CRC security flaw [14] and pass the authentication, where $M_1 = CRC(EPC_i \,||\, A \,||\, B \,||\, K_i)$.

3) *MitM attack analysis*

When $reader_j$ interrogates tag_i, an adversary initiates the MitM attack to intercepts the messages sent between $reader_j$ and tag_i. Afterwards, the adversary pretends to be $reader_j$ (resp., tag_i) to forward modified or replayed messages to tag_i (resp., $reader_j$) for passing the authentication and delivering some forged information.

Because the server and tag_i do the authentication and then update their keys according some information securely embedded in the authentication information, it is impossible for an adversary to inject or modify forged information to pass the authentication and then affect the update of keys. The proposed scheme can thus resist the MitM attack.

4) *Tracking attack analysis*

Although the adversary cannot obtain the plaintext information of tag_i directly, it can track the tag's location if tag_i responds $reader_j$ interrogation queries with the same portion of information continuously.

In the proposed scheme, tag_i updates P_i, K_i after each successful authentication of the server. Furthermore, tag_i generates random number r_2 for the next response. Therefore, for different round of the communication, tag_i sends to $reader_j$ different information of (M_1, B) which is affected by the updated values of P_i, K_i and r_2. The proposed scheme can thus resist the tracking attack.

5) *Replay attack analysis*

An adversary obtains the information (M_1, B, r_1) transmitted between tag_i and $reader_j$, and then initiates the replay attack to try to spoof the server by transmitting previously obtained information to pass the authentication, where the information is obtained from the following messages.

Message 1: R → T: (r_1)

Message 2: T → R: (M_1, B)

We show below the proposed scheme can resist the replay attack.

The adversary tries to pass the authentication by replaying (M_1, B, r_1) later, but this will fail. The reason is that r_2, P_i, K_i are updated after each authentication to be r_2', P_i^{new}, K_i^{new} in the next round, and thus the legitimate M_1, B in the next round (denoted by M_1' and B' respectively) should be $M_1' = CRC(EPC_i \| A' \| B' \| K_i^{new})$ and $B' = (P_i^{new} \| r_1') \oplus r_2'$. Therefore, the adversary cannot replay the obtained information (M_1, B, r_1) to pass the authentication.

6) *Forward secrecy attack analysis*

In the forward secrecy attack, the adversary compromises keys shared by tag_i and $reader_j$ and then tries to calculate previous keys to reveal information transmitted earlier between tag_i and $reader_j$.

Suppose that the adversary has compromised the secret keys P_i and K_i shared by tag_i and the server. Since P_i and K_i are calculated by evoking the PRNG, which is equivalent to a one way hash function, on pervious keys of P_i and K_i. Therefore, no previous keys of P_i and K_i can be obtained even when P_i and K_i are compromised at some instance. The proposed scheme can thus resist the forward secrecy attack.

7) *DoS attack analysis*

An adversary initiates the DoS attack by intercepting the message (M_2) sent from $reader_j$ to tag_i, where $M_2 = CRC(EPC_i \| r_2 \| P_i) \oplus K_i$. In that way, the adversary prevents tag_i from updating the shared keys and makes the shared keys stored on the server different from those stored on tag_i. Therefore, the server (and hence $reader_j$) and tag_i cannot communicate properly henceforth.

To resist the DoS attack, the new and the old keys (P_i^{old}, K_i^{old} P_i^{new} K_i^{new}) are all stored on the server. In the case that tag_i updates the keys unsuccessfully; the server can still allow tag_i to pass the authentication and resynchronizes the keys with tag_i for later communication. Therefore, the proposed scheme can resist the DoS attack.

B. Comparisons

Because the proposed scheme uses only ultralightweight operations, such as the RNG, PRNG and the CRC operator, it conforms to the EPC C1G2 standard. Therefore, only the schemes [11-13] that conform to the EPC C1G2 standard have been compared with the proposed scheme.

Table 2 compares the proposed scheme with related ones in terms of the communication cost (i.e., the number of bits transmitted) during the authentication phase. We use numbers to exemplify the expressions in Table 2 under the assumption that the hello message, key and tag identity are 128 bits, the output of L_{PK} by XOR key with PRNG and the output of L_{CK} by XOR key with CRC are 128 bits and the output of the cyclic redundancy check is 16 bits. By Table 2, we can observe that the proposed scheme has lower communication cost than other schemes.

Table 3 compares the proposed scheme with related schemes in terms of the computation cost during the authentication phase. In Table 3, T_{XOR}, T_{PRNG}, T_{CRC}, and T_H are the execution time or the computation cost for the XOR, PRNG, CRC and hash function operation, respectively, and n is the number of tag_i. Note that the exclusive-or operation are very low computation-cost operations and the computation costs of other operations are of the ascending order: T_{PRNG}, T_{CRC} and T_H. By Table 3, we can observe that the proposed scheme, Chien and Chen's scheme [11] and Chen and Deng's scheme [12] have nearly the same computation cost. Only the Huang and Jiang's scheme [13] has lower computation cost than other schemes.

Table 4 shows the resistible ability of various attacks among Chien and Chen's scheme [11], Chen and Deng's scheme [12], Huang and Jiang's scheme [13] and the proposed scheme.

Table 2. Communication cost comparisons

Schemes	Communication costs
Chien and Chen's [11]	$2L_N + 2\ L_{CK}$ (=512 bits)
Chen and Deng's [12]	$2L_{Hello} + 3L_N + 1L_{CRC}$ (=656 bits)
Huang and Jiang's [13]	$1L_{ID} + 1L_N + 1L_K + 2L_{PK}$ (=640 bits)
Proposed Scheme	$2L_N + 2L_{CRC}$ (=288 bits)

Note that L_{Hello}, L_N, L_{PK}, L_{CK}, L_{CRC}, L_{ID} and L_K are the bit length of the hello message, RNG output, XOR key with PRNG output, XOR key with CRC output, CRC output, key and identity, respectively.

Table 3. Computation cost comparisons

Schemes	Computation costs	
	Tag_i	Server
Chien and Chen's [11]	$2T_{XOR} + 2T_{CRC} + 2T_{PRNG}$	$1T_{XOR} + 1T_{CRC} + 2T_{PRNG} + n/2T_{Comp}$ ($T_{Comp} = 2T_{XOR} + 2\ T_{CRC}$)
Chen and Deng's [12]	$5T_{XOR} + 2T_{CRC}$	$3T_{XOR} + 1T_{CRC} + n/2T_{Comp}$ ($T_{Comp} = 2T_{XOR} + 1T_{CRC}$)
Huang and Jiang's [13]	$6T_{XOR} + 5T_{PRNG}$	$6T_{XOR} + 3T_{PRNG} + 1T_H + 1T_{Comp}$ ($T_{Comp} = 4T_{XOR} + 2\ T_{PRNG}$)
Proposed Scheme	$5T_{XOR} + 2T_{CRC} + 2T_{PRNG}$	$2T_{XOR} + 1T_{CRC} + 2T_{PRNG} + n/2T_{Comp}$ ($T_{Comp} = 4T_{XOR} + 2T_{CRC}$)

Note that T_{XOR}, T_{PRNG}, T_{CRC} and T_H are the execution time for the XOR, PRNG, CRC and hash function operation, respectively, and n is the number of tag_i.

Table 4. Security comparisons

Attacks \ Schemes	Chien and Chen's [11]	Chen and Deng's [12]	Huang and Jiang's [13]	Proposed Scheme
Resist forged-server attack	No	No	Yes	Yes
Resist forged-tag attack	No	No	Yes	Yes
Resist MitM attack	No	Yes	Yes	Yes
Resist tracking attack	No	No	No	Yes
Resist replay attack	No	No	Yes	Yes
Resist forward secrecy	No	No	Yes	Yes
Resist DoS attack	No	No	Yes	Yes

Table 4 shows the comparisons of schemes in terms of what attacks they can resist. By Table 4, we observe that Chen and Deng's scheme only can resist the MitM attack and Huang and Jiang's scheme cannot resist the tracking attack. Moreover, Chien and Chen's scheme all suffer from the tracking, MitM, replay and DoS attacks. However, the proposed scheme can resist the forged-tag, forged-server, MitM, tracking, replay, forward secrecy and DoS attacks.

V. CONCLUSION

This paper has proposed an ultimate ultralightweight and efficient authentication scheme conforming to the EPC C1G2 RFID standard to resist various attacks, such as the forged-tag, forged-server, MitM, tracking, replay, forward secrecy and DoS attacks. The proposed scheme uses only ultralightweight operators, like the RNG, PRNG, CRC and XOR, on tags to conform to the EPC C1G2 standard. Furthermore, thorough comparisons and security analyses have been performed for the proposed scheme to demonstrate its superiority to other related schemes in terms of the communication cost, computation cost and security.

REFERENCES

[1] S. L. Garfinkel, A. Juels, R. Pappu, "RFID privacy: An overview of problems and proposed solutions," IEEE Security & Privacy Magazine, Vol. 3, pp. 34-43, 2005.

[2] R. Weinstein, "RFID: A technical overview and its application to the enterprise," IT Professional, Vo. 7, No. 3, pp. 27–33, 2005.

[3] Z. Chen, L. Liu, D. Yan, Y. Shen, H. Wang, "Research on the Authentication Mechanisms of RFID System," in Proc. of 2nd International Conference on e-Business and Information System Security. pp. 1-4, May, 2010.

[4] E. O. Blass, A. Kurmus, R. Molva, T. Strufe, "PSP: Private and secure payment with RFID," Computer Communications, Vol. 36, pp. 468-480, February, 2013.

[5] R. Doss, S. Sundaresan, W. Zhou, "A practical quadratic residues based scheme for authentication and privacy in mobile RFID systems," Ad Hoc Networks, Vol. 11, pp. 383-396, January, 2013.

[6] Y. P. Liao, C. M. Hsiao, "A secure ECC-based RFID authentication scheme integrated with ID-verifier transfer protocol," Ad Hoc Networks, March, 2013. http://dx.doi.org/10.1016/j.adhoc.2013.02.004.

[7] G. N. Khan, G. Zhu, "Secure RFID Authentication Protocol with Key Updating Technique," in Proc. of 213 22nd International Conference on Computer Communications and Networks (ICCCN), pp. 1-5, August, 2013.

[8] X. Yi, L. Wang, D. Mao, Y. Z Cho, "An Gen2 Based Security Authentication Protocol for RFID system," in Proc. of the 2012 International Conference on Applied Physics and Industrial Engineering, Vol. 24, pp. 1385-1391, 2012.

[9] Z. Y. Wu, S. C. Lin, T. L. Chen, C. Wang, "A Secure RFID Authentication Scheme for Medicine Applications," in Proc. of 2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), pp. 175-181, July, 2013.

[10] EPCglobal web site: http://www.epcglobalinc.org/

[11] H. Y. Chien, C. H. Chen, "Mutual authentication protocol for RFID confirming to EPC Class 1 Generation 2 standards," Computer Standards & Interfaces, Vol. 29, Issue 2, pp. 254-259, 2007.

[12] C. L. Chen, Y. Y. Deng, "Conformation of EPC class 1 generation 2 standards RFID system with mutual authentication and privacy protection," Engineering Applications of Artificial Intelligence, Vol.22, pp. 1284–1291, 2009.

[13] Y. C Huang and J. R Jiang, "An Ultralightweight Mutual Authentication Protocol for EPC C1G2 RFID Tags," in Proc. of 2012 International Symposium on Parallel Architectures, Algorithms and Programming (PAAP'12), pp. 133-140, December, 2012.

[14] P. L. Pedro, C. H. C. Julio, M. E. T. Juan, C. A. Jan, "Cryptanalysis of an EPC Class-1 Generation-2 standard compliant authentication protocol," Engineering Applications of Artificial Intelligence(2011), Vol. 24, No. 6, pp. 1061-1069, 2011.

[15] E. Y. Choi, D. H. Lee, J. I. Limb, "Anti-cloning protocol suitable to EPCglobal Class-1 Generation-2 RFID systems," Computer Standards & Interfaces, pp. 1124-1130, November, 2009.