Noname manuscript No.
(will be inserted by the editor)

A secure ownership transfer protocol using EPCglobal Gen-2 RFID

Chin-Ling Chen · Yu-Chung Huang · Jehn-Ruey Jiang

Received: date / Accepted: date

Abstract Radio Frequency Identification (RFID) is a relatively new technology. In recent years, it has been shown to be convenient and feasible in many applications. However, there are security issues which need to be addressed. Due to the wireless transmission of the RFID system, malicious people can gain the information in the RFID tags, and the user's privacy is invaded. Although there have been many protection methods proposed for RFID security, the system has remained vulnerable to various attacks. In this paper, we propose a conforming of the EPCglobal Class 1 Generation 2 standards RFID ownership transfer protocol with provable security. The proposed scheme can resist several attacks and ensure a secure transaction.

Keywords Security, EPC, Mutual authentication, Ownership transfer, RFID

C-L Chen
Department of Computer Science and Information Engineering Chaoyang University of Technology Taichung 41349, Taiwan, R.O.C.
Tel: +886-4-23323000 Ext. 4761
Fax: +886-4-23742375
E-mail: clc@mail.cyut.edu.tw

Y-C Huang
Department of Computer Science and Information Engineering National Central University, Jhongli City, Twiain, R. O. C.
E-mail: 985402024@cc.ncue.edu.tw

J-R Jiang
Department of Computer Science and Information Engineering National Central University, Jhongli City, Twiain, R. O. C.
E-mail: : jrjiang@csie.ncu.edu.tw

1 Introduction

1.1 Background

Radio Frequency Identification is becoming increasingly popular for use in various applications. An RFID system consists of antennae, hosts, readers, and tags [17]. An RFID tag is a small chip attached to an object and is used in conjunction with an RFID reader. RFID readers can be PDAs, mobile phones, or any type of device capable of querying the object identity stored in a RFID tag. RFID readers also can retrieve detailed information about the object stored in a backend server database.

When a reader sends a request message to a tag, the tag responds via radio frequency signals. At their most basic, passive tags simply transmit a static serial number in response to a reader's query. This renders RFIDs susceptible to various latent attacks (denial of service attack, man-in-the-middle attack, replay attack, forged-server and forged-tag attacks etc.) [10].

(1) Privacy: If the Electronic Product Code (EPC) in the tag is not encrypted, the attacker can obtain the message from the user's RFID tag. Anyone could use a reader to obtain the EPC in the tag and query the database for the related information and the privacy of the tag owner would be violated.

(2) Tracking: For a tag, the same message is always given to a reader. If an attacker intercepts a message from the user's RFID tag, the attacker can track the tag and forward the message after copying the message.

Recently, an RFID Class 1 Generation 2 (C1G2) standard has been issued by EPCglobal [11]. It defines RFID standards as follows:
(1) The RFID tag is passive, and it is triggered by readers.

(2) The RFID tag communicates on the UHF band (800-960 MHz) and its communication range is from 2 m to 10 m.
(3) The RFID tag only supports on-chip 16-bit Pseudo-Random Number Generators (PRNG), and a 16-bit Cyclic Redundancy Code (CRC) checksum is used to detect errors in transmission.
(4) The RFID's privacy protection mechanism must make the tag permanently unusable once it receives the kill command with a valid 32-bit kill PIN (e.g., tags can be killed at the point-of-sale).
(5) Read/write to RFID tag memory is allowed only once it is in secure mode (i.e., after receiving an access command with a valid 32-bit access PIN).
RFID tags can only be considered as storage media, not as smart tags. Thus, access to computing resources is limited. There are about 500-5000 logic gates in current RFID tags. Thus, similar encryption and hash function mechanisms [1,2,7] are infeasible for EPCglobal C1G2 RFID tags. None of these protocols conforms to EPCglobal C1G2 RFID standards.

To overcome security threats, Juels [1], Duc et al. [8], and Karthikeyan and Nesterenko [16] proposed new schemes for GEN-2 RFID-conformed tags. In their schemes, only lightweight operations (PRNG and CRC) supported on a GEN-2 RFID tag were used; their schemes could be implemented on the resource-limited GEN-2 RFID tags, thus replacing the use of hash functions, public key cryptography, and conventional encryptions.

In addition, the literature reviews [7,18] have addressed RFID tag-related sources. However, the numerous of previously proposed schemes [8,12,13,15,16] could be implemented on EPCglobal C1G2 RFID tags. In 2009, Pedro et al. [15] proposed a cryptanalysis of a novel authentication protocol conforming to EPCglobal C1G2 RFID standards. Pedro et al. proved that Chien et al.'s method [12] suffered from both forged-tag and forged-server attacks.

Other methods [13,14,19] have proposed ownership transfer, but these still utilized the encryption/decryption method, and regarded the RFID tag as a "smart tag." Since current RFID tag logic gates number about 500-5000, computing resources are limited, which makes such schemes impractical. That is, traditional symmetrical or asymmetrical encryption [4–6] is not suitable for tag's operations.

1.2 Review of related works

Due to the previous method [15], Pedro et al. proposed a cryptanalysis of a novel authentication protocol conforming to EPCglobal C1G2 RFID standards. Pedro et al. proved that Chien et al.'s scheme [12] suffered from forged-tag and forged-server attacks, as did other previous methods [9,12,18]; thus, the problem of determining how to design a conforming EPCglobal C1G2 RFID standard became an important research issue. Moreover, some researchers [14,19] have adapted RFID technology to ownership transfer applications, which is of considerable interest. Our aim, therefore, was to design a provable secure and conforming EPCglobal GEN-2 RFID ownership transfer protocol. The related works are reviewed in the following section.

(1) Karthikeyan-Nesterenko's scheme [18]:

We have illustrated a communication scenario using Karthikeyan-Nesterenko's scheme in Fig 1. The reader coordinates a value K with the tag during the registration stage, and the tag and the reader store two P*P square matrices, M_1 and M_2. This scheme uses the AND operation but, while it conforms to EPC Class 1 Generation 2 standards, it cannot prevent illegal tag access [1]. If the attacker replaces the current Z with a previous Z' , the attacker can replay Y' in the next session to trick the tag into wrongly accepting the request and so access the tag. Furthermore, the value X , which is stored in the tag, is fixed. Since there is no random value involved in this scheme, it cannot avoid location traces. This scheme also has a high database loading requirement. In order to determine the correct , the reader must search $K = KM^{-1}$ for all K and M_1^{-1} in the database.

(2) Duc et al.'s scheme [9]:

Duc et al. proposed a tag-to-backend database authentication protocol. The security of Duc et al.'s protocol was based on key synchronization between tags and the backend database. Fig. 2 illustrates a communication scenario based on the key synchronization scheme.

The last message of the protocol is comprised of an End Session command, which is sent to both tag and reader. Interception of one of these messages causes a synchronization loss between the tag and the server, which means the tag and the reader will no longer be able to authenticate, which is an extremely serious situation. This protocol also presents backward secrecy problems which compromise the EPC by allowing an attacker to trace back all past communications.

(3) Chien and Chen's scheme [12]

Chien and Chen's proposed scheme is based on the EPCglobal C1G2 standards, where PRNG and CRC are supported on the passive tags. They assumed that an attacker could monitor and modify the communications between the reader and the tags, but the communication between the reader and the backend server was secure. The passive tags were vulnerable, and the contents of a tag could be derived by the attacker once

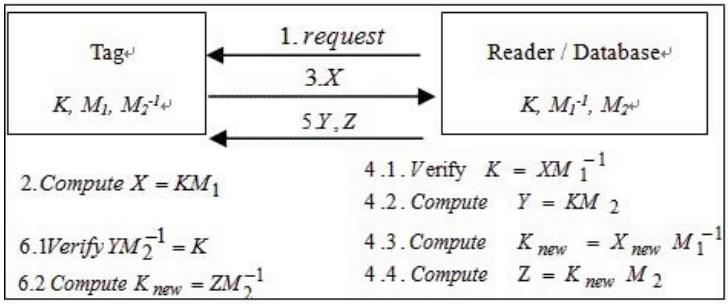

Fig. 1 Scenario using Karthikeyan-Nesterenko's scheme

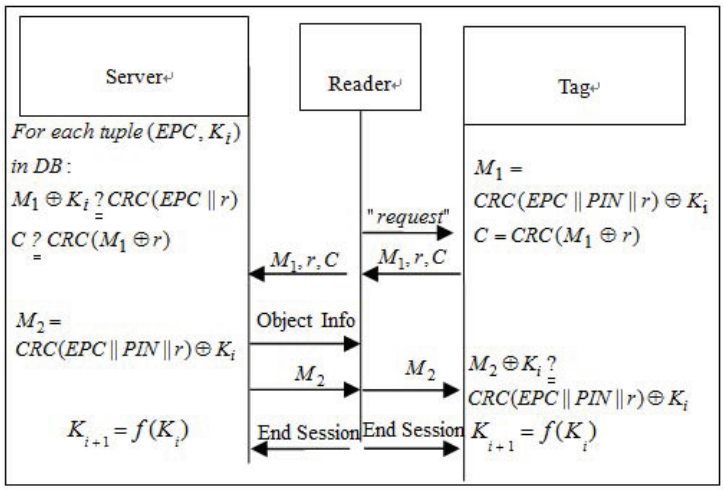

Fig. 2 Scenario using Duc et al.'s scheme

it was compromised. Fig. 3 illustrates a communication scenario utilizing a mutual authentication scheme.

Each tag shares with the reader some private information: EPC, authentication key (K_x) and access key (P_x). This information is used to build messages M_1 and M_2 in order to prove its authenticity. However, an attacker is able to supplant a legitimate tag via Pedro et al's proof [15].

(4) Ownership transfer scheme [14]:

Another application is used in ownership transfer. In Fig. 4, Osaka et al.'s ownership transfer scheme is illustrated. Once the reader proposes a request (which includes a random value r), and tags respond a ($a = H(E_k(ID) \oplus r), E_k()$) , means encrypting message with symmetric key k; means the hash function) to the reader, the database verifies the hash message. Then, it generates new encrypted information $e = E_k(ID) \oplus E'_k(ID)$, and responds by sending the ID information $info(ID)$ and e to the reader. Subsequently, the reader transfers e to the tag, and then the tag updates $E'_k(ID)$ by $E'_k(ID) = e \oplus E_k(ID)$.

This scheme has several security problems. First, it cannot prevent location tracing because the random value r is fixed. Second, the tag sends messages without authenticating the reader first. Third, the attacker can modify the tag's message, thus causing tag updates to consist of erroneous encrypted ID. Additionally, the tag requires a hash calculation, which does not conform to EPCglobal Class 1 Generation 2 standards. Finally, the entire database must be searched to compare the tag's response: $a = H(E_k(ID) \oplus r)$; resulting in the database loading being too high.

In this paper, we have designed an EPCglobal Class 1 Generation 2 standard-conforming RFID ownership transfer protocol which can resist several types of attacks. The rest of the paper is organized as follows: in Section 2, the Preliminaries introduce the related cyclic redundancy codes; the proposed protocol is presented in Section 3; the results are analyzed and discussed in Section 4; and finally, conclusions are given in Section 5.

2 Preliminaries

2.1 EPCglobal C1G2 standards

Below is an introduction to the notations used in our scheme. In the EPCglobal C1G2 standards, the computing resources of tags are limited. Tags can only op-

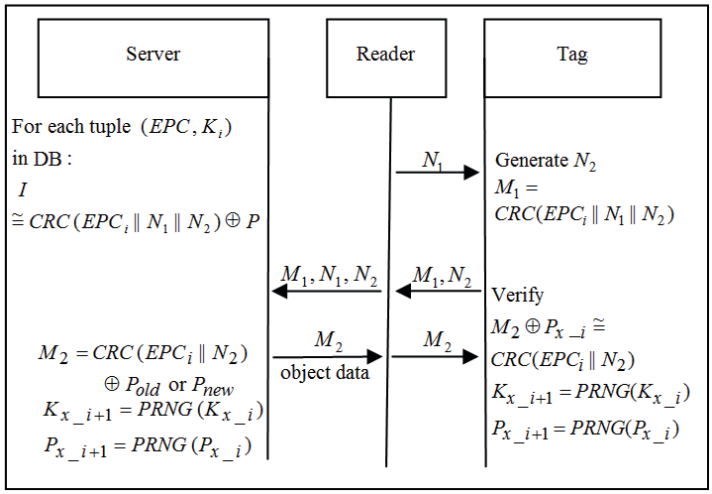

Fig. 3 Scenario using Chien and Chen's scheme

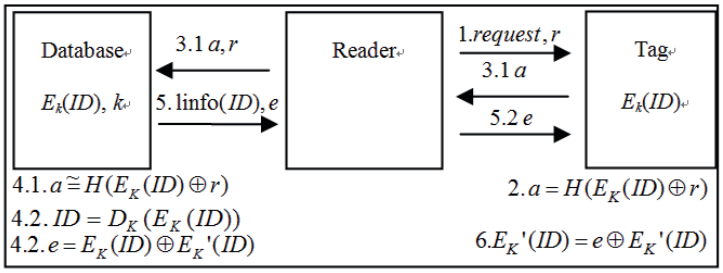

Fig. 4 Scenario using Karthikeyan-Nesterenko's scheme

erate CRC functions, exclusive-or operations, and generate random numbers; other complex operations (such as hash functions, symmetric encryption and asymmetric encryption) do not conform to the standards. According to the EPCglobal C1G2 standards, RFID tags store two keys: the kill key ($Kill\_key_i$) and the access key ($Access\_key_i$), the i means the index.

1. Kill key ($Kill\_key_i$): used to verify the legitimacy of the transmitted messages.
2. Access key ($Access\_key_i$): used to write data to an EPCglobal C1G2 RFID tag's memory.

2.2 Cyclic Redundancy Check - CRC

The Cyclic Redundancy Check (CRC) is a checksum algorithm 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.

2.3 CRC properties

On the basis of the CRC lineal property, Pedro et al. proposed the cryptanalysis of a novel authentication protocol to show that Chien et al.'s scheme [12] had faults. The following theorem is Pedro et al.'s basis for cryptanalysis [15].

Theorem 1: Let F2[x] be the ring of polynomials over F2 (binary field). For any CRC (independent of its divider polynomial) and for any values $a, b, candd \in F2[x]$, it holds that

$$CRC(a||b) \oplus CRC(c||d) = CRC(a \oplus c||b \oplus d) \quad (1)$$

Proof: From the definition in (1) above, one can write:

$$CRC(a||b) = (a \cdot x^n \oplus b) \cdot x^n \oplus d_1(x) \cdot p(x) \quad (2)$$

$$CRC(c||d) = (c \cdot x^n \oplus d) \cdot x^n \oplus d_2(x) \cdot p(x) \quad (3)$$

for certain polynomials and F2[x]. Substituting these values in the left side of (1) we obtain the following:

$$(a \cdot x^n \oplus b) \cdot x^n \oplus d_1(x) \cdot p(x) \oplus (c \cdot x^n \oplus d) \cdot x^n \oplus d_2(x) \cdot p(x) \quad (4)$$

Rearranging terms in this expression we get:

$$((a \oplus c) \cdot x^n \oplus (b \oplus d) \cdot x^n \oplus (d_1(x) \oplus d_2(x)) \cdot p(x) \quad (5)$$

That is, the corresponding expression for (analogous to Eq. (3) and Eq. (4)).

Corollary 1: In particular, if in Eq. (2) we have , then,

$$\begin{aligned} & CRC(a||b) \oplus CRC(a||d) \\ &= CRC((a \oplus a)||(b \oplus d)) \\ &= CRC(0||b \oplus d) \\ &= CRC(b \oplus d) \end{aligned} \tag{6}$$

because $0 \cdot x^n \cong 0 \cdot p(x)$ Corollary 2: In Eq. (2), if $c = b$, then,

$$\begin{aligned} & CRC(a||b) \oplus CRC(b||d) \\ &= CRC(a||(b \oplus b)||d) \\ &= CRC(a||0||d) \\ &= CRC(a||d) \end{aligned} \tag{7}$$

Based on Pedro et al. proposed authentication protocol property, we have proposed a secure RFID mutual authentication scheme that conforms to EPCglobal C1G2 standards and improves security.

3 Proposed ownership transfer scheme

3.1 Environmental conditions

In our proposed protocol, we have assumed that the RFID tags are used in high cost products (such as notebooks, mobile phones, PDAs, or any type of device with the capability required). Our scheme has been divided into three phases: (1) Initialization phase, (2) Purchase phase, and (3) Ownership transfer phase. A brief scenario is illustrated in Fig. 5.
Step1:A product having an RFID tag is sent to a commercial agent via various channels.
Step2:Consumer A wants to buy the product and to verify it via mutual authentication.
Step3:Original Consumer A transfers ownership to new Consumer B.

3.2 Notation

M_{req}: request message
N_i: nonce
$\oplus$:exclusive-or operation
PID_i:pseudonym identification code of the ith tag
RID_iith reader's identity
SK_i:session key shared by server and reader
E_{SK_i}:use the session key SK_i to encrypt message m
D_{SK_i}:use the session key SK_i to decrypt message m
EPC_i:96-bit EPC (Electronic Product Code) of the ith tag
$CRC(x)$:Cyclic Redundancy Check (CRC) function
$Kill\_key_i$:32-bit kill key of the ith tag
$Access\_key_i$:32-bit access key of the ith tag
$PRNG$:32-bit pseudo-random number generator
$A \cong Bi$:compare whether A is equal to B
$||$:concatenation operation
$DATA_i$:product information of the ith tag
SN:serial number of the product
OTi:ownership transfer message of the product
PW:ownership transfer message of the product
PW_{new}:new consumer's password

3.3 Initialization phase

The OEM (Original Equipment Manufacturer), the commercial agent, and the authorized agent's reader, obtain the identification from the brand company via online authentication to verify whether the identification is correct according to the end server in advance.

Each tag and the authorized agent's reader must register with the brand company server. The backend server issues the corresponding Electronic Product Code (EPC_i), serial number (SN), pseudonym identification (PID_i), an initial kill key ($Kill\_key_i$), and an initial access key ($Access\_key_i$) to a tag. The server also issues the reader identification (RID_i) and the session key (SK_i) to a reader.

3.4 Purchase phase

In this section, we describe how a consumer can identify counterfeit products and buy genuine products from commercial agents.

The tags and the servers perform the mutual authentication procedures and verify whether each is legal. The purchase scenario is illustrated in Fig. 6. The pseudonym and key updating procedures are also executed for each transaction.
Step 1: When the reader wants to access a tag, it generates N_1 and computes

$$A = CRC(N_i) \tag{8}$$

Then, it sends request message M_{req} and A to the tag.
Step 2: Upon receiving the request message, the tag generates a nonce N_2 and computes X, B and C_T as follows:

$$X = N_2 \oplus Kill\_key_i \tag{9}$$

$$B = CRC(A||X||SN) \tag{10}$$

$$C_T = CRC(EPC_i||B) \tag{11}$$

Then, it responds (C_T, N_2, PID_i) to the reader.
Step 3: After receiving the tag's response, the reader will involve A and its identity RID_i into the transmission messages and forward ($C_T, N_2, PID_i, A, RID_i$) to the server.
Step 4: When the server receives the authentication request from the reader, the server checks whether PID_i

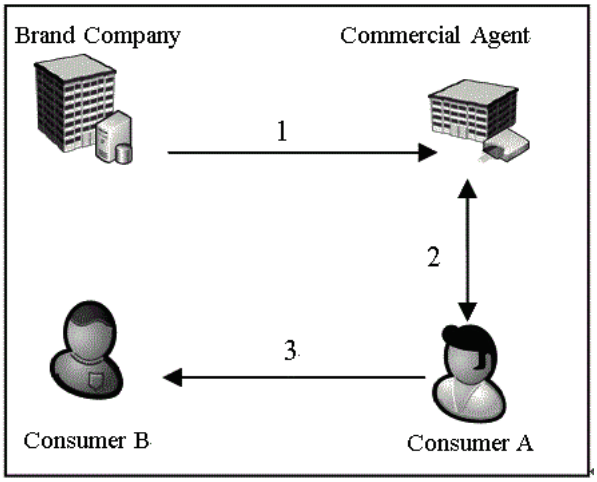

Fig. 5 Scenario for our scheme

and RID_i , the tag's pseudonym and reader's identification, existing in the database equal the received PID_i and RID_i . If the above verification is correct, the server uses A, N_2, SN and the kill key ($Kill\_key_i$) to calculate $X' and\ B'$, as follows:

$$X' = N_2 \oplus Kill\_key_i \tag{12}$$
$$B' = CRC(A||X||SN) \tag{13}$$

Then, the server verifies whether C_T is correct, as follows:

$$C_T \cong CRC(EPC_i||B') \tag{14}$$

If Eq.(14) does not hols, the server will terminate the session. If the equality holds, the server computes Y, C_1, and C_S, as follows:

$$Y = N_2 \oplus Kill\_key_i \tag{15}$$
$$C_1 = E_{SK_i}(DATA_i) \tag{16}$$
$$C_S = CRC(EPC_i||Y||Access\_key_i) \oplus Kill\_key_i \tag{17}$$

If the equality holds, the server transmits the messages (C_S, C_1) to the reader.

Moreover, the server updates the pseudonym identification (PID_i), kill key ($Kill\_key_i$), and access key ($Access\_key_i$) simultaneously for the next session communication as follows:

$$PID_{i_{new}} = PRNG(PID_i) \tag{18}$$
$$Kill\_key_{i_{new}} = PRNG(Kill\_key_i) \tag{19}$$
$$Access\_key_{i_{new}} = PRNG(Access\_key_i) \tag{20}$$

Step 5: After receiving the transmission messages (C_S, C_1), the reader forwards C_S to the tag and obtains the product information $DATA_i$ as follows:

$$DATA_i = D_{SK_i}(C_1) \tag{21}$$

Step 6: Upon receiving the message C_S of the reader, the tag uses the CRC function to verify its correctness as follows:

$$C_S \cong CRC(EPC_i||Y||Access\_key_i \oplus Kill\_key_i \tag{22}$$

If Eq.(22) does not hold, then the protocol aborted. The tag also updates the pseudonym identification code (PID_i), kill key ($Kill\_key_i$) and access key ($Access\_key_i$) simultaneously, as with the server's operations in Step 4.

$$PID_{i_{new}} = PRNG(PID_i) \tag{23}$$
$$Kill\_key_{i_{new}} = PRNG(Kill\_key_i) \tag{24}$$
$$Access\_key_{i_{new}} = PRNG(Access\_key_i) \tag{25}$$

Step 7: Once the consumer decides to purchase the product, the consumer must key in a password to the reader, and then the authorized commercial agent forwards the hash value of password $H(PW)$ to the brand company's server for registration via a secure channel.
Step 8: The server'ss database will be updated relative to this record ($PID_i, RID_i, Kill\_key_i, Access\_key_i, SK_i, DATA_i, SN, H(PW)$) and these parameters sent to the authorized commercial agent.

3.5 Ownership transfer phase

If Consumer A wants to resell and transfer ownership of the product to Consumer B, he/she can go to the authorized agent to demonstrate the legitimacy of the product to B. If $DATA_i$ is the same as the original, Consumer B only need trust the authorized reader. Fig. 7 shows the flow chart of the ownership transfer phase.
Step 1:Consumer A (initial owner of the product) enters PW (self-chosen) at the purchase time and generates a hash value $H(PW)$, Consumer B (new consumer of the product) enters a new password PW_{new} and generates

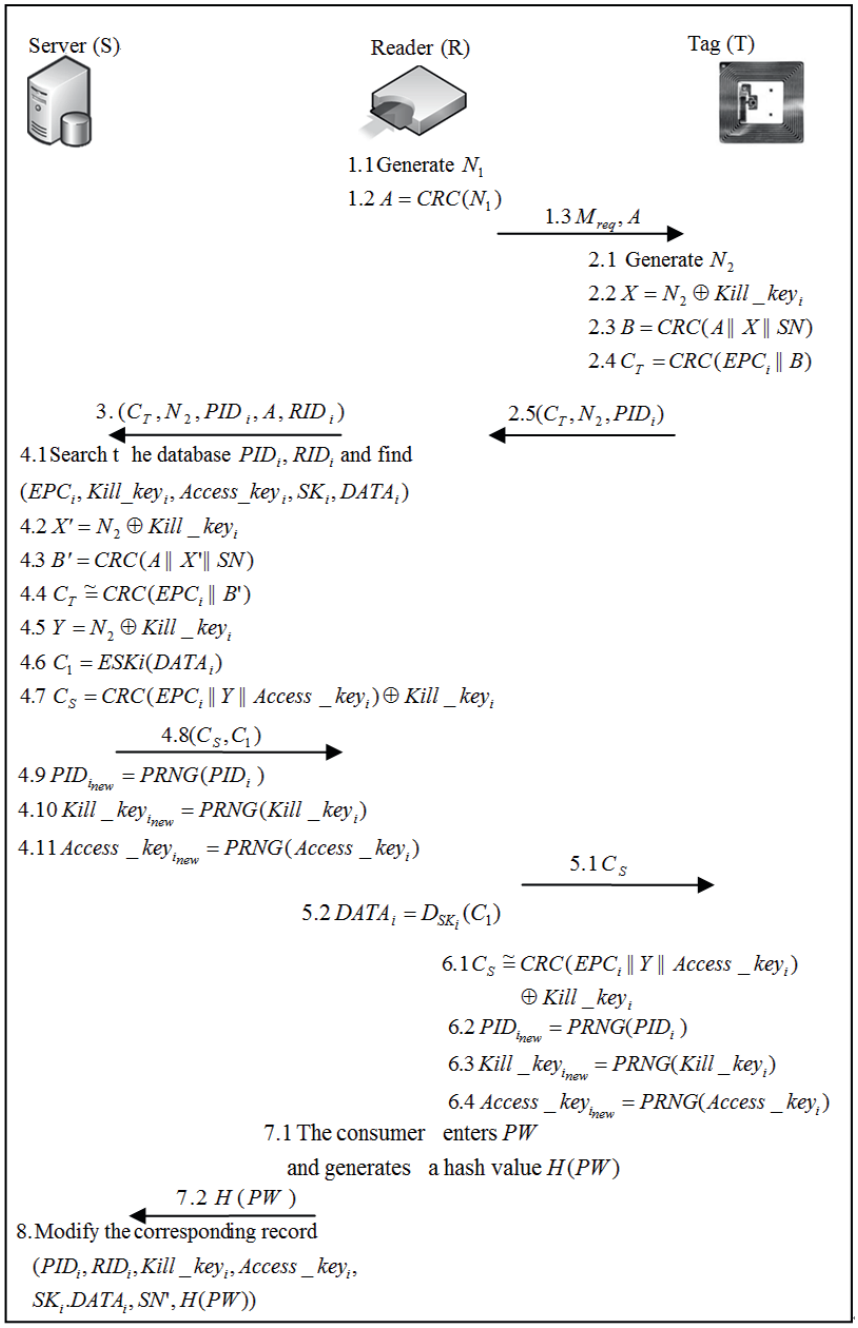

Fig. 6 Scenario for our scheme

a hash value $H(PW_{new})$, and the reader uses its own session key to generate the message OT as follows:

$$OT = E_{SK_i}(H(PW), H(PW_{new})) \quad (26)$$

After that, the authorized reader will transmit the message (OT, RID_i, PID_i) to the server.

Step 2: When the server receives the ownership transfer request from the reader, the server use the session key SK_i to decrypt the message OT as follows:

$$(H(PW), H(PW_{new})) = D_{SK_i}(OT) \quad (27)$$

and then, check if the $H(PW)$ of the PID_i which is in the database is equal to the received $H(PW)$. The server generates a new serial number SN' and updates relevant records: $(PID_i, RID_i, Kill\_key_i, Access\_key_i, SK_i, DATA_i, SN', H(PW_{new})$

The server uses the session key to generate message C_2 as follows:

$$C_2 = E_{SK_i}(DATA_i, Kill\_key_i, Access\_key_i, SN') \quad (28)$$

Then, the server calculates Y as follows:

$$Y = CRC(Access\_key_i || SN') \oplus Kill\_key_i \quad (29)$$

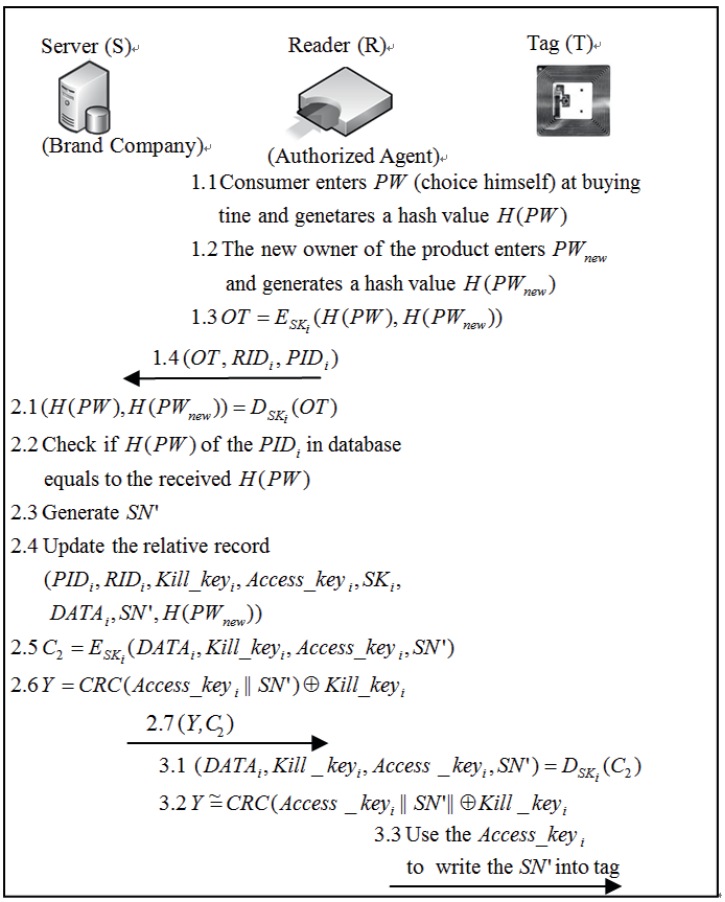

Fig. 7 Scenario for our scheme

The server transmits the messages (Y, C_2) to the reader.

Step 3: The reader uses the session key to verify the correction of as follows:

$$(DATA_i, Kill\_key_i, Access\_key_i, SN') = D_{SK_i}(C_2) \quad (30)$$

$$Y \cong CRC(Access\_key_i || SN') \oplus Kill\_key_i \quad (31)$$

The verification is correct, the reader prints the transaction receipt (including $DATA_i$) for Consumer B, if Eq.(31) does not hold, then the protocol aborted. The reader uses the access key $(Access\_key_i, SN\ )$ to write the serial number SN' (i.e. $SN = SN'$) into the tag.

4 Security Analysis and Discussion

In this section, we have analyzed the security of our scheme and compared it with others.

4.1 Security Analysis

The proposed scheme also proved beneficial to RFID product ownership transfer and could resist several attacks, based on Pedro et al.'s [15] cryptanalysis.

4.1.1 Resist forged-tag attack

In order to accomplish this attack, an adversary only needs to listen to iteration messages between the reader and the legitimate tag. Each tag shares with the reader some private information: EPC_i, kill key $(Kill\_key_i\ )$ and access key $(Access\_key_i\ )$. This information is used to build messages A and C_T in order to prove its authenticity. However, a passive attacker eavesdropping on messages will be able to supplant a legitimate tag. The following iteration messages are transmitted and can be intercepted by an attacker between the reader and legitimate tag as described below:

(1) R $\rightarrow$ T : M_{req}, A

(2) T $\rightarrow$ R : C_T, N_2, PID_i

Once the attacker holds the information of M_{req}, A, C_T, $N_2, and PID_i$, the attacker can build a message as follows:

$$C_T = CRC(EPC_i||B) \quad (32)$$

Although the attacker does not know the private information stored in the tag $(EPC_i, Kill\_key_i\ and\ Access\_key_i)$, the message C'_T can be easily computed. The different values of $C_T and\ C'_T$ are calculated by the XOR oper-

ation and, according to Corollary 1, the following expression can be derived to get:

$$C_T \oplus C'_T = CRC(EPC_i||B) \oplus CRC(EPC_i||B') \quad (33)$$

Message C'_T is easily computed as follows:

$$\begin{aligned} C'_T &= C_T \oplus CRC(B \oplus B') \oplus CRC(N_2 \oplus N'_2) \\ &= CRC(EPC_i||B||N_2) \oplus \\ &\quad CRC((B \oplus B')||(N_2 \oplus N'_2)) \\ &= CRC(EPC_i||B'||N'_2) \end{aligned} \quad (34)$$

In our scheme, value B, which involved C_T , is not transmitted between the reader and the tag. Therefore, the attacker cannot calculate the next correct C'_T value from the intercepted messages.

4.1.2 Resist forged-server attack

Our scheme was able to resist forged-server attacks as follows:

4.1.2.1 Purchase phase

The attacker can listen to iteration messages between a legitimate tag and a server. However, an attacker is able to supplant a legitimate server.
Step1: R $\rightarrow$ T: M_{req}, A
Step2: T $\rightarrow$ R: C_T, N_2, PID_i
Step3: R $\rightarrow$ S: $C_T, N_2, PID_i, A, RID_i$
Step4: S $\rightarrow$ R: C_S, C_1
Step5: R $\rightarrow$ T: C_S If an attacker intercepts the transmitted messages of $C_T, N_2, PID_i, A, RID_i, H(PW)$, and between the server and reader, the attacker can supplant the server without knowing all its private information ($EPC_i, Kill\_key_i Access\_key_i, DATA_i$) and the attacker can build a message as follows:

$$C'_S = CRC(EPC_i||Y'||Access\_key'_i) \oplus Kill\_key'_i \quad (35)$$

For the same reason, the server does not transmit Y and the access key ($Access\_key_i$) between the server and the reader, so the attacker cannot calculate the next correct communication parameter C'_S from the intercepted message.

4.1.2.2 Ownership transfer phase

When an attacker intercepts the iteration messages between the server and the reader, he/she is able to forge a legitimate server.
Step1: S $\rightarrow$ R: Y, C_2

The attacker obtains the transmitted message between the server and reader. He/she is able to compute the next correct message Y' to spoof the reader.

Suppose Y and Y' are legal messages. The different values of Y and Y' are calculated by the XOR operation, to obtain:

$$\begin{aligned} &Y \oplus Y' \\ &= (CRC(Access\_key_i||SN) \oplus Kill\_key_i) \\ &\oplus (CRC(Access\_key'_i||SN') \oplus Kill\_key'_i) \\ &= CRC(Access\_key_i \oplus Access\_key'_i)||(SN \oplus SN')) \\ &\oplus Kill\_key_i \oplus Kill\_key'_i \end{aligned} \quad (36)$$

Although, the message Y' is easily computed as follows:

$$\begin{aligned} &Y' = Y \oplus (CRC(Access\_key_i \oplus Access\_key'_i)||(SN||SN')) \\ &\oplus Kill\_key_i \oplus Kill\_key'_i \\ &= CRC(Access\_key_i||SN) \oplus Kill\_key_i) \oplus \\ &CRC(Access\_key_i \oplus Access\_key'_i)||(SN \oplus SN') \\ &\oplus Kill\_key_i \oplus Kill\_key'_i \\ &= (CRC(Access\_key'_i||SN') \oplus Kill\_key'_i) \end{aligned} \quad (37)$$

The server does not transmit the legal messages SN and $Access\_key_i$, which are involved into Y , to the reader; thus, the attacker cannot calculate the next correct communication parameter Y' from the intercepted messages.

4.1.3 User location privacy

Although the attacker cannot obtain the plain text from the tag, the attacker can still trace the user's location when tags respond to readers' queries with the same identifier.

The success of this attack depends on preventing tag-key updating. If the attacker intercepts the messages between the reader and the legitimate tag, he/she will be able to track the user's location for the following reason:
1st communication:
Step1: R $\rightarrow$ T: M_{req}, A
Step2: T $\rightarrow$ R: $C_T = CRC(EPC_i||B), N_2, PID_i$
n^{th} communication :
Step1: R $\rightarrow$ T: M_{req}, A
Step2: T$\rightarrow$ R: $C'_T = CRC(EPC_i||B'), N'_2, PID_i$
Now, the attacker intercepts $C_T$and C'_T computes the XOR of the messages:

$$C_T \oplus C'_T \quad (38)$$

$$= CRC(EPC_i||B) \oplus CRC(EPC_i||B') = CRC(B \oplus B') \quad (39)$$

If messages C_T and C'_T come from the same tag, the attacker can verify the transmitted messages from the same tag as follows:

$$\begin{aligned} &C_T = CRC(EPC_i||B) \\ &C'_T = CRC(EPC_i||B') \\ &Verify\ C'_T \cong C_T \end{aligned} \quad (40)$$

In our scheme, the tag does not transmit B between reader and tag and the pseudonym identification code of the tag PID_i is updated into a new one, $PID_{i_{new}}$, after

the transaction. Therefore, even though the attacker intercepts messages $C_T$and N_2 of that tag's response to a legal reader; the attacker cannot trace the user's location.

4.1.4 Resist replay attack

Each tag shares with the reader some private information: EPC_i, $Kill\_key_i$ and $Access\_key_i$. This information is used to build messages and in order to prove its authenticity.

If an attacker intercepts these messages(EPC_i, $Kill\_key_i$ and $Access\_key_i$) between the tag and the reader, the attacker can spoof the server by transmitting previously obtained C_T and A to pass the authentication. This scenario is described as follows:
The attacker intercepts the 1st communication message:
Step1: R $\rightarrow$ T: M_{req}, A
Step2: T $\rightarrow$ R: $C_T = CRC(EPC_i||B), N_2, PID_i$
The legitimate nth communication:
Step1: R $\rightarrow$ T: M_{req}, A
Step2: T$\rightarrow$ R: $C'_T = CRC(EPC_i||B'), N'_2, PID_i$
The attacker replays previously obtained C_T to pass authentication, but it fails. The reason is described as follows:

$$\begin{aligned} &C_T = CRC(EPC_i||B) \\ &C'_T = CRC(EPC_i||B') \\ &C'_T \neq C_T \end{aligned} \tag{41}$$

Since B are N_2 updated for each transaction and parameter B is not transmitted in plaintext, the attacker cannot spoof the server by transmitting the previous obtained C_T and A to pass the authentication.

4.1.5 Forward secrecy

In this section, we show that an attacker cannot compromise a tag and obtain its resident data; the attacker cannot obtain any secret tag information.

Suppose that an attacker listens to iteration messages (A, N_2, C_T, C_S) between a legitimate reader and a legitimate tag, and stores these values. Then, the tag will suffer from forward secrecy. Due to the EPC_i being obtained by the attacker, the attacker will be able to obtain the secret keys ($Kill\_key_i$ and $Access\_key_i$), and to generate correct communication C'_T . A detailed scenario of this attack can be described as follows:
Step1: R $\rightarrow$ T: M_{req}, A
Step2: T $\rightarrow$ R: C_T, N_2, PID_i
Step3: R $\rightarrow$ T: C_S
The attacker obtains the transmitted messages between the server and reader and computes the XOR operation of messages C_S and C'_S . So, message C'_S is easily computed.

However, in our scheme, the $kill\_key_i$ and $Access\_key_i$ are updated for each transaction, and the parameters $Y, kill\_key_i$ *and* $Access\_key_i$ are not transmitted in plaintext. Thus, if an attacker intercepts the messages between server and tag, the attacker cannot access the tag's secret data.

4.1.6 Resist man-in-the-middle attack

The proposed scheme was able to resist man-in-the-middle attacks. The reason is described as follows:

An attacker intercepts the communication messages between the tag and the reader. For example, an attacker pretends a legal role; when a reader wants to query a tag, the attacker intercepts the message from the reader and then transfers it to the tag as follows
Step1: R $\rightarrow$ T: M_{req}, A
Step2: T $\rightarrow$ R: C_T, N_2, PID_i
Step3: R $\rightarrow$ S: $C_T, N_2, PID_i, A, RID_i$
Step4: S $\rightarrow$ R: C_S, C_1
Step5: R $\rightarrow$ T: C_S

The tag and the server calculate C_S values by using EPC_i, Y *and keys*($Kill\_key_i$ *and* $Access\_key_i$) as follows:

$$C_S = CRC(EPC_i||Y||Access\_key_i) \oplus Kill\_key_i \tag{42}$$

If an attacker can hold and modify the messages, message C'_S is easily computed as follows:

$$C'_S = CRC(EPC_i||Y'||Access\_key'_i) \oplus Kill\_key'_i \tag{43}$$

In our scheme, the correct kill key($Kill\_key_i$) and access key ($Access\_key_i$) are protected by related parameters and updated for each transaction. Thus, attackers using a forged kill key ($Kill\_key_i$) and access key ($Access\_key_i$) to pass the tag's authentication will fail.

The reason is described as follows:

$$\begin{aligned} &C_S = CRC(EPC_i||Y||Access\_key_i) \oplus Kill\_key_i \\ &C'_S = CRC(EPC_i||Y'||Access\_key'_i) \oplus Kill\_key'_i \\ &C_S \neq C'_S \end{aligned} \tag{44}$$

The attacker cannot calculate the next correct communication parameter C'_S from the intercepted message to spoof the tag.

4.1.7 Ownership transfer

The proposed scheme can be applied to high-priced products for ownership transfer. In Step 2 of the ownership transfer phase, Consumer A (old owner) enters PW (chosen himself) at the purchase time and generates the hash value$H(PW)$. Likewise, Consumer B

(new owner) enters a new password PW_{new} and generates the hash value $H(PW_{new})$. The reader uses the session key to generate the OT as follows:

$$OT = E_{SK_i}(H(PW), H(PW_{new})) \quad (45)$$

Afterward, the reader transmits the message (OT, RID_i, PID_i) to the server. Upon receiving the message from the reader, the server uses the session key SK_i to encrypt message OT as follows:

$$(H(PW), H(PW_{new})) = D_{SK_i}(OT) \quad (46)$$

Then, the server checks whether $H(PW)$ of the PID_T in the database equals the received . The server regenerates SN' and transfers the ownership to Owner B. The authorized reader uses the access key ($Access\_key_i$) to write the new serial number SN' into the tag. Thus, our scheme achieves an ownership transfer. We compared the communication cost of the proposed scheme with those of previous ownership transfer schemes during the purchase phase, as shown in Table 1.

As shown in Table 2, the two relative transmission rates are 3.6 Mbps. Note that within the environment of 3.6 Mbps, the longest communication cost is required by the Purchase phase, the total data transmission time is only 0.029 ($6T_{XOR}$+$6T_{PRNG}$+ $6T_{CRC}$+ $1T_{SYD} + 1T_{SYE} + 1T_H$/(3600*8)) mini-seconds. Therefore, the transmission time of our proposed scheme is much faster.

Since the EPCglobal C1G2 standards only support exclusive-OR, random number generation, and CRC operations for tag operations, previous schemes [13, 14] have used a symmetric cryptosystem or one-way hash functions to implement their applications. These schemes did not conform to the EPC C1G2 standards; thus, they were not suitable for the current low-cost tags. Seo et al.'s operation [19] was more lightweight than our scheme. However, the reader and server operations were complex and their scheme used proxies to perform complex computations (as in asymmetric cryptosystems). Simultaneously, the proposed scheme was able to resist various attacks and utilized mutual authentication. None of the previous methods achieved all requirements, but the proposed method was able to do so. The security comparison is shown in Table 3.

Finally, we have compared our proposed scheme with related schemes which used smart tag mechanisms, as shown in Table 4.

5 Conclusions

At present, the cost of RFID tags remains high, despite the fact that much of the literature asserts that RFID tags should provide sufficient computational resources. Since current tags are limited by logic gates, such proposals are impractical.

Though many researchers have proposed schemes for RFID systems, few have conformed to EPCglobal C1G2 standards. However, the proposed scheme has been shown to have a provable RFID mutual-authentication scheme that conforms to EPCglobal C1G2 standards. Our scheme was able to resist Pedro et al.'s attack and enhance security. To sum up, our scheme was able to achieve the following:
(1) Resist forged-tag attack
(2) Resist forged-server attack
(3) User location privacy
(4) Resist replay attack
(5) Forward secrecy
(6) Resist man-in-the-middle attack
(7) Ownership transfer
For all products, a user can distinguish originals and fakes. Our scheme could be used in a lightweight RFID system that conformed to EPCglobal Class 1 Generation 2 standards. In the future, online verification should be integrated with the EPC network.

References

1. A. Juels, Strengthening EPC tags against cloning, Proceedings of the 4th ACM workshop on Wireless security, pp.67–76. (2005)
2. A.D. Henrici, P.Muller, Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers, In the Proceedings of PerSec'04 at IEEE PerCom, pp. 149–153, Mar. (2004).
3. 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, pp. 1284–1291, Dec. (2009)
4. C. L. Chen, Y. Y. Chen, and Y. H. Chen, Group-Based Authentication to Protect Digital Content for Business Applications, The International Journal of Innovative Computing, Information and Control, Vol. 5, No. 5, pp. 1243–1251, May. (2009)
5. C. L. Chen, An "All-In-One" Mobile DRM System Design, The International Journal of Innovative Computing, Information and Control, Vol.6, No.3A, pp.897–911, Mar. (2010)
6. C. L. Chen, Y. L. Lai, C. C. Chen, and Y. L. Chen, A Smart-card-based Mobile Secure Transaction System for Medical Treatment Examining Reports, The International Journal of Innovative Computing, Information and Control, In press.
7. D. Molnar, D. Wagner, Privacy and security in library RFID: issues, practices, and architectures, Conference on Computer and Communications Security (CCS'04), Washington, pp. 210–21, Oct. (2004)
8. D.N. Duc, J. Park, H. Lee, K. Kim, Enhancing security of EPCglobal GEN-2 RFID tag against traceability and cloning, The 2006 Symposium on Cryptography and Information Security, Japan, pp. 17–20, Jan. (2006)

Table 1 Comparison of the Communication cost

Communication cost				
Roles	Saito and Sakurai. [13]	Osaka et al. [14]	Seo et al. [19]	Our scheme
Tag	1 T_{SYE} + 1T_{SYD} (384 bits)	1T_H + 2 T_{XOR} (192 bits)	1T_{XOR} (16 bits)	3T_{XOR}+3T_{PRNG}+ 3T_{CRC} (144 bits)
Reader	Needn't (0 bits)	1T_{RNG} (16 bits)	1T_{ASYD} + 1T_{ASYE} (2048 bits)	1T_{SYD} + 1T_H (384 bits)
Server	NT_{XOR} (32 bits)	1T_{XOR} + 1T_{SYD} + 2T_{SYE} + NT_H (912 bits)	1T_{ASYD} + 1T_{ASYE} (2048 bits)	3T_{XOR}+3T_{PRNG}+ 3T_{CRC} + 1T_{SYE} (336 bits)
Total	416 bits	1120 bits	4112 bits	864 bits

N: number of the tags (if N = 2)
T_{XOR}: time for executing an exclusive-or operation (16 bits)
T_{PRNG}: time for executing a pseudo-random number generation operation (16 bits)
T_{CRC}: time for executing a Cyclic Redundancy Check (CRC) function (16 bits)
T_{SYE}: time for executing a symmetric encryption operation (192 bits)
T_{SYD}: time for executing a symmetric decryption operation (192 bits)
T_{ASYE}:time for executing an asymmetric encryption operation (1024 bits)
T_{ASYD}:time for executing an asymmetric decryption operation (1024 bits)
T_H:time of the hash function (160 bits)

Table 2 Comparison of the Transmission time

Transmission time (ms)(3.6 Mbps)				
Roles	Saito and Sakurai. [13]	Osaka et al. [14]	Seo et al. [19]	Our scheme
Tag	0.013	0.006	0	0.005
Reader	0	0	0.071	0.013
Server	0.001	0.038	0.071	0.011
Total	0.014	0.044	0.142	0.029

9. D. Molnar, A. Soppera and D.Wagner, A scalable, delegatable pseudonym protocol enabling ownership transfer of RFID tags, selected areas in cryptography, LNCS 3897, pp. 276–290. (2006)
10. E. Y. Choi, D. H. Lee b,J. I. Limb, Anti-cloning protocol suitable to EPCglobal Class-1 Generation-2 RFID systems, Computer Standards & Interfaces, pp. 1124–1130, Nov. (2009)
11. EPCglobal web site: http://www.epcglobalinc.org/
12. H. Y. Chien, C. H. Chen, Mutual authentication protocol for RFID conforming to EPC class 1 generation 2 standards, Computer Standards & Interfaces, Vol. 29, No. 2, pp. 254–259, Feb. (2007)
13. J. Saito and K. Sakurai, Owner transferable privacy protection scheme for RFID tags. Proc. of Computer Security Symposium 2005 (CSS2005), Japan, pp. 283–288, Oct. (2005)
14. 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, Japan, Vol. 2, pp. 1090-1095. Nov. (2006)
15. P. L. Pedro, H. C. Julio Cesar, E. T. Juan M, R.Arturo, Cryptanalysis of a novel authentication protocol conforming to EPC-C1G2 standard, Computer Standards & Interfaces, Vol. 31, No. 2, pp. 372–380. (2009)
16. S. Karthikeyan, M. Nesterenko, RFID security without extensive cryptography, Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks, 2005, Virginia, Nov.7, pp. 63–67. (2005)
17. S. L. Garfinkel, A. Juels, R. Pappu, RFID Privacy: An overview of problems and proposed Solutions, IEEE Security & Privacy Magazine, vol. 3, no. 3, pp. 34–43, May. (2005)

Table 3 Security comparison

	Schemes			
Attacks	Saito and Sakurai. [13]	Osaka et al. [14]	Seo et al. [19]	Our scheme
Resist forged-tag attack	No	No	No	Yes
Resist forged-server attack	No	No	No	Yes
User location privacy	Yes	Yes	Yes	Yes
Resist replay attack	No	No	No	Yes
Forward secrecy	No	No	No	Yes
Resist man-in-the-middle attack	No	No	No	Yes
Ownership transfer	Yes	Yes	Yes	Yes

Table 4 Mechanism comparison

	Schemes			
Mechanism	Saito and Sakurai. [13]	Osaka et al. [14]	Seo et al. [19]	Our scheme
Mutual authentication	No	No	No	Yes
Protected transmission	No	Yes	Yes	No
Authority management	Yes	Yes	Yes	Yes
Encryption method	symmetric operation	hash+symmetric operation	hash+symmetric operation	CRC
Tag can be used many times	Yes	Yes	Yes	No
Conforms to EPC C1G2	No	No	Yes	Yes

18. S. A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engels. Security and privacy aspects of low-cost radio frequency identification systems, security in pervasive computing, LNCS 2802, pp. 201–212. (2004
19. Y. Seo, T. Asano, H. Lee, and K. Kim, A Lightweight Protocol Enabling Ownership Transfer and Granular Data Access of RFID Tags, the 2007 Symposium on Cryptography and Information Security Sasebo, Japan, pp. 23–26, Jan. (2007)