Computer Networks 105 (2016) 47–59

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

RFID ownership transfer protocol based on cloud

Tianjie Cao[a,1], Xiuqing Chen[b,1,*], Robin Doss[c], Jingxuan Zhai[a], Lucas J Wise[d], Qiang Zhao[b]

[a] School of Computer Science and Technology, China University of mining and technology, Xuzhou 221116, China
[b] School of Medicine Information, Xuzhou Medical University, Xuzhou 221000, China
[c] School of Information Technology, Deakin University, Burwood, VIC 3125, Australia
[d] School of Mines, China University of mining and technology, Xuzhou 221116, China

ARTICLE INFO

Article history:
Received 27 April 2015
Revised 23 May 2016
Accepted 23 May 2016
Available online 24 May 2016

Keywords:
RFID
Internet of Things
Backward untraceability
Strong forward untraceability

ABSTRACT

RFID and Cloud computing are widely used in the IoT (Internet of Things). However, there are few research works which combine RFID ownership transfer schemes with Cloud computing. Subsequently, this paper points out the weaknesses in two protocols proposed by Xie *et al.* (2013) [3] and Doss *et al.* (2013) [9]. To solve the security issues of these protocols, we present a provably secure RFID ownership transfer protocol which achieves the security and privacy requirements for cloud-based applications. To be more specific, the communication channels among the tags, mobile readers and the cloud database are insecure. Besides, an encrypted hash table is used in the cloud database. Next, the presented protocol not only meets backward untraceability and the proposed strong forward untraceability, but also resists against replay attacks, tracing attacks, inner reader malicious impersonation attacks, tag impersonation attacks and desynchronization attacks. The comparisons of security and performance properties show that the proposed protocol has more security, higher efficiency and better scalability compared with other schemes.

1. Introduction

Cloud computing and RFID (Radio frequency identification) technologies are increasingly influencing the applications of IoT (Internet of Things). A typical RFID system will include numerous mobile RFID readers and potentially support several thousands of low cost RFID tags. The message processing and key storages can be read and updated by the mobile readers, which are normally supported by the cloud and not by the backend server. In this way, the mobile user can offer pervasive RFID service securely via the internet whenever and wherever. RFID technologies are emerging in various domains, such as animal authentication, asset tracking, supply chain management, highway toll collection, intelligent building, smart electric home appliances, intelligent transport systems and surveillance systems. An important aspect of RFID security is secure ownership transfer of the RFID tag as the tagged product changes control over the distribution chain. The OT (Ownership transfer) requires that control (i.e., communication capabilities) of a RFID tag is transferred from the current owner to the new owner. Secure ownership transfer requires, at a minimum, the establishment of new shared secrets between the tag and the new owner. Widespread RFID technologies adoption has drawn close attention to massive challenges that are critical and should be overcome before the explosion of security and privacy attacks.

RFID OT protocols have achieved widespread success in ubiquitous computing and Cloud computing owing to low cost, automatic authentication and broad applicabilities. However, the security and privacy of current OT protocols deployed in RFID cloud based applications is still not guaranteed. The main reason for this being that RFID OT protocols have not been developed for cloud-based environments. Therefore, it is essential that RFID clouds should provide provably secure and private ownership transfer.

The readers communicates with the tags and cloud-database through the wireless channels, the security of which cannot be guaranteed. While the communications among the tags, readers and the cloud sever are assumed to be insecure. Recently, the security and privacy vulnerabilities of RFID schemes based on cloud sever [1–3] have attracted many attentions. In addition, the protocols [4–7] use various methods to realize ownership transfer and authentication, but they do not use the cloud database. The security, privacy and cost in RFID schemes are the main factors that slow down the rapid and widespread deployment of the RFID technology in cloud computing. To reduce the tag cost, resist these security attacks, protect owners privacy, and prevent unauthorized communication among tags, readers and cloud database, the RFID protocol using quadratic residue cryptography is a good choice. Mean while, because of the limited capacity of tags in computation and storage, many proposed scheme based on quadratic residue

* Corresponding author.
E-mail address: tb12170002@cumt.edu.cn, xiuqingchen@126.com (X. Chen).
[1] These authors contributed equally to this work.

http://dx.doi.org/10.1016/j.comnet.2016.05.017
1389-1286/

cryptography and cloud database can be directly used in large-scale RFID applications.

Consequently, the requirements of OT protocols providing cloud storage, efficiency, security, and untraceable analysis with low cost have become significant for the future applications of RFID OT schemes. One of the most important attempts to fulfill the requirements is the OT protocol based on quadratic residue mechanisms, cloud database and security model. The most resent protocols are proposed by Xie et al. [3] and Doss et al. [9] which adopt cloud database and quadratic residue, respectively. However, some of them may not achieve the limited capacity of tags and the others may lack untraceable analysis.

The untraceability properties [8] are important privacy requirements for the OT protocols. Alagheband et al. [8] used untraceability analysis for the RFID protocols. However, many OT protocols lack untraceability analysis and are vulnerable to various attacks. In order to limit the vulnerabilities and make the OT protocol secure and anonymous, it is necessary to analyze the untraceability of OT protocols.

Moreover, we also address the untraceability issues on the two schemes [3,9], and then provide solutions for the security and privacy issues. Despite these disadvantages, the security and the privacy of the entities (tag owners, mobile reader holders, and pervasive cloud databases) are the main concerns in the rapid and widespread applications of the Cloud computing technology. Data storage and processing are moved from the server to a cloud database, which improves the security and efficiency in the scheme. This paper proposes an ownership transfer protocol based on quadratic residues which can meet security co-existence requirements of cloud databases and RFID ownership transfer systems.

The main contributions of this paper include:

(1) Outlining the disadvantages in the two schemes proposed by Xie *et al.* [3] and Doss *et al.* [9].
(2) When Cloud computing is applied to RFID ownership transfer scenarios, the superior CROP (cloud-based RFID OT protocol) is proposed. It inherits pay-on-demand resource deployment, great scalability and pervasive accessibility from Cloud computing, without lacking security and privacy protection.
(3) The most important part of the paper is the first comprehensive, untraceable analysis of RFID ownership transfer protocol. The scheme achieves untraceability privacy properties and is resistant to eavesdropping, manipulation, replay and desynchronization attacks. The performance properties of the proposed protocol are scalable with computational complexity O(1) and are superior to other protocols.

The rest of the paper is organized as follows. Related works are reviewed in Section 2. Section 3 introduces some notations, security requirements of cloud-based RFID and security models are introduced. Section 4 points out the weaknesses of Xie et al.'s and Doss et al.'s protocols. The cloud-based RFID ownership transfer protocol is presented in Section 5. Section 6 formally proves the improved protocol and investigates its security in detail. Finally, in Section 7, the conclusions are summarized.

2. Related works

The schemes based on cloud database [1–3] reduces the computation load for low-cost tags and shortens the overall authentication time. The OT schemes without cloud database [4–7] use various methods to realize OT and low-cost. Nevertheless, most of these approaches have security issues and lack untraceability analysis.

For example, Jiang et al. [1] proposed a data storage framework in Cloud computing platform not only collecting IoT data by sensors and RFID readers, but also enabling efficient storing of structured and unstructured data. The schemes [2,3] introduce the cloud database to the RFID system and make full use of the advantages of cloud technology to improve the performance of the RFID system. The security and privacy cloud-based scheme [3] is highly dependent on VPN (Virtual Private Network) and EHT (Encrypted Hash Table) in the Cloud computing service. These schemes also provide the important guidelines for OT protocol design by adding the cloud database. For instance, the OT protocol [5] presented by Chen and Chien is conforming to the EPC Class-1 Generation-2 (EPCC1G2) standards which strike a balance between low-cost and functionality, with less security issues. The new proposed protocol [6], based on sliding window mechanisms, resists against desynchronization attacks.

Recently, Zhang et al. [10] solved the tag search problem and proposed the ITSP (iterative tag search protocol). Besides the readers request the Cloud computing to improve computation capacity in a large-scale RFID system. Khan et al. [11] proposed a novel cryptographic authentication protocol that resisted Denial-of-Service attacks and provided significantly lower cost. Li et al. [12] pointed out that Srivastava et al.'s protocol [13] suffered from privacy damage in that an attacker may connect to the medical DB server that store medical information associated with tagged objects from stolen/lost readers (the malicious inner reader). Therefore, they presented a secure authentication protocol to resist the malicious inner reader attacks and provided the higher system efficiency compared with Srivastava et al.'s protocol.

In addition, there are privacy issues in the cloud applications and RFID ownership transfer protocols. In the cloud-based authentication schemes, wireless open internet connections among the tags, mobile readers and the public cloud are insecure, since a cloud provider is not trustworthy. The current scheme [3] lays emphasis on the authentication protocol based on Cloud computing. Recently, the security and privacy problems have been solved by EHT in the Cloud computing service of cloud-based schemes. However, this paper finds that the cloud-based RFID authentication schemes are subjected to secret parameter disclosure attacks. Within the aspect of data storing and accessing in ownership transfer protocols, the ownership transfer scheme faces a series of challenges, such as the rapid data generation, complicated requirements of data management and others.

The protocol based on quadratic residue [14] pointed out that the scheme [15] was subjected to tag impersonation attacks, desynchronization attacks and tracing attacks. Because of the vulnerabilities in the data structure of the tag in [17], the tag's outputs (X, T) in [16] and (x'', t'') in [17] are similar to the tag's structure in the scheme [15], the protocols [16,17] are subjected to the same attacks, such as tag impersonation attacks and tracing attacks. In addition to the above attacks, it is necessary to analyze inner reader malicious impersonation attacks in the OT protocols.

Doss *et al.* presented the quadratic residue property in two schemes [9,18], and claimed that their scheme [9] achieved strong security and privacy properties, such as resistance to replay attacks, resistance to desynchronization attacks, resistance to DB impersonation attacks, forward secrecy, and forward untraceability. However, the schemes lack comprehensive untraceability analysis.

In the current versions of the ownership transfer schemes, there are no types of cloud-based OT schemes in practical cloud applications, because they lack comprehensive security and privacy considerations. In other words, the reader holders and tag owners need to achieve requirements of access anonymity and data privacy in cloud-based RFID ownership transfer protocols. For solving these problems, this paper develops the suitable ownership transfer schemes in Cloud computing setting.

Table 1
Notations and descriptions.

Notations	Descriptions
$R_i(R_{i+1})$	The current mobile reader R_i (The new mobile reader R_{i+1}).
DB, k	The database or sever, security parameter.
Adv, Adv$^-$, *Adv*$^+$	The adversary, weak adversary, narrow-strong adversary.
A^B	The probabilistic polynomial time adversary,$A^B \in$ (*Adv, Adv*$^-$, *Adv*$^+$).
Pr	The probability when the experiment $\mathrm{Exp}_{P,A^B}^{c-\mathrm{Untra}}(k)$ tends to 1.
$T_i(T_{i+1})$	The current tag T_i (The new mobile reader T_{i+1}).
$r_i(r_{i+1})$	The random numbers created by the tag T_i (T_{i+1}).
TID, T	The tag identifier.
h (*TID*)	The hash value of the identifier(*TID*).
R_{TID}, *RID, R*	The identifier of the reader, $R_{TID} = h(TID) \oplus r$.
SID, S(*La*)	The numbers authentication sessions between a reader and a tag, (*La*: the last number of sessions).
Data	A session about the user, the reader, and the tag.
K_{TID}	A key $K_{TID} = v_1 \| v_2 \| v_3 \| \ldots \| v_m$ shared by the tag and DB.
v_p, v_{p+l}	The *p*th random number from K_{TID}, where $l = \|p - m\|$.
n_s, n_T, n_{T2}	Three different nonces of the same tag.
n, p, q	A positive integer stored in R_i, two large prime numbers *p* and *q* ($n = pq$).
n′, *p*′, *q*′	A positive integer stored in R_{i+1}, two large prime numbers *p*′ and *q*′ ($n' = p'q'$).
N_T, *NT*(N_R, *NR*)	The random number generated by tag (reader).
H( )	Hash function with output length *L*. H(): $(0, 1)* \rightarrow (0, 1)^L$.
^{i}*Mes*, Mes_{Adv}	The *i*th session of message *Mes*, the attacker creates Mes_{Adv}.
Rkey, Tkey	All the keys of the reader (tag).
Sec_1, Sec_2, Sec_3	The sessions between reader and tag.
EHT	The encrypted hash table contains the index *I* and contents *K* and *M*.
I_i, I_{i+1}	The index for the reader R_i and R_{i+1}.
K_i, M_i, K_{i+1},K_{i+1}	The contents for the reader R_i and R_{i+1}.
r_i, r_{i+1},r_{i+2}	The shared keys for R_i, R_{i+1} and R_{i+2},respectively.
K_{TIDc}, K_{TIDn}	Two shared keys, K_{TIDc} for R_i and K_{TIDn} for R_{i+1}.
$v_{p'}$	The *p*′th random numbers are drawn from K_{TIDc} ($l = \|p - m\|$).
t_i	The Mersenne prime t_i stored by R_i, is used to create $n = 2^{t_i} - 1$.
t_{i+1}	The Mersenne prime t_{i+1} stored by R_{i+1}, is used to create $n' = 2^{t_{i+1}} - 1$.

3. The security requirements and security model for the cloud-based OT protocol

The notations and descriptions of all schemes are shown in Table 1.

3.1. The security requirements of the cloud-based RFID ownership transfer system

There are two assumptions in the cloud-based RFID OT scheme. On the one hand, the cloud database is insecure [22,23], the attackers cannot decode the keys from the stealing the information of the corrupted cloud database. On the other hand, the communications among the tags, the readers and the cloud sever are usually through the wireless channel, which are assumed to be insecure. The attackers cannot decode the transmitted messages from the monitoring information of the communication. The two security requirements based on the assumptions are as follows.

(1) Data privacy of the Cloud-storage service
The cloud DB provides the data privacy required to offer data storage service by encrypting the keys of two entities (tag and reader).
(2) Access anonymity of the inquiry service
The cloud DB offers anonymity of access and data inquiry service by encrypting reader's/tag's transmitted messages.

The data cloud-storage and inquiry services are secure, since the keys and transmitted information of the two entities are encrypted.

3.2. Security model

The hypotheses of the model are that the channels of tag-reader, reader-DB and tag–tag are insecure. The security and privacy properties are proved by using the untraceability definitions in [8] and the oracles in Vaudenay model [19] for the RFID protocols. Then we enhance the unreasonable assumption of forward untraceability (*Adv*$^+$ misses the $(i+1)$th session), since *Adv*$^+$ can continuously monitor the tags' outputs at the application environment. Specifically we present the definition of strong forward untraceability for the ownership transfer protocols.

Definition 1 ((Forward untraceability) [20])**.** The narrow-strong adversary cannot trace the tag at the round i' that $i' \geq i+2$, even though *Adv*$^+$ corrupts the target tag in the *i*th session and misses the $(i+1)$th session.

Definition 2 ((Backward untraceability) [20])**.** Even if *Adv*$^+$ corrupts the *i*th keys of the target tag, she/he cannot trace the target tag's transactions and keys in the i' session $i' < i$-1.

Definition 3 (Strong forward untraceability)**.** It is impossible for *Adv*$^+$ to trace the tag in the i' session $i' \geq i$ +1, even though *Adv*$^+$ corrupts the tag's keys in the *i*th session.

The security model use the following oracles.

Init(1^k) activates the tag and reader to output the new session;

Send(m) allows the attacker to send arbitrary messages to the readers and tags;

Corrupt(t) responds tag's secret key.

More specifically, we add the **Compute** and **Compare** oracles to complete the model.

Compute(m) allows the adversary to calculate the target tag's outputs in any session using the corrupted keys and the known encryption structure.

Compare(m) allows the attacker to compare the calculated values with the monitored messages. Finally, the attacker outputs a bit d (if $d = 1$, the attack succeeds; else $d = 0$, the attack fails).

The adversary can only issue the corrupt query to t^i in the current session. The definition of Untraceable Privacy (Upri) [21], which is essential for private ownership transfer of RFID tags.

The adversary A^B controls the communications between all protocol parties (tag and reader) by interacting with them as defined by the protocol, formally captured by A^B's ability to issue queries of the following form:

Definition 4 (Untraceable Privacy+ (Upri+))**.** Upri+ is defined using the game **G** played between a malicious adversary A^B and a collection of reader and tag instances. A^B runs the game **G** whose setting is as follows.

Phase 1 (Intialization): A^B can send any queries (**Execute, Send, Corrupt**).

Phase 2 (Learning): A^B can send any queries (**Execute, Send, Corrupt**).

Phase 3 (Challenge): A^B can send any queries (**Execute, Send, Corrupt**).

1. At some point during ***G***, A^B will choose a fresh session on which to be tested. Depending on a randomly chosen bit $d \in (0,1)$, A^B compares that t^* is from the session $(j-i-1, j-i+1, j-i+2)$.

2. A^B continues making any queries (**Execute, Send, Corrupt**) at will.

Phase 4 (Guessing): Eventually, A^B terminates the game simulation and outputs a bit d by using **Compute, Compare** queries, as its guess of the value of d. The success of A^B in winning $\boldsymbol{G}$ and thus breaking the notion of Upri+ is quantified in terms of A^B's advantage in distinguishing whether t^* is from t^i.

i.e. it correctly guessing d. This is denoted by $Adv_{A^B}^{\text{UPri+}}(k)$ where k is the security parameter. We consider the following games for the above privacy properties:

$\text{Exp}_{A^B}^{c-\text{Untra}}(k)$

Phase 1 (Initialization):
$\text{Setup}(1^k) \rightarrow (Rkey^j, Tkey^i)$;
Phase 2 (Learning):
$A_1^{\text{Init,SendTag,SendReader,Execute,Corrupt}}(R, T, Tkey^i) \rightarrow (R^j, t^i, Sec_1, Tkey^i)$;
Phase 3 (Challenge):
$A_2^{\text{Init,SendTag,SendReader}}(R^{\#}, t^*, Sec_1) \rightarrow (R^{\#}, t^*, Sec_2)$;
Phase 4 (Guessing):
$A_3^{\text{Compute}}(Sec_1, Sec_2, Tkey^i) \rightarrow (R^{\#}, t^*, Sec_3)$;
$A_4^{\text{Compare}}(Sec_2, Sec_3) \rightarrow (R^{\#}, t^*, d)$;
//(under Narrow-strong attacker model)
If $c = j - i + 1$ (Backward untraceability)
$\text{B-untra}(R^j, t^{i-1}, A_1, A_2, A_3, A_4) \rightarrow d$;
If $c = j - i - 1$ (Strong forward untraceability)
$\text{S-untra}(R^j, t^{i+1}, A_1, A_2, A_3, A_4) \rightarrow d$;
If $c = j - i + 2$(Forward untraceability under missing the $(i+1)^{\text{th}}$ section)
$\text{F-untra}(R^{j+2}, t^i, A_1, A_2, A_3, A_4) \rightarrow d$;
output d

Different from other privacy games, the adversary cannot obtain any of the owner's secret information. The advantages of the adversary against the above games are defined by

$Adv_{A^B}^{\text{B-Untra}}(k) = \Pr[\text{Exp}_{A^B}^{\text{B-Untra}}(k) \rightarrow 1]$,
$Adv_{A^B}^{\text{S-Untra}}(k) = \Pr[\text{Exp}_{A^B}^{\text{S-Untra}}(k) \rightarrow 1]$,
and $Adv_{A^B}^{\text{F-Untra}}(k) = \Pr[\text{Exp}_{A^B}^{\text{F-Untra}}(k) \rightarrow 1]$.

Consider that the attacker executes the ownership transfer protocol and outputs d which indicates whether the protocol is untraceable or not. If the three advantages become zero, the ownership transfer protocol meets the privacy properties.

4. The weaknesses of two schemes

RFID ownership transfer schemes presented by Doss et al. (2013) suffer from desynchronization attacks, tracing attacks and inner reader malicious impersonation attacks. In addition, the original cloud-based RFID authentication protocol proposed by Xie et al. (2013) is subject to key disclosure attacks.

4.1. The weaknesses of doss et al.'s protocol

4.1.1. Cryptographic analysis of an ownership transfer scheme in an open loop system

Figs. 1 and 2 describe the ownership transfer schemes in an open loop RFID system and in a closed loop RFID system, respectively. The attacks are introduced in these schemes in this section. Due to the page limitation, the detailed steps of the original schemes are omitted.

A. Inner legitimate reader impersonation attacks and tracing attacks

Definition 5 (Inner legitimate reader impersonation attacks)**.** There are two cases under the weak attacker as follows: If the old readers are legal in RFID system and their subjective requirements are malicious, they can use the known messages and compute the old readers' outputs which are the same as the new readers'. Therefore, the old reader can impersonate the new reader and achieves ownership transfer without the participation of the new reader. Similarly, the impersonation attacks of malicious new readers are the same as the above process.

For instance, the new owner transmits the tag's key n' in plaintext to the old owner who can calculate the same value H. In other words, the old reader can trace the new tag using the updated key n' in the next session. In addition, when the old owner can generate the same values (N, H) which are verified successfully, the old owner impersonates the new owner to achieve ownership transfer.

Then the tag receives the fake data (N, H) and verifies whether the received messages are legal, after verifying that the replayed messages are legal, the tag proceeds to compute the next step. Therefore, the malicious old owner impersonates the new reader successfully.

B. Outer illegitimate reader impersonation attacks

Definition 6 (Outer illegitimate reader impersonation attacks)**.** Outer illegitimate reader modifies the legitimate reader's outputs which can be verified by the tag under the weak attacker.

For example, the steps of the attack are as follows:

S1. Adv^- monitors a normal run of the ownership transfer scheme in an open loop RFID system from step 1 to step 5 in Fig. 1. The tag which is undergoing ownership transfer, is only within the communication range of the potential new owner.

S3. Adv^- blocks the messages (N, H) in step 5.1 and modifies the transferred messages as follows:

a. $N_{Adv} = N \oplus n_s$
b. $H_{Adv} = H \oplus n_s = h(TID) \oplus v_{p+l} \oplus N \oplus PRNG(v_{p+l} \oplus n_s) \oplus r$

Subsequently, Adv^- forwards the messages (N_{Adv}, H_{Adv}) to the tag T_i.

S3. The tag T_i receives (N_{Adv}, H_{Adv}), extracts $n' = N_{Adv} \oplus PRNG(v_{p+l} \oplus n_s) \oplus r$ using N_{Adv} and computes H_T as follows:

$$\begin{aligned} H_T &= h(TID) \oplus v_{p+l} \oplus n_s \oplus n' \\ &= h(TID) \oplus v_{p+l} \oplus n_s \oplus N_{Adv} \oplus \\ &\quad PRNG(v_{p+l} \oplus n_s) \oplus r \\ &= h(TID) \oplus v_{p+l} \oplus N \oplus \\ &\quad PRNG(v_{p+l} \oplus n_s) \oplus r \end{aligned}$$

The tag compares the computed H_T with the received H_{Adv}. If the two values are equal to each other, the spoofed reader is verified by the tag.

C. Tag impersonation attacks

The attacker modifies the tag's output x'' which can be verified by the reader, as there is no validation message containing the message x'' in reader's verification process. Therefore, the scheme suffers from tag impersonation attacks.

D. Desynchronization attacks

After the attacker modifies the reader's outputs, the modified messages are verified by the tag, which leads to desynchronization attacks. The reason is that the reader and tag use different parameters to update the tag's keys, respectively.

S1. Adv^- monitors a normal run of the ownership transfer scheme in an open loop RFID system from step1 to step 7 in Fig. 1. The tag undergoing ownership transfer is only within the communication range of the potential new owner. Then the new owner R_{i+1} selects a new key K'_{TID}.

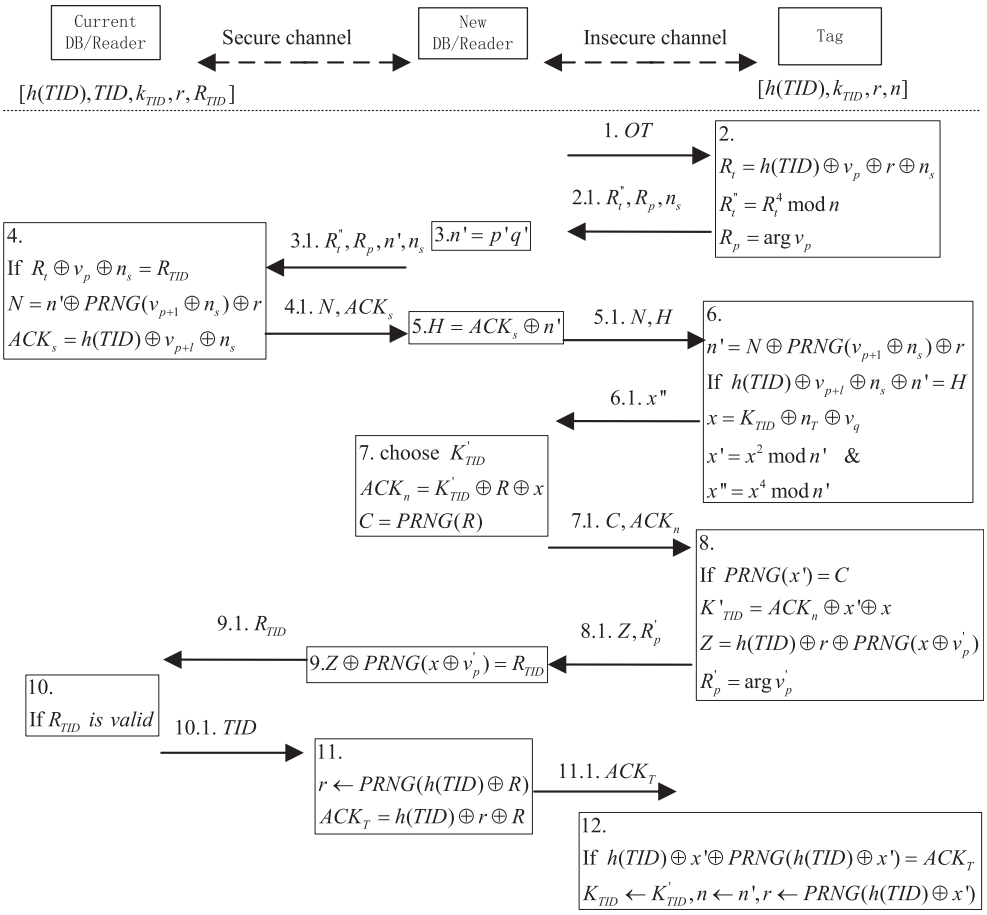

Fig. 1. The ownership transfer scheme in an open loop RFID system.

- **S2.** Adv^{-} blocks the messages (ACK_n, C) of the step 7.1 of Fig. 1, and modifies the transferred message ACK_n as $ACK_{nAdv} = ACK_n \oplus \Delta$ (Δ is a random number). Subsequently, Adv^{-} forwards the messages (ACK_{nAdv}, C) to the tag T_i.
- **S3.** The tag T_i receives (ACK_{nAdv}, C) and compares the computed $PRNG$ (x') using its key x' with the received C. If $PRNG(x')$ is equal to C, the tag updates the key using the received ACK_{nAdv} as follow: $K'_{TID} = ACK_{nAdv} \oplus x \oplus x' = ACK_n \oplus \Delta \oplus x \oplus x'$.
- **S4.** The remaining sessions in the protocol are implemented. At last, the desynchronization attack occurs, since T_{i+1} and R_{i+1} update the key K'_{TID} using different parameters.

Furthermore, if the old owner knows the updated secret key n', she/he can trace the new tag. Therefore, forward untraceable property of the tag is destroyed. The solution is that the new owner passes the encrypted n' rather than the plain text n' to the old owner. As the malicious reader is not able to resolve the prime number, its counterfeit goal cannot be achievable in the whole process.

Since the old owner passes TID to the new owner in the step 9.1, based on which the new owner can trace the historical information of the tag T_i. It severely threatens the backward untraceability and data privacy of the database.

4.1.2. Cryptographic analysis of ownership transfer scheme in a closed loop RFID system

Similar to the attacks on the open loop RFID system, the scheme in a closed loop RFID system in Fig. 2 suffers from outer reader impersonation attacks and the desynchronization attacks. We describe the attack processes without repeating the same steps for brevity purposes.

A. The outer reader impersonation attack

This attack in a closed loop RFID system is similar to the attack in Section 3.1.1. If the attacker monitors and blocks the messages (ACK_c, N) in step 4.1, then she/he can replay ACK_c and modify the transferred message N as $N_{Adv} = N \oplus \Delta$. Next, the modified contents are validated by the tag, the outer reader impersonation attack succeeds.

B. The desynchronization attack

In order to succeed in the validation process of the tag, the attacker replays the message C and randomly modifies the message ACK_n in step 6.1. Then the tag's key K_{TIDc} is updated by the unauthorized user, which leads to the desynchronization attack, because the attacker can optionally modify the message ACK_n as $ACK_{nAdv} = ACK_n \oplus \Delta$.

In a word, the protocol designers know about the above attacks which continue to be successful, since the protocol does not check data integrity for each transmitted message. Therefore, it is imperative that RFID protocols have a safe effective mechanism for handling ownership transfer. Based on this discussion, it is easy to attack the ownership transfer scheme. The attacker can modify the messages (N, H, ACK_n) in an open loop scheme and (N, ACK_n) in a closed loop scheme, since the schemes lack the data integrity for the transmitted messages.

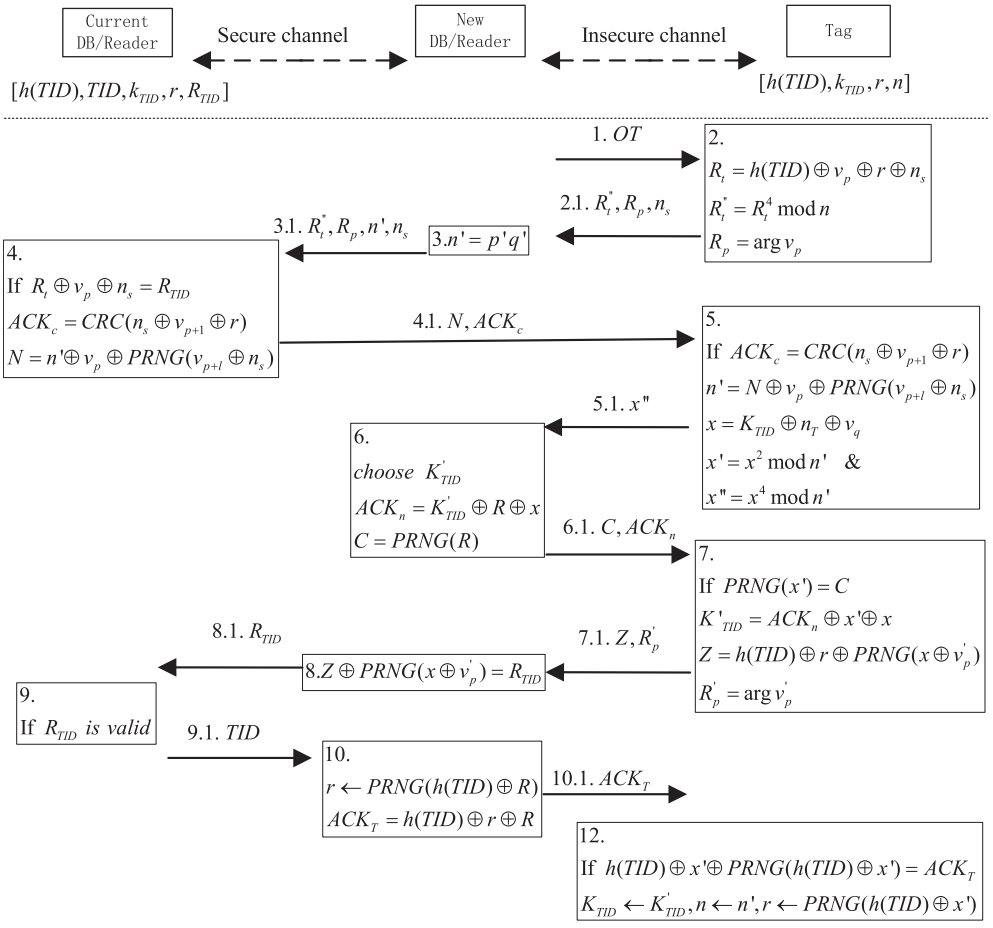

Fig. 2. The ownership transfer scheme in a closed loop RFID system.

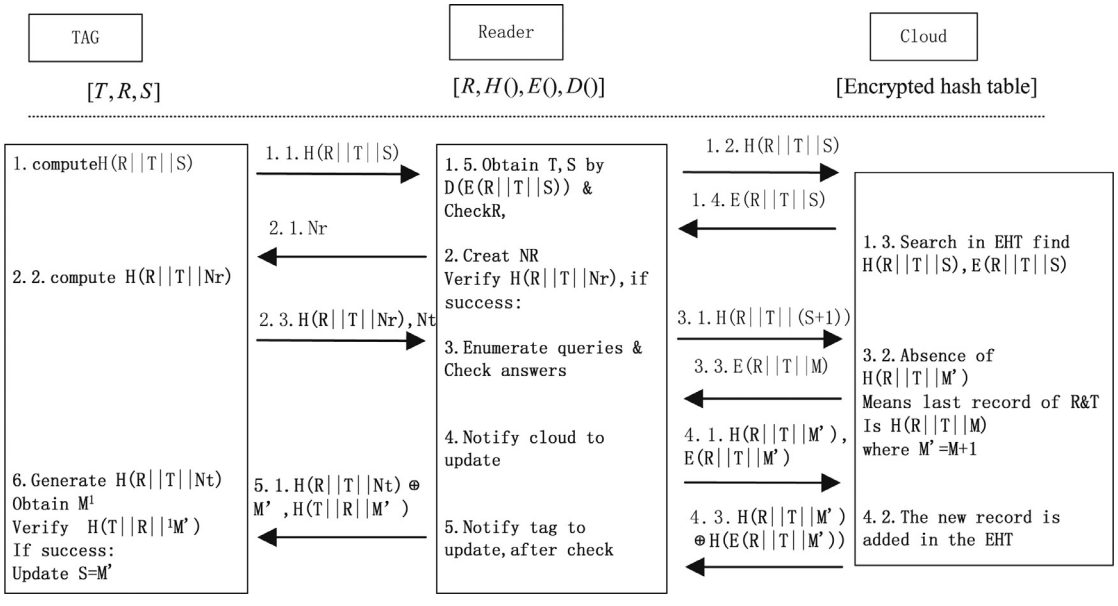

Fig. 3. Cloud-based RFID authentication scheme.

4.2. *The weaknesses of Xie* et al.*'s protocol*

To avoid repetition, the details in steps which are similar to the original scheme in Fig. 3 are omitted. The designers claim that their protocol is in optimal security. However, we present a passive attack which can retrieve the $(i+1)$th key SID in the ith session by eavesdropping. The complexity of this attack is eavesdropping only one session between the tag and a legitimate reader. In addition, we show that an active adversary can retrieve secret parameters more efficiently with the complexity of two sessions. The success probability of the given attack is "1". In other words, the scheme does not meet backward untraceability and forward untraceability.

The above protocol is subjected to tracing attacks, replay attacks, tag impersonation attacks, outer reader impersonation attacks and key disclosure attacks. The steps of the attacks are described as below.

Phase 1 (Learning 1):

S1.1. Adv^{-} chooses the target tag T_i, monitors one run of the cloud-based RFID authentication protocol from step1 to step 5, and obtains the messages (H(R‖T‖S), ^{1}Nr, H(R‖T‖^{1}Nr), ^{1}Nt, H(R‖T‖^{1}Nt) $\oplus^1$M′, H(T‖R‖^{1}M′)) between tag and reader.

S1.2. Adv^{-} blocks **step** 5.1 of Fig. 3 from the reader to T_i and prevents the tag from updating its keys.

Phase 2 (Challenge 1):

S2.1. Adv^{-} monitors the tag T^*, when T^* sends the same request H(R‖T‖S) to the reader, then she/he decides that T^* is T_i. Therefore, Adv^{-} can trace the tag T_i using the same request H(R‖T‖S).

S2.2. When the tag replays H(R‖T‖S) to the reader, the reader verifies successfully using the received message H(R‖T‖S), and then sends ^{2}Nr to T_i. Therefore, the protocol is subjected to the replay attacks.

S2.3. Adv^{-} continues to transmit ^{1}Nt instead of ^{2}Nr to T^*.

S2.4. The tag sends the computed message H(R‖T‖^{1}Nt) and the challenge ^{2}Nt to the reader. The scheme is subjected to outer reader impersonation attacks.

S2.5. Adv^{-} blocks the messages (H(R‖T‖^{1}Nt), ^{2}Nt), terminates the scheme. Then, Adv^{-} computes the old key ^{1}M′= H(R‖T‖^{1}Nt) $\oplus^1$M′$\oplus$H(R‖T‖^1 Nt) using the monitored H(R‖T‖^{1}Nt) $\oplus^1$M′. In other words, Adv^{-} can calculate the ith key S(^{1}M′) of the tag in the $(i+1)$th session. Therefore, the scheme is not backward untraceable.

The protocol does not meet forward untraceability and is subjected to tag impersonation attacks and key disclosure attacks. The steps of the attacks are illustrated as below.

Phase 1 (Learning 2):
The Learning 1 phase is the same as the Learning 2.

Phase 2 (Challenge 2):
The **Step** 2.1 and **Step** 2.2 are the same as the Challenge 1.

S2.3. Adv^{-} monitors (H(R‖T‖^{2}Nr),^{2}Nt). Then, she/he transmits (H(R‖T‖^{2}Nr),^{2}Nr) instead of (H(R‖T‖^{2}Nr),^{2}Nt) to T_i.

S2.4. The reader verifies the modified messages (H(R‖T‖^{2}Nr), ^{2}Nr) as the legal data, and outputs (H(R‖T‖^{2}Nr) $\oplus^2$M′, H(T‖R‖^{2}M′)). Therefore, the tag impersonation attacks succeed.

S2.5. Adv^{-} blocks the messages (H(R‖T‖^{2}Nr) $\oplus^2$M′, H(T‖R‖^{2}M′)) and computes ^{2}M′ = H(R‖T‖^{2}Nr)$\oplus^2$M′ $\oplus$ H(R‖T‖^{2}Nr).

On the other hand, Adv^{-} can compute the new key S (^{2}M′) since the key update mechanism M′ = M + 1. The scheme does not meet forward untraceability. In addition, the data privacy of the tag's key is broken, since the cloud creates M′ which is used to update and store the tag's key in the plaintext.

5. The proposed CROP protocol

To counteract such flaws, the CROP protocol in the cloud platform is presented to protect against various attacks in supply chain management. There are four kinds of participants in Fig. 4: tag owner, reader holder and cloud provider. The notations and descriptions of CROP protocol are shown in Table 1.

Table 2
Encrypted hash table.

Index	Content
H($h(TID)‖K_{TIDc}‖r_i$)	TID^4 mod n‖(H($K_{TIDc}‖r_i$) $\oplus r_{i+1}$)
H($h(TID)‖K_{TIDn}‖r_{i+1}$)	TID^4 mod n'‖(H($K_{TIDn}‖r_{i+1}$) $\oplus r_{i+2}$)
...	...

This paper presents a new EHT, which pre-saves the encrypted keys for every reader. In order to resist against the attack of an untrustworthy cloud provider, an EHT is utilized to protect the stored data and access anonymity. Its structure is illustrated in Table 2. The index which is a hash digest $K_i = H(\mathrm{h}(TID)‖K_{TIDc}‖r_i)$ uniquely denotes the current session with K_{TIDc} and r_i. In order to save the storage space of the tag, the protocol stores the key t_i instead of n. The key n of R_i is calculated using t_i (the Mersenne prime)and the form $2^{t_i}-1$. For instance, $n = 2^{t_i}-1$ and $n' = 2^{t_{i+1}}-1$. The record indexed by $H(h(TID)‖K_{TIDc}‖r_i)$ is M_i=(TID^4 mod $n‖H(K_{TIDc}‖r_i)\oplus r_{i+1}$). The fields ($K_{TIDc}$, r_i) are used to check the integrity of the cipher text after decryption by the reader R_i. For example, the field K_{TIDn} of R_{i+1} is used as the update-secret of the tag for ownership transfer from the old reader to the new reader, the key t_{i+1} is fixed and pre-allocated for R_{i+1}. In order to protect the privacy of R_{i+2} against R_{i+1}, the content $H(K_{TIDn}‖r_{i+1})\oplus r_{i+2}$ is extracted as the pre-distribution key r_{i+2} of R_{i+2} for the tag^{i+1} in the $(i+2)$th session.

The scheme has three phases: an initialization phase, an off-line authentication phase and an ownership transfer phase. The three phases are described as below.

(1) The initialization phase
The R_{i+1} stores the Mersenne prime t_{i+1} instead of n', which is used to compute n' and save the storage space. The R_{i+1} knows two large prime numbers p', q' for $n' = p'q'$. Each tag is set up with (r_{i+1}, r_i, $h(TID)$, K_{TIDc}). The mobile R_i and R_{i+1} keeps the keys (t_i, r_i, $h(TID)$, K_{TIDc}) and (t_{i+1}, r_{i+1}, $h(TID)$, K_{TIDn}), respectively. The data structure of EHT is listed in Table 2, which is initialized for all readers in the system.

(2) An off-line authentication phase

S1. The mobile reader R_{i+1} queries to the tag T_i with an ownership transfer flag.

S2. On receiving the OT flag, the tag T_i generates two nonces (n_T, n_s) and calculates the messages as follows:

$$R_t = h(TID)\oplus v_p \oplus n_s \oplus r_i;$$
$$R_t'' = R_t^4 \mathrm{mod} n;\ R_p = \arg v_p;$$
$$R_{p'} = \arg v_{p'};\ E = r_{i+1}\oplus n_T;$$
$$F = r_i \oplus n_s.$$

S2.1. Then, T_i forwards (R_t'', R_p, $R_{p'}$, E, F) to R_{i+1}.

S2.2. R_{i+1} sends (R_t'', R_p, F) to R_i. In order to protect the privacy of R_{i+1}, the encrypted key n' is sent from R_{i+1} to R_i through the insecure channel.

S3. On receiving the responses (R_t'', R_p, F), R_i executes the following steps.
To acknowledge the received message R_t'' from the tag T_i, R_i solves for the least positive residue $\widetilde{R_t}$ of R_t^2 modulo n and obtains the value of R_t^2 using the legendre symbols of these square roots modulo p and q. R_i retrieves n_s from the received F, and computes $h(TID)\oplus r_i$ which matches for $R_t\oplus v_p\oplus n_s$ with R_p. Then it makes sure the tag is legal. At last, R_i computes $ACK_s = h(TID)\oplus v_{p+l}\oplus n_s$.

S3.1. Then, R_i transmits ACK_s to R_{i+1}.

S3.2. Then, R_i sends $H(h(TID)‖K_{TIDc}‖r_i)$ and $H(K_{TIDc}‖r_i)$ to the cloud.

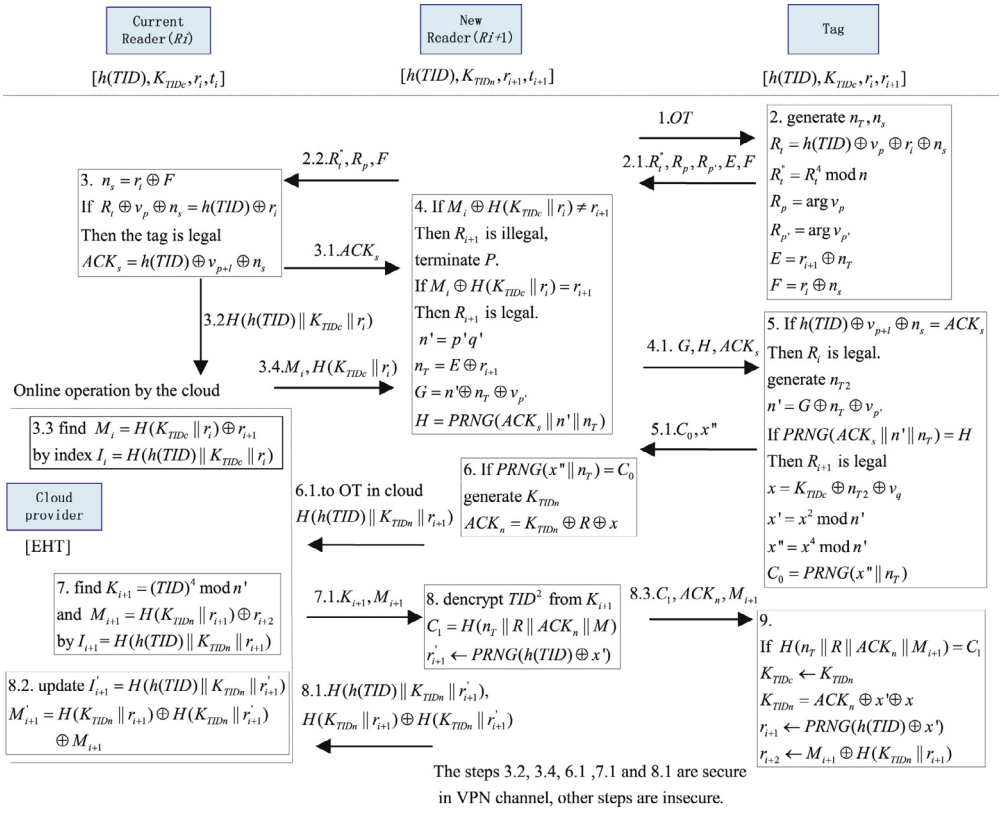

Fig. 4. RFID ownership transfer protocol based on cloud.

S3.3. Then, the cloud finds M_i=TID^4 mod $n\|(H(K_{TIDc}\|r_i) \oplus r_{i+1})$ by the index $H(h(TID)\|K_{TIDc}\|r_i)$.

S3.4. Then, the cloud transmits M_i and $H(K_{TIDc}\|r_i)$ to R_{i+1}.

S4. R_{i+1} extracts r'_{i+1} by computing $(H(K_{TIDc}\|r_i) \oplus r_{i+1}) \oplus H(K_{TIDc}\|r_i)$. Then, it compares the extracted value r'_{i+1} with the stored key r_{i+1}. If $r'_{i+1} \neq r_{i+1}$, R_{i+1} confirms that R_i is illegal and terminates the protocol. If $r'_{i+1} = r_{i+1}$, R_{i+1} confirms that R_i is legal and continues to execute the protocol. Then, R_{i+1} computes $G = n' \oplus n_T \oplus v_{p'}$ and $H = PRNG(ACK_s\|n'\|n_T)$ using the received messages. The validation message H is used to ensure that data integrity of (n', n_T, ACK_s), which prevents the attacker from modifying the transmitted information of the new reader R_{i+1}.

(3) The ownership transfer phase

S5. T_i implements the following steps to authenticate whether the old owner and new owner are legal or not by using the received messages (G, H, ACK_s). At first, the tag computes $ACK'_s = h(TID) \oplus v_{p+l} \oplus n_s$ using its keys ($h(TID)$,v_{p+l}, n_s). If $ACK'_s \neq ACK_s$, the tag verifies that the R_i is illegal and terminates the protocol. If $ACK'_s = ACK_s$, the tag verifies that R_i is legal and continues to execute the protocol. The tag generates the nonce n_{T2}, computes the value $n' = G \oplus n_T \oplus v_{p'}$ and verifies whether the computed value $PRNG(ACK_s\|n'\|n_T)$ is equal to the received H or not. If the above equation is false, the tag verifies that the R_{i+1} is illegal and terminates the protocol. If the above equation holds true, it demonstrates that R_{i+1} communicates with T_i in secure channels, and the values (G, H, ACK_s) are not modified by the attacker. Then T_i continues to calculate $x = K_{TIDc} \oplus n_{T2} \oplus v_q$. It also calculates two residues modulo n', $x' = x^2$ modulo n' and $x'' = x^4$ modulo n', $C_0 = PRNG(x''\|n_T)$.

S5. T_i then forwards the messages (x'', C_0) to R_{i+1}.

S6. In order to acknowledge the challenge, R_{i+1} computes $PRNG(x'' \|n_T)$ and compares with the received value C_0. If $PRNG(x''\|n_T) \neq C_0$, R_{i+1} terminates the scheme. If $PRNG(x''\|n_T) = C_0$, Then, R_{i+1} obtains (R, x) from x'', decodes for the least positive residue $R = x^2$ modulo n' and verifies the key x^2 with the Legendre symbols of these square roots modulo p' and q'. R_{i+1} chooses a new key K_{TIDn}, computes $ACK_n = K_{TIDn} \oplus R \oplus x$, and updates $r_{i+1} = PRNG(h(TID) \oplus x')$.

S6.1. The new reader R_{i+1} informs that the privilege of the old owner R_i on the tag T_i is being revoked. R_{i+1} sends the OT flag and the index $H(h(TID)\|K_{TIDn}\|r_{i+1})$ to the cloud. In a wireless mobile RFID system, the tag,R_i and R_{i+1} need to authenticate one another. For instance, the old owner R_i confirms that the tag is legal in **S3**; Then the new owner R_{i+1} confirms that R_i is legal in **S4**; The tag confirms that R_i and R_{i+1} are legal by checking ACK_s and H in **S5**. Accordingly, the steps 7–9 explain the ownership transfer process in the cloud.

S7. The cloud finds (K_{i+1}, M_{i+1}) by the index $H(h(TID)\|K_{TIDn}\|r_{i+1})$ from EHT.

S7.1. The cloud sends (K_{i+1}, M_{i+1}) to R_{i+1}.

S8. R_{i+1} decrypts TID^2 mod n' from K_{i+1}, and calculates $C_1 = H(n_T\|R\|ACK_n\| M_{i+1})$.

S8.1. Then, R_{i+1} sends its calculated messages $I'_{i+1} = H(h(TID)\|K_{TIDn}\|r'_{i+1})$, and $H(K_{TIDn}\|r_{i+1}) \oplus H(K_{TIDn}\|r'_{i+1})$ to the cloud.

S8.2. Then, the cloud updates the index $I'_{i+1} = H(h(TID)\|K_{TIDn}\|r'_{i+1})$, and parts of content $M'_{i+1} = H(K_{TIDn}\|r_{i+1}) \oplus H(K_{TIDn}\|r'_{i+1}) \oplus M_{i+1}$ in the EHT for the R_{i+1}. After online-updating the index and content which contain the key r'_{i+1} for R_{i+1}, the tag and R_{i+1} will willremain synchronous in the $(i+1)$th session. Even if

the attacker downloaded sensitive and encrypted data from the cloud, she/he cannot obtain the keys of the tag and readers.

S8.3. Then, R_{i+1} sends (M_{i+1}, C_1, ACK_n) to the tag.

S9. The tag uses its stored information to compute $H(n_T\|\mathrm{R}\|\mathrm{ACK}_n\|M_{i+1})$ and compares with the received C_1. If $H(n_T\|\mathrm{R}\|\mathrm{ACK}_n\|M) \neq C_1$, the tag terminates the scheme. If the equation holds, the tag affirms that the received messages (M_{i+1}, K_{TIDn}) are not modified by the attacker. If valid, the tag acknowledges with the updated process as follows:
$K_{TIDc} = K_{TIDn}$;
$K_{TIDn} = ACK_n \oplus x' \oplus x$;
$r_{i+2} = M_{i+1} \oplus H(K_{TIDn}\|r_{i+1})$;
$r_{i+1} = PRNG(h(TID) \oplus x')$.
Meanwhile, the ownership transfer is accomplishable and the $(i+1)$th tag stores the shared key r_{i+2} for R_{i+2} in the $(i+2)$th session.

6. Security, privacy and performance analysis

6.1. Formal privacy and security analysis

The proposed CROP scheme adapts the enhanced security model which reflects the running environment and different attacker's ability based on Vaudenay model. The privacy properties are formally proved by Theorems 7, 8 and 9, while the security properties are proved by Theorems 10, 11, 12 and 13.

After the Learning phase, the attacker frees $vtag^e$ to the set of the tag. In the Challenge phase, the attacker chooses the tested tag $vtag^x$ and assumes that its keys of $vtag^x$ are updated from ${}^{i}vtag^e$ in the $(i+1)$th session. In the Guessing phase, the attacker computes the keys and outputs of ${}^{i+1}vtag^e$ using the corrupted keys and tag encryption structure with the non-negligible probability. Then, in order to determine whether ${}^{i+1}vtag^x$ is updated from ${}^{i}vtag^e$ or not, she/he compares the outputs ${}^{i+1}vtag^e$ with ${}^{i+1}vtag^x$. If the equation holds true, then $x = e({}^{i+1}K_{TIDx} = {}^{i+1}K_{TIDe})$, else $x = |1-e|({}^{i+1}K_{TIDx} \neq {}^{i+1}K_{TIDe})$. In the proposed protocol, the keys and outputs in the $(i+1)$th session depends on the values $({}^{i}x'_e, {}^{i}x_e)$. Then, the security of the proposed protocol is based on the intractability of the integer factorization problem. If the attacker solves the values of ${}^{i}x'_e$ and ${}^{i}x_e$ with the negligible probability, then the protocol is secure. The possibility of computed data ${}^{i}x'_e$ and ${}^{i}x_e$ are discussed in the following cases.

(1) The attacker uses the corrupted key ${}^{i}K_{TIDe}$ and the key-update structure ${}^{i}x_e = K_{TIDe} \oplus n_{T2} \oplus v_q$ to obtain the value ${}^{i}x'_e$ and ${}^{i}x_e$. However, it is impossible to compute ${}^{i}x_e$, since the value ${}^{i}n_{T2}$ is random.

(2) In order to solve the value ${}^{i}x_e$, the attacker uses the output structure ${}^{i}x''_e = ({}^{i}x_e)^4 \bmod n'$ with the monitored messages n' and ${}^{i}x''_e$. However, it is impossible to compute ${}^{i}x_e$, since a public-key cryptosystem is based on intractable for factoring large numbers.

Therefore, the attacker cannot compute the keys and outputs of ${}^{i+1}vtag^e$. Subsequently, she/he cannot compare ${}^{i+1}vtag^e$ with ${}^{i+1}vtag^x$.

Theorem 7. *The proposed CROP protocol achieves strong forward untracebility under the Narrow-strong attacker model.*

Proof. Initialization phase:

1: **Create**Tag(K_{TID0}), **Create**Tag(K_{TID1}).

Learning phase:

2: **Draw**Tag (K_{TIDe}) → $vtag^e$, where $e \in (0, 1)$.
3: **Corrupt**($vtag^e$) → ${}^{i}h(K_{TIDe})$, ${}^{i}K_{TIDe}$, ${}^{i}r_i$, ${}^{i}r_{i+1}$.
4: **Send**R_{new}-tag(π, Init, K_{TIDe}) → ${}^{i}OT$.
5: **Send**Tag-R_{new}($vtag^e$, ${}^{i}OT$) → ${}^{i}R''_t, {}^{i}R_p, {}^{i}R_{p'}, {}^{i}E, {}^{i}F$.
6: **Send**R_{new}-tag (π, ${}^{i}R''_t$, ${}^{i}R_p$, ${}^{i}R_{p'}$, ${}^{i}E$, ${}^{i}F$) → ${}^{i}G$, ${}^{i}H$, ${}^{i}ACK_S$.
7: **Send**Tag-R_{new}($vtag^e$, ${}^{i}G$, ${}^{i}H$, ${}^{i}ACK_S$) → ${}^{i}x''_e$, ${}^{i}C_0$.
8: **Send**R_{new}-Adv(π, ${}^{i}x''_e$, ${}^{i}C_0$) → ${}^{i}M$, ${}^{i}ACK_n$, ${}^{i}C_1$.
9: **Free**($vtag^e$).

Challenge phase:

10: **Draw**Tag (K_{TIDx}) between 2 tags → $vtag^x$.
11: **Launch** → ${}^{i+1}\pi$.
12: **Send**R_{new}-tag(${}^{i+1}\pi$, Init, K_{TIDx}) → ${}^{i+1}$OT.
13: **Send**Tag-R_{new}($vtag^x$, ${}^{i+1}OT$) → ${}^{i+1}R''_t$, ${}^{i+1}R_p$, ${}^{i+1}R_{p'}$, ${}^{i+1}E$, ${}^{i+1}F$.
14: **Send**R_{new}-tag (${}^{i+1}\pi$, ${}^{i+1}R''_t$, ${}^{i+1}R_p$, ${}^{i+1}R_{p'}$, ${}^{i+1}E$, ${}^{i+1}F$) → ${}^{i+1}G$, ${}^{i+1}H$, ${}^{i+1}ACK_S$.
15: **Send**Tag-R_{new}($vtag^x$, ${}^{i+1}G_1$,${}^{i+1}H_1$,${}^{i+1}ACK_S$) → ${}^{i+1}x''_x$, ${}^{i+1}C_0$.
16: **Send**R_{new}-tag(${}^{i+1}\pi$, ${}^{i+1}x''_x$, ${}^{i+1}C_0$) → ${}^{i+1}M$, ${}^{i+1}ACK_n$, ${}^{i+1}C_1$.

Guessing phase:

17: **Compute**(${}^{i}x''_e$,${}^{i+1}x''_x$,${}^{i+1}h(K_{TIDe})$,${}^{i+1}K_{TIDe}$, ${}^{i+1}r_i$,${}^{i+1}r_{i+1}$) → ${}^{i+1}x''_e$.
18: **Compare**(${}^{i+1}x''_e$, ${}^{i+1}x''_x$) → d.
If either ${}^{i+1}x''_x = {}^{i+1}x''_e$, then $x = e$, else $x = |1-e|$.
19: Output whether τ ($vtag^x$) $= {}^{i+1}K_{TIDe}$
$Adv_{A^B}^{\text{S-untra}}(k) = 0 \ll \varepsilon$.
20: Output d=0.

On the one hand, the attacker computes the tag's keys (r_i, r_{i+1}, $h(TID)$, K_{TID}) in the $(i+1)$th session using the known cryptographic structures and the ith keys. Specifically, the common key-update parameter of the $(i+1)$th keys is ${}^{i}x'_e$. If the attacker computes ${}^{i}x'_e$ using the monitored ith and $(i+1)$th messages, then she/he can calculate the tag output ${}^{i+1}x''_x$ and the $(i+1)$th keys. If the attacker cannot compute $(i+1)$th outputs and keys, she/he cannot compare the computed value ${}^{i+1}x''_e$ with the monitored ${}^{i+1}x''_x$. For example, the attacker cannot solve ${}^{i}x'_e$ using ${}^{i}x''_e$ and n', due to the difficulty about the factor decomposed of great number n'. In addition, the attacker cannot compute $x = K_{TIDe} \oplus n_{T2} \oplus v_q$ using the corrupted K_{TIDe}, since n_{T2} and v_q are random. Therefore, the attacker cannot compute ${}^{i}x'_e$ using $x' = x^2$ modulo n'.

At last, the attacker cannot distinguish the target tag $vtag^e$ from $vtag^x$ in the ith session. □

Theorem 8. *The proposed CROP protocol meets backward untracebility under the Narrow-strong attacker model.*

Proof. Initialization phase:

1: **Create**Tag(K_{TID0}), **Create**Tag(K_{TID1}).

Learning phase:

2: **Draw**Tag (K_{TIDe}) → $vtag^e$, where $e \in (0, 1)$.
3: **Corrupt**($vtag^e$) → ${}^{i+1}h(K_{TIDe})$, ${}^{i+1}K_{TIDe}$, ${}^{i+1}r_i$, ${}^{i+1}r_{i+1}$.
4: **Send**R_{new}-tag(${}^{i+1}\pi$, Init, K_{TIDe}) → ${}^{i+1}OT$.
5: **Send**Tag-R_{new}($vtag^e$, ${}^{i+1}OT$) → ${}^{i+1}R''_t$,${}^{i+1}R_p$, ${}^{i+1}R_{p'}$, ${}^{i+1}E$, ${}^{i+1}F$.
6: **Send**R_{new}-tag(${}^{i+1}\pi$, ${}^{i+1}R''_t$, ${}^{i+1}R_p$, ${}^{i+1}R_{p'}$, ${}^{i+1}E$, ${}^{i+1}F$) → ${}^{i+1}G$, ${}^{i+1}H$, ${}^{i+1}ACK_S$.
7: **Send**Tag-R_{new}($vtag^e$, ${}^{i+1}G_1$,${}^{i+1}H_1$,${}^{i+1}ACK_S$) → ${}^{i+1}x''_e$, ${}^{i+1}C_0$.
8: **Send**R_{new}-tag (${}^{i+1}\pi$, ${}^{i+1}x''_e$, ${}^{i+1}C_0$) → ${}^{i+1}M$, ${}^{i+1}ACK_n$, ${}^{i+1}C_1$.
9: **Free**($vtag^e$).

Challenge phase:

10: **Draw**Tag (K_{TIDx}) between two tags → $vtag^x$.

11: **Launch** $\rightarrow {}^{i}\pi$.
12: **Send**R$_{new}$-tag(π, Init, K_{TIDx}) $\rightarrow {}^{i}OT$.
13: **Send**Tag-R$_{new}$($vtag^{x}$, ${}^{i}OT$) $\rightarrow {}^{i}R''_{t}$, ${}^{i}R_{p}$, ${}^{i}R_{p'}$, ${}^{i}R_{p'}$, ${}^{i}E$, ${}^{i}F$.
14: **Send**R$_{new}$-tag(π, ${}^{i}R''_{t}$, ${}^{i}R_{p}$, ${}^{i}R_{p'}$, ${}^{i}E$, ${}^{i}F$) $\rightarrow {}^{i}G$, ${}^{i}H$, ${}^{i}ACK_{S}$.
15: **Send**Tag-R$_{new}$($vtag^{x}$, ${}^{i}G$, ${}^{i}H$, ${}^{i}ACK_{S}$) $\rightarrow {}^{i}x''_{x}$, ${}^{i}C_{0}$.
16: **Send**R$_{new}$-tag(π, ${}^{i}x''_{e}$, ${}^{i}C_{0}$) $\rightarrow {}^{i}M$, ${}^{i}ACK_{n}$, ${}^{i}C_{1}$.

Guessing phase:

17: **Compute**(${}^{i}x''_{e}$, ${}^{i+1}x''_{x}$, ${}^{i+1}h(K_{TIDe})$, ${}^{i+1}K_{TIDe}$, ${}^{i+1}r_{i}$, ${}^{i+1}r_{i+1}$) $\rightarrow {}^{i}x''_{e}$
18: **Compare**(${}^{i}x''_{e}$, ${}^{i}x''_{x}$) $\rightarrow d$,
If either ${}^{i}x''_{e} = {}^{i}x''_{x}$, then $x = e$, else $x = |\ 1\text{-}e|$.
19: Output whether τ ($vtag^{x}$) $=^{i+1} K_{TIDe}$.
$Adv^{\text{B-untra}}_{A^B}(k) = 0 \ll \varepsilon$.
20: Output d=0.

The attacker cannot compute the ith keys of the tag, even if the attacker corrupts the $(i+1)$th keys of the tag and monitors the $(i+1)$th sessions. Meanwhile, the attacker cannot speculate and modify the ith output ${}^{i}x''_{e}$ without the tag's keys. The attacker cannot compare the computed value ${}^{i}x''_{e}$ with the monitored value ${}^{i}x''_{x}$, she/he is unable to distinguish between $vtag^{e}$ and $vtag^{x}$ in the ith session. Therefore, $Adv^{\text{B-untra}}_{A^B}(k)$ is negligible in k. In addition, the proposed CROP protocol meets backward untracebility for R_{i+2}. Since the keys r_{i+2} of R_{i+2} are encrypted storage in cloud database, the attacker, R_i and R_{i+1} cannot be obtained r_{i+2}. □

Theorem 9. *The proposed CROP protocol achieves the forward untracability.*

Proof. If the proposed CROP protocol achieves strong forward untracability, then it meets forward untracability definition, since the attackers ability in strong forward untracability definition is stronger than forward untracability. □

An RFID ownership transfer protocol P meets backward untraceability, strong forward untraceability and forward untraceability, for any probabilistic polynomial time adversary A^B, $Adv^{\text{F-untra}}_{A^B}(k)$, $Adv^{\text{B-untra}}_{A^B}(k)$ and $Adv^{\text{S-untra}}_{A^B}(k)$ are negligible in k, respectively.

Theorem 10. *The proposed CROP protocol is resistant to inner legitimate and outer illegitimate reader impersonation attacks under the weak attacker.*

Proof. The two attacks are analyzed by using three instances under the weak attacker model, respectively.

(1) Resistance against the legitimate new reader impersonates the old reader
The new reader cannot compute the same outputs as, and deduce the key of, the old reader's outputs (G, H) by using the communication between the old reader and the new reader under the assumption that the old reader does not participate in the RFID systems.
(2) Resistance against the legitimate old reader impersonates the new reader
The old reader monitors the inputs and outputs of the new reader, but the old one cannot compute the same output $ACKs$ and the updated keys of the new reader by using the old reader's keys and the known messages under the assumption that the new reader does not create the message $ACKs$.
On the one hand, the old reader computes the entities' keys from the monitored ith messages. Since the transmitted messages are encrypted, the attacker cannot obtain any entities' keys in any session. On the other hand, the attacker corrupts the ith tag's keys and monitors the transmitted messages, she/he cannot compute the messages which are the same as the outputs of the new reader.
(3) Resistance against outer illegitimate reader impersonation attack
In order to prevent the attacker from modifying the messages and achieve the data integrity of the transmitted information, each output has a corresponding verification message. Specifically verification messages contain the transmitted information and the unknown tag's key. In other words, the attacker cannot construct legal validation messages under the weak model. □

Theorem 11. *The proposed CROP protocol resists replay attacks and tracing attacks under the weak attacker model.*

Proof. For the modified C_0, the tag's output x'' must be different in each section, since n_{T2} and v_p are changed. In addition, the verification message C_0 also becomes quite uncertain, because the messages (n, x'', n_T) are changed. This analysis is also applied to the reader's outputs. There is no fixed relationship among the different information in different sessions. Thus, the scheme resists tracing attacks and replay attacks. □

Theorem 12. *The proposed CROP protocol resists tag impersonation attacks and desynchronization attacks under the weak attacker model.*

Proof. The desynchronization attacks are analyzed by using three cases under the weak attacker model, respectively.

(1) The attacker prompts the reader to update the tag's keys twice after the tag impersonation attack, but the tag does not update its keys.
Due to the modification of message x'' in our improved scheme, it is quite difficult for an attacker to forge the validation message C_{0adv}. It means that the attacker cannot easily produce a set of fake information from the tag that can be verified by the reader.
(2) When the modified outputs of the reader are verified by the tag, then the reader normally updates the tag keys and the tag updates its keys using the wrong parameters.
Theorem 10 has proved that the reader impersonation attacks are impossible, so the desynchronization attacks fail in the proposed protocol.
(3) The attacker prevents the reader from updating the tag keys, and allows the tag to update its keys.
The designed protocol adapts the reader to update the tag keys before the tag updates its key. Therefore, the desynchronization attacks cannot exist in this situation. Ultimately, the improved scheme can prevent the system from desynchronization attacks and tag impersonation attacks.

□

Theorem 13. *The proposed CROP protocol meets database security.*

Proof. The requirements of database security are introduced as follow. Firstly, the keys of the tag and the reader are stored and transmitted in ciphertext. Secondly, the inputs and outputs of the database are transferred in encrypted form. For example, the encrypted messages and hashed data are listed in the EHT, and the decryption is not executed by the cloud but by the readers themselves. Therefore, any keys of tags and readers are not revealed by the malicious or compromised cloud and reader. □

Table 3
Comparisons of the security, privacy and performance properties.

Protocols		[3]	[16]	[17]	[18]	[9]	Ours
Security and privacy properties	S1	NO	YES	YES	YES	YES	YES
	S2	NO	NO	NO	NO	NO	YES
	S3	NO	YES	YES	NO	NO	YES
	S4	YES	YES	YES	NO	NO	YES
	S5	NO	YES	YES	NO	NO	YES
	S6	–	–	–	NO	NO	YES
	S7	NO	YES	YES	YES	YES	YES
	S8	NO	NO	NO	NO	NO	YES
	S9	NO	NO	NO	NO	NO	YES
	S10	YES	NO	NO	NO	NO	YES
Performance properties	PR1	4H	4H + 3M	3M	2M + 2C	3M	2H + 3M
	PR2	3/1	4/4	4/8	4/5	4/5	4/4
	PR3	O(1)	O(4)	O(4)	O(n)	O(n)	O(1)
	PR4	YES	NO	NO	NO	NO	YES
	PR5	YES	NO	NO	NO	NO	YES
	PR6	NO	NO	NO	YES	YES	NO

S1: Replay attack resistance; S2: Traceability attack resistance; S3: Tag impersonation attack resistance; S4: Desynchronization attack resistance; S5: Outer illegitimate reader impersonation attack resistance; S6: Inner legitimate reader impersonation attack resistance; S7: Backward untracebility; S8: Strong forward untracebility; S9: Forward untracebility; S10: Database security. PR1: Computation (T); PR2: Storage space (T/R); PR3: Scalability; PR4: Pervasive (ubiquitous) authentication. PR5: Off-line authentication; PR6: Conforming to EPCC1G2. H: Hash function; C: Cyclic Redundancy Code; M: Mod.

This paper provides a brief overview of these formal definitions to analyze the proposed CROP protocol. In order to prove that the proposed CROP protocol meets private requirements, the formal analysis methods are applied by using the RFID privacy model. In addition, the proof results show that the proposed protocol meets the reader and tag impersonation attack resistance, desynchronization attack resistance, replay attack resistance, tracing attack resistance, and database security.

6.2. Performance and applicability analysis

A. Computation cost and storage space

In the proposed CROP protocol, the mobile readers perform the operations such as encryption, decryption, data storing and searching assignments, meanwhile the cloud supports the keys-update and distributes mobile readers' keys.

However, the tag only supports Chinese Remainder Theorem encryption, due to the use of EHT which protects the privacy of new reader R_{i+2} against old reader R_i, the current reader R_{i+1} and the cloud. In addition, the tag reduces the storage space, since it does not need to decrypt p' and q', and does not store the key n' of the new reader. The current reader R_i stores t_i rather than n, which reduces the storage space. Furthermore, the readers are required to support Chinese Remainder Theorem encryption and decryption due to the use of EHT which keeps client's privacy from being revealed to the cloud.

B. Scalability

The computational complexity (scalability) is that a tag is identified by a verifier (mobile reader or cloud). The new reader is able to find the matched tag's record using r_i with scalability O(1), since $H(h(TID)\|K_{TIDc}\|r_i)$ and $H(h(TID)\|KTID_n\|r_{i+1})$ as two indexes are created by an old reader and by a new reader, then are sent to a cloud. The tag reads the keys of R_{i+2} from the EHT with computational complexity O(1)in the ith session. However, the scalability of the schemes [16,17] is O(4). The reason is that a solution exists for $x'' = x^4 \bmod n'$, there are the four possible solutions (x', x) according to the Chinese Remainder theorem, only one of those would be a quadratic residue modulo n' satisfying $x' = x^2 \bmod n'$.

C. Off-line authentication and online update

An off-line reader authenticates tags without connecting to a cloud. In this way, the method improves the efficiency of the system verification and saves verification time. However, in order to prevent the old and the new reader from obtaining the key of R_{i+2}, the shared key between tag and reader R_{i+2} is updated online in the cloud database.

D. Ubiquitous (pervasive) authentication

The proposed CROP protocol utilizes Cloud computing to execute ubiquitous (pervasive) authentication by mobile readers wherever and whenever, provided that the login user's identity (r_i) is constantly changing. Therefore, the proposed CROP protocol is ubiquitous.

6.3. Evaluations and comparisons

The proposed CROP protocol compares with related schemes and is evaluated in terms of the security, privacy and performance properties. The comparisons of the security, privacy and performance properties are listed in Table 3.

According to the above comparisons, the proposed scheme is middleweight and low-cost to support PRNG and hash functions. It is not necessary to analyze the schemes [3,16,17] which are not OT protocols in term of the inner legitimate reader impersonation attack. Compared to other schemes, the proposed scheme's advantages lie in:

(1) The proposed scheme is resistant against replaying, tracing and desynchronization attack and achieves the untraceability properties in the Section 6.1.
(2) The proposed scheme is scalable and offers the ubiquitous and off-line authentication service when it is applied to the large-scale and low-cost applications in Section 6.2.

It means that the proposed CROP protocol is scalable in the large-scale application with a huge number of tags. Thus, the privacy properties of tags and readers are required to be protected against the attackers and the cloud provider. Therefore, the privacy requirements of different readers are achieved in the ownership transfer process.

7. Conclusions

Existing RFID protocols are inapplicable to ownership transfer and cloud-based applications, since they do not meet the primary requirements. Moreover, in order to support mobile, remote and cloud-platform data access, the proposed CROP scheme integrates cloud service and quadratic residue mechanisms to provide backward untraceablity, forward untraceablity and strong forward untraceablity properties. The ownership transfer scheme in the cloud platform is expected to be applied in a variety of applications. In the future, we will optimize the performance (authentication efficiency) by reducing the number of hash computations in the DB, integrate more practical features for supply chain management, and explore the simulation of the real experiment. The open issues include energy-efficient data processing in supply chain management, software infrastructures for supporting IoT and interaction models for hand-held and mobile devices.

Conflicts of interest

The authors declare no conflict of interest.

Acknowledgments

This work is supported by the Fundamental Research Funds for the Central Universities (no. 2015XKMS086).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.comnet.2016.05.017

References

[1] L.N. Sun, Building intelligent parking lot based on RFID and cloud computing technology, in: Proceedings of International Conference on Mechatronics and Semiconductor Materials (ICMSCM 2013), Xian, Peoples R China, SEP 28-29, 2013, pp. 1550–1553.
[2] S.M. Chen, M.E. Wu, H.M. Sun, K.H. Wang, CRFID: an RFID system with a cloud database as a back-end server, Future Gener. Comput. Syst. 30 (2014) 155–161.
[3] W. Xie, L. Xie, C. Zhang, Q. Zhang, C.J. Tang, Cloud-based RFID authentication, in: Proceedings of IEEE International Conference on RFID, APR 30-MAY 02, Orlando, FenLan, 2013, pp. 168–175.
[4] W. Xie, L. Xie, C. Zhang, Q. Wang, C. Wang, C.J. Tang, TOA a tag-owner-assisting RFID authentication protocol toward access control and ownership transfer, Secur. Commun. Netw. 7 (5) (2014) 934–944.
[5] J. Munilla, F. Guo, W. Susilo, Cryptanalaysis of an EPCC1g2 standard compliant ownership transfer scheme, Wirel. Pers. Commun. 72 (1) (2013) 245–258.
[6] X. Wang, C.W. Yuan, Scalable and resynchronisable radio frequency identification ownership transfer protocol based on a sliding window mechanism, IET Inf. Secur. 8 (3) (2014) 161–170.
[7] S. Martinez, M. Valls, C. Roig, J.M. Miret, F. Gine, A secure elliptic curve-based RFID protocol, J. Comput. Sci. Technol. 24 (2) (2009) 309–318.
[8] M.R. Alagheband, M.R. Aref, Simulation-based traceability analysis of RFID authentication protocols, Wirel. Pers. Commun. 77 (2) (2014) 1019–1038.
[9] R. Doss, W.L. Zhou, S. Yu, Secure RFID tag ownership transfer based on quadratic residues, IEEE Trans. Inf. Forensics Secur. 8 (2) (2013) 390–401.
[10] D. Zhang, Y. Qian, J. Wan, S. Zhao, An efficient RFID search protocol based on clouds, Mob. Netw. Appl. 20 (3) (2015) 356–362.
[11] G.N. Khan, M. Moessner, Low-cost authentication protocol for passive, computation capable RFID tags, Wirel. Netw. 71 (2) (2015) 565–580.
[12] C.T. Li, C.Y. Weng, C.C. Lee, A secure RFID tag authentication protocol with privacy preserving in telecare medicine information system, J. Med. Syst. 39 (8) (2015) 77.
[13] K. Srivastava, A.K. Awasthi, S.D. Kaul, A secure RFID tag authentication protocol with privacy preserving in telecare medicine information system, J. Med. Syst. 39 (1) (2015) 153.
[14] T.J. Cao, P. Shen, E. Bertino, Cryptanalysis of some RFID authentication protocols, J. Commun. 3 (7) (2008) 20–27.
[15] Y. Chen, J.S. Chou, H.M. Sun, A novel mutual authentication scheme based on quadratic residues for RFID systems, Comput. Netw. 52 (12) (2008) 2373–2380.
[16] T.C. Yeh, C.H. Wu, Y.M. Tseng, Improvement of the RFID authentication scheme based on quadratic residues, Comput. Commun. 34 (3) (2011) 337–341.
[17] R. Doss, S. Sundaresan, W.L. Zhou, A practical quadratic residues based scheme for authentication and privacy in mobile RFID systems, Ad Hoc Netw. 11 (1) (2013) 383–396.
[18] R. Doss, W.L. Zhou, A secure tag ownership transfer scheme in a closed loop RFID system, in: Proceedings of IEEE Wireless Communications and Networking Conference (WCNC), Apr 01, Paris, France, 2012, pp. 164–169.
[19] S. Vaudenay, in: K. Kurosawa (Ed.), On privacy models for RFID, vol. 4833, Springer, Berlin Heidelberg, 2007, pp. 68–87.
[20] S. Kardaş, S. Çelik, A. Arslan, A. Levi, in: G. Avoine, O. Kara (Eds.), An efficient and private RFID authentication protocol supporting ownership transfer, vol. 8162, Springer, Berlin Heidelberg, 2013, pp. 130–141.
[21] C.W. Phan, J. Wu, K. Ouafi, et al., Privacy analysis of forward and backward untraceable RFID authentication schemes, Wirel. Pers. Commun. 61 (1) (2011) 69–81.
[22] D. Boopathy, M. Sundaresan, Secured Cloud Data Storage-Prototype Trust Model for Public Cloud Storage, in: Proceedings of International Conference on ICT for Sustainable Development, Springer, Singapore, 2016.
[23] S. Kardas, S. Celik, M.A. Bingol, et al., A New Security and Privacy Framework for RFID in Cloud Computing, in: Proceedings of the 2013 IEEE International Conference on Cloud Computing Technology and Science, vol. 01, IEEE Computer Society, 2013, pp. 171–176.

Tianjie Cao received the BS and MS degree in mathematics from Nankai University, Tianjin, China and the PhD degree in computer software and theory from State Key Laboratory of Information Security of Institute of Software, Chinese Academy of Sciences, Beijing, China. He is a professor of computer science in School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, China. From 2007 to 2008, he has been a visiting scholar at the Department of Computer Sciences and CERIAS, Purdue University. His research interests are in security protocols and network security.
Email: tb12170002@cumt.edu.cn.

Xiuqing Chen received her bachelor's degree and master's degree from the China University of Mining and Technology. She has been a Ph.D. degree candidate in Computer Software and Theory from the China University of Mining and Technology. She will be a teacher in School of Medicine Information, Xuzhou Medical College, Xuzhou, 221000, China in September 2015. Her research interests include security protocols and network security.
Email: xiuqingchen@126.com.