Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/080450

A New Scalable RFID Delegation Protocol

Xiuqing Chen[1,*], *Tianjie Cao*[1,*] *and Yu Guo*[2]

[1] School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China
[2] School of Materials Science and Engineering, China University of Mining and Technology, Xuzhou 221116, China

Received: 19 Aug. 2013, Revised: 21 Nov. 2013, Accepted: 22 Nov. 2013
Published online: 1 Jul. 2014

Abstract: The radio frequency identification (RFID) protocols are vulnerable to various attacks from an active or passive adversary. We shed light upon some existing security flaws in these delegation protocols. It is useful to mitigate many security weaknesses in such delegation protocols to promote the acceptance of RFID tags. We propose that a scalable RFID delegation protocol will be against traceability attacks with a stateful variant so that it provides the claimed security requirements. Compared with the previous schemes, we emphasize three critical distinctions in our protocol. First, the reader and the tag decrease one bitwise XOR of the message and reduce the computational complexity. Second, a solution to reducing the maximum search complexity can be that add two different flags to the tags responses in order to distinguish delegation request from delegation update. Third, the number of exchanged flows during different cases of our revision is same. Finally, the proposed scheme achieves scalability and untraceability property without leading to a security collision.

Keywords: RFID, delegation protocol, authenticate, scalability, untraceability.

1 Introduction

Radio frequency identification (RFID) is a wireless contact that enables the reader to identify objects automatically in a wide variety of environments.

RFID system mainly consists of four components: tags, a set of online readers, a set of off-line readers, and a backend database. Consumer products identification, inventory control and contact less credit-card are the applications where RFID devices are deployed [1,2]. However, the widespread application of RFID systems to supply-chain management may expose challenging security issues either to corporations or to individuals [3, 4]. Some lightweight authentication protocols [5,6] can provide privacy conditions, but suffer from the scalability issue in the large-scale RFID deployments. However, other protocols have serious security and privacy flaws that contradict with the security requirements. In [3], Erguler et al. introduce the various attacks on the privacy of RFID systems based on unbalanced state.

More precisely, a backend database requiring a linear computational complexity brings up the scalability problem. Adding the delegation mechanism may be a solution to achieving the scalability property of the proposed protocol.

Song et al. present an efficient RFID delegation protocol [7] (the W0 scheme) that takes constant time to identify a tag. The Habibi's protocol [8] (the W1 scheme) has scalability weakness and suffers from the backward traceability attack. Another security and private analysis on the W0 protocol is made by Erguler [9] (the W2 scheme). They present tag impersonation, desynchronization, and traceability attacks on the W0 protocol. Afterwards, we show new security flaws in the W2 protocol and prove that the update protocol is vulnerable to backward traceability attack.

After analyzing various protocols, we conclude that the best protocol should be designed in terms of scalability and untraceability security [10]. Furthermore, we propose a scalable RFID delegation protocol (SRDP) which is partly based on the W2 protocol. The proposed methods do not abandon primitive structure, nor do they incur more computation complexity. Instead, our protocol utilizes methods that have already been available in protocol to improve protocol scalability. Except its obvious advantage in identification efficiency, our protocol also has advantages in security.

The remaining of the article is organized as follows. Section 2 discusses preliminary and related work and points out the vulnerabilities in Erguler's scheme. Section

[*] Corresponding author e-mail: xiuqingchen@126.com, tjcao@cumt.edu.cn

3 proposes the revision. Section 4 shows security analysis and performance comparisons. At last, conclusions and further work are given in Section 5.

2 Preliminary and related work

Many researches have been conducted to provide a scheme for scalable RFID delegation. The goal in those researches is to achieve the following requirements. We aim to review Erguler's scheme and show its weaknesses in the remaining part of this section.

Secure communication schemes for constructing the formal privacy models in the proof should provide some security requirements [11], which are described as follows:

(a) **Resistance to impersonation attack**

Impersonation attacks in stateful protocols can make a tag (or reader) leak secrets to an adversary. An adversary may eavesdrop communications and impersonate a tag (or reader) using the compromised tag (or reader).

(b) **Resistance to DoS attack**

The protocol is vulnerable to DoS attack regardless of the weakness of protocol in detail. DoS attack can be achieved by malicious query. The protocol requires updating-and-synchronizing states between the server and the tag to prevent desynchronization attack.

(c) **Resistance to tracking attack**

Many privacy models have been proposed, which require indistinguishability [12] and forward untraceability [13].

The concept of untraceability is that an adversary can not distinguish two different tags according to the given messages of these tags. The concept of forward untraceability is as follows: even if an adversary gets the secret of a tag, he is not able to trace history messages of that tag. In other words, a protocol achieves untraceability if an adversary could not trace a tag by deducing from the link among different rounds. This kind of tracking attack must be taken into account in the analysis of RFID protocols.

RFID protocol desires the following performance requirements as follows [7]:

(a) **Computation**

Regarding the complexity of tag computations, a better scheme takes less hash operations.

(b) **Communication**

The number of communication messages between a tag and a reader should be reduced.

(c) **Scalability**

The server performs a tedious search $O(n)$ over the list of all tag entries in *DB*. Such an exhaustive search procedure brings up the scalability conflict. In general, the verification of a tag takes only O (1) to find the match in the look-up tables. RFID protocol needs such look-up tables which comply with the large-scale RFID applications while addressing scalability issues.

Table 1: A list of parameters and notations

Notation	Description
T	Tag
IDR^+	Online reader identifiers
IDR^-	Off-line reader identifiers
DB	The backend database
A	Adversary
$h()$	One-way hash function
$e(),f(),g(),d()$	Keyed one-way hash
$Req1(Req2)$	delegation request 1 (update2)
z	a key shared by *DB* and tag
δ	a stamp is computed by *DB*
k_j	$= h(s_j)$, computed by *DB*
$\|$	Concatenation operation
n	The number of tags
c,m,m'	The value of a couter

In [14], Sharaf pointed out these flaws and later modified the W0 scheme by applying the TCP/IP three-way handshake protocol [15] in the secret update and session termination.

It is useful to introduce a counter in the delegated readers. Such a delegated mechanism will be effective in ensuring controlled delegation and scalability property. At the opposite of this mechanism, the proposal [16] has the advantage of supporting controlled delegation without needing a counter on the tag's side. Fernndez-Mir et al. propose the RFID protocol that not only achieves controlled delegation but also improves the system scalability [17].

2.1 Review of the W2 protocol

The following notations and a list of parameters are used throughout the article in the TABLE I.

A recapitulative overview of the W2 scheme is provided in Figure 1 for convenience.

Initial Phase: Each tag stores k,x and a counter $c = m$. *DB* stores $\hat{s}$, $\hat{k}$ and the identifiers $x_0, x_1, \cdots, x_m$ as the corresponding entry, where $\hat{s}$ and $\hat{k}$ represent the most recent previous values of s and k, respectively.

Authentication Phase: The procedure of the authentication phase is depicted as follows.

Case 1 **(Tag Authentication)**

1. *DB* can transmit random string r_R and IDR^- to the tag.

2. Upon reception of r_R and IDR^-, the tag checks whether c equals to zero. If not, then the following steps are performed:

(a) The tag calculates $\delta = d_z(IDR^- \| k)$, $M_T = f_k(r_R \| x \| \delta)$ and updates x to $e_k(x \| z)$ and c to $c-1$, where k and δ are the key belonging to the identified tag. Then it transmits (r_R, M_T, x) to the offline reader.

(b) Upon reception of (M_T, x, r_R), *DB* performs the following steps.

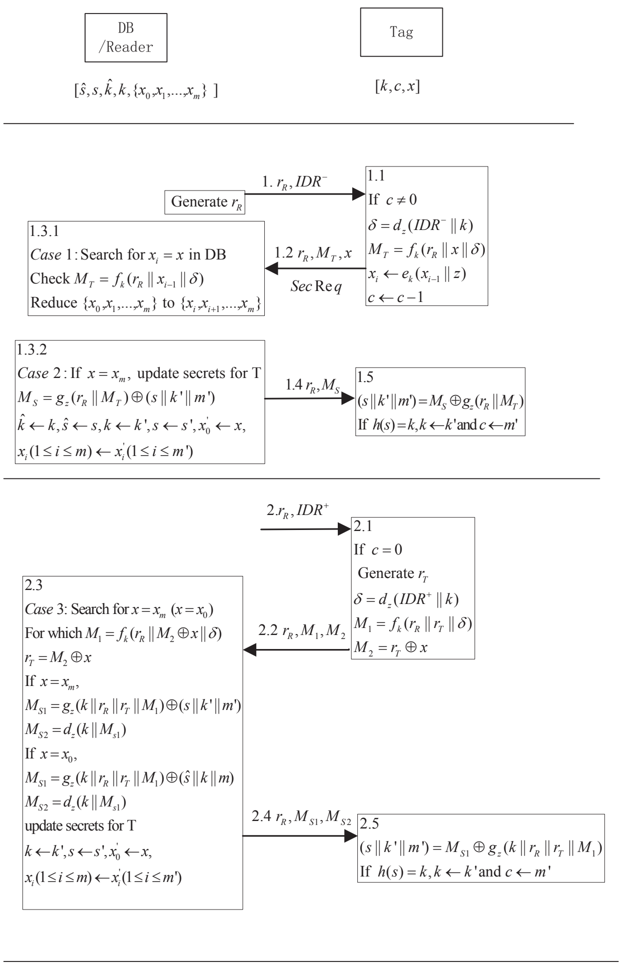
DB /Reader
Tag
$[\hat{s}, s, \hat{k}, k, \{x_0, x_1, \dots, x_m\}]$
$[k, c, x]$
Generate r_R
1. r_R, IDR^-
1.1
If $c \neq 0$
$\delta = d_z(IDR^- \| k)$
$M_T = f_k(r_R \| x \| \delta)$
$x_i \leftarrow e_k(x_{i-1} \| z)$
$c \leftarrow c-1$
1.2 r_R, M_T, x
$Sec\,Req$
1.3.1
Case 1: Search for $x_i = x$ in DB
Check $M_T = f_k(r_R \| x_{i-1} \| \delta)$
Reduce $\{x_0, x_1, \dots, x_m\}$ to $\{x_i, x_{i+1}, \dots, x_m\}$
1.3.2
Case 2: If $x = x_m$, update secrets for T
$M_S = g_z(r_R \| M_T) \oplus (s \| k' \| m')$
$\hat{k} \leftarrow k, \hat{s} \leftarrow s, k \leftarrow k', s \leftarrow s', x_0' \leftarrow x,$
$x_i(1 \le i \le m) \leftarrow x_i'(1 \le i \le m')$
1.4 r_R, M_S
1.5
$(s \| k' \| m') = M_S \oplus g_z(r_R \| M_T)$
If $h(s) = k, k \leftarrow k'$ and $c \leftarrow m'$
2. r_R, IDR^+
2.1
If $c = 0$
Generate r_T
$\delta = d_z(IDR^+ \| k)$
$M_1 = f_k(r_R \| r_T \| \delta)$
$M_2 = r_T \oplus x$
2.2 r_R, M_1, M_2
2.3
Case 3: Search for $x = x_m$ $(x = x_0)$
For which $M_1 = f_k(r_R \| M_2 \oplus x \| \delta)$
$r_T = M_2 \oplus x$
If $x = x_m$,
$M_{S1} = g_z(k \| r_R \| r_T \| M_1) \oplus (s \| k' \| m')$
$M_{S2} = d_z(k \| M_{s1})$
If $x = x_0$,
$M_{S1} = g_z(k \| r_R \| r_T \| M_1) \oplus (\hat{s} \| k \| m)$
$M_{S2} = d_z(k \| M_{s1})$
update secrets for T
$k \leftarrow k', s \leftarrow s', x_0' \leftarrow x,$
$x_i(1 \le i \le m) \leftarrow x_i'(1 \le i \le m')$
2.4 r_R, M_{S1}, M_{S2}
2.5
$(s \| k' \| m') = M_{S1} \oplus g_z(k \| r_R \| r_T \| M_1)$
If $h(s) = k, k \leftarrow k'$ and $c \leftarrow m'$

Fig. 1: The W2 scheme.

i. *DB* uses the received x to quickly search for x_i and identifies the tag in constant time.

ii. Checks that $M_T = f_k(r_R \parallel x_{i-1} \parallel \delta)$. If the equation is verified, *DB* authenticates the tag. Next, it reduces the corresponding tags' identifiers from $\{x_0, x_1, \cdots, x_m\}$ to $\{x_i, x_{i+1}, \cdots, x_m\}$.

iii. If $x \neq x_m$, then the protocol terminates successfully.

Case 2 **(Secret Update I)**

iv. If $x = x_m$,

A. The tag's secrets are renewed: $k' \leftarrow k$,

$k \leftarrow g_z(M_T \| r_R \| k \| x)$, $m' \leftarrow h(k)$, $x_0' \leftarrow x$, $x_i(1 \le i \le m) \leftarrow x_i'(1 \le i \le m')$ where $x_i' = e_k'(x_{i-1}' \parallel z)$ for $1 \le i \le m$, where $x_0 = x$.

B. Computes $M_s = g_z(r_R \| M_T) \oplus (s \| k' \| m')$ and transmits (r_R, M_s) to the tag.

C. Updates the secrets of the identified tag from $\{\hat{k}, \hat{s}, k, s, x_0, x_m\}$ to $\{k, s, k', s', x_1', x_m'\}$.

(c) Upon reception of (r_R, M_s), the tag checks whether c equals to zero.

i. If $c = 0$, T checks $(s \| k' \| m') = M_s \oplus g_z(r_R \| M_T)$ using k, r_R and M_T. If $h(k) = k$, T authenticates *DB* successfully and updates its secrets from $\{k, c\}$ to $\{k', m'\}$.

ii. If $c \neq 0$, suspends the protocol.

Case 3 **(Secret Update II)**

2. Upon reception of r_R and IDR^+ , the tag produces a random nonce r_T and checks whether c equals to zero. If yes, a step by step description is also given below:

(a) The tag calculates the messages $\delta = d_z(IDR^+ \| k)$, $M_1 = f_k(r_R \| r_T \| \delta)$ and $M_2 = r_T \oplus x$. Then it transmits $(r_R, M_1, M_2, SecReq)$ to *DB*.

(b) Upon reception of r_R, M_1, M_2 and *SecReq*, *DB* implements the following steps.

i. Searches for a value $x = x_m$ or $x = x_0$ for which $M_1 = f_k(r_R \| M_2 \oplus x \| \delta)$. Then *DB* finds the match and authenticates the tag.

ii. If $x = x_m$,

A. Then the secrets are renewed: $k \leftarrow k'$, $k' \leftarrow h(s')$, $m' \leftarrow h(k)$, $x_0' \leftarrow x$, $x_i(1 \le i \le m) \leftarrow x_i'(1 \le i \le m')$ where $x_i' = e_k'(x_{i-1}' \| z)$.

B. Computes $M_{s1} = g_z(k \| r_R \| r_T \| M_1) \oplus (s \| k' \| m')$ and $M_{s2} = d_z(k \| M_{s1})$.

C. Updates the secret of the identified tag from $\{\hat{k}, \hat{s}, k, s, x_0, x_m\}$ to $\{k, s, k', s', x_1', x_m'\}$.

iii. If $x = x_0$, *DB* calculates $M_{s1} = g_z(k \| r_R \| r_T \| M_1) \oplus (\hat{s} \| k \| m)$ and $M_{s2} = d_z(k \| M_{s1})$, then transmits (r_R, M_{s1}, M_{s2}) to the tag.

iv. *DB* updates the secret of the identified tag from $\{\hat{k}, \hat{s}, k, s, x_0, x_m\}$ to $\{k, s, k', s', x_1', x_m'\}$, $k = g_z(M_1 \| r_T \| r_R \| k \| x)$, $m' = h(k)$ where $x_i' = e_k'(x_{i-1}' \| z)$.

(c) Upon reception of (r_R, M_{s1}, M_{s2}), the tag checks whether $(s \| k' \| m') = g_z(k \| r_R \| r_T \| M_1) \oplus M_{s1}$. If $h(s) = k$, T authenticates *DB* successfully and updates its secrets from $\{k, c\}$ to $\{k', m'\}$.

2.2 Vulnerabilities in W2 scheme

In this section, we show the security drawback of the W2 scheme and demonstrate that the Erguler's protocol is vulnerable to tag tracking attack.

We find that the W0 protocol suffers from tag tracking attack under an assumption of a compromised delegated reader scenario in [18]. The assumption is reasonable. The delegated readers can be mobile devices such as mobile phones, mobile computers, and a portable card reader and so on. These readers have an access to *DB* and work in offline model. Hence, it is not difficult for an adversary to steal all the information by the lost or dishonest reader. This possible security flaw apparently makes the W2 protocol insecure. So an adversary obtaining the secrets from a compromised reader can easily verify a fake tag as valid. Indeed, due to these potential risks, the protocols usually use temporarily delegation model so that the delegated reader can identify the tags with a limited number of times.

In addition to this, the W1 protocol does not provide a limited delegation mechanism: a delegated reader can not know the specific number of the tags. Therefore, an adversary can make rogue queries on a tag more than the legal number n that lead to DoS attack.

There are privacy attacks that an adversary can trace tags from the number of protocol message flows. The tracing attack violates the tag's location privacy of the protocol. Intuitively, a protocol satisfies untraceability property if an adversary is not able to distinguish two different tag's outputs. The concept of untraceability is used formally in security models, such as by Avoine [19]. According to [11], as a formal definition, forward untraceability can be defined in terms of privacy experiments.

Definition 1 (Forward untraceability). In a complete protocol, an adversary can not obtain the tag's secret from its responses without a valid reader.

2.3 Privacy experiment

Our scheme uses formal privacy model to evaluate the privacy level of the W2 protocol. The adversary A has access to running the following oracles [1].

- O$^{Execute}(R,T,m,i)$: The adversary A passively eavesdrops on the communication channel and monitors exchanged messages between tag and reader in session i of a truthful protocol execution.

- O$^{Send}(R,T,m,i)$: This oracle models the adversary's ability to block or modify the message m that is sent from R to T (respectively T to R) in session i of a truthful protocol execution. It outputs the response r from T (R).

- O$^{Reveal}(K)$: This oracle models the adversary's ability to expose the tag's permanent information. There is no point calling the reveal oracle on the same tag more than one time.

- O$^{Test}(R,T,i)$: This oracle models the untraceability test. When this query is invoked for session i, a random bit $b \in \{0, 1\}$ is generated, and then A is given $T_b \in \{T_0,T_1\}$. Finally, A wins if it can correctly guess the bit b.

We define the adversary's advantage of successfully tracing the tag as $UPriv$.

The game played between an adversary A and all instances is divided into the following three phases. Forward untraceability is presented by the following game between the challenger (all instances in the RFID system) and a polynomial time adversary A. The adversary is able to distinguish between two different tags using above four oracles. The attack experiment consists of three steps: The learning phase, the challenge phase and the guess phase.

- **Learning phase**: A chooses two fresh tags (T_0,T_1), and R. A makes queries on T_0, T_1, and R using the O^{Send}, O^{Execute}, and O^{Reveal} oracles for many times of its choice. A continues calling $O^{Execute}$ queries on T_0 and T_1 in the round i. Finally, it obtains a transcript ψ_0 and ψ_1.

- **Challenge phase**: A sends an O^{Test} query on the set $\{T_0,T_1\}$. A selects a random bit $b \in \{0,1\}$, A is given a tag T_b from the set $\{T_0,T_1\}$. A continues calling any $O^{Execute}$ and O^{Send} queries. A makes queries on T_b and R by calling $O^{Execute}(T_b^{`})$ in the next round. Finally, it obtains a transcript $\psi^{`}$.

- **Guess phase**: If $\psi^{`} = \psi_0$, $A$ guesses $b = 0$ and decides $T_b^{`} = T_0$, otherwise guesses $T_b^{`} = T_1$.

$$Adv_A^{UPriv}(k) = \left| pr(T_b^{'} = T_b) - \frac{1}{2} \right|$$

$$0 \le Adv_A^{UPriv} \le \frac{1}{2}$$

The protocol is untraceable if the adversary's advantage is said to be negligible and converges to zero using the security parameter k.

2.4 The traceability attack for the W2 scheme

We point out the disadvantage in location privacy of the W2 scheme, namely traceability attack. The traceability attack exists when the update procedure begins in the *Case* 1 of the W2 scheme. The *Case* 1 involves two message flows: the flow (r_R, IDR^-) of sending by the server and the flow of replying (r_R, M_T, x) by the tag. But when the protocol turns into the update procedure, *Case* 2 or *Case* 3, it takes a third flow sended by the server for the update. Therefore, an adversary can make precise area location easier by analyzing this case. The formal analysis is modeled in the following three phases.

- **Learning phase**: For a selected tag with pseudonym T_i, A sends some random nonce r to T_i by calling O^{Send} (r,T_i). Then A repeats the previous step until T_i response contains $SecReq$, demanding a secret update request and drives T_0 into *Case* 3 (c = 0) in *Case* 1. A continues calling $O^{Execute}$ queries on T_0 and T_1 in the round i. Finally, it obtains a transcript ψ_0 with two flows $\{(r_R,IDR^-),\ (r_R,M_T,x)\}$ or ψ_1 with three flows $\{(r_R$, $IDR^+)$, $(r_R$, $M_1)$, $(r_R$,M_{s1},$M_{s2})\}$

- **Challenge phase**: A sends an $O^{Test}(T_b^{`})$ query on the set $\{T_0,T_1\}$. $A$ selects a random bit $b \in \{0,1\}$ and gives a tag $T_b$ from the set $T_0,T_1$ in session $i$+1 of the protocol. Then $A$ continues calling $O^{Execute}(T_b^{`})$ and $O^{Reveal}(T_b^{`})$ queries on $T_b$ and $R$ in the round $i+1$. Finally, it obtains a transcript $\psi^{`}$.

- **Guess phase**: If $\psi^{`} = \psi_0$, $A$ guesses $b = 0$ and decides $T_b^{`} = T_0$, otherwise guesses $T_b^{`} = T_1$.

As a result,

$$Adv_A^{UPriv}(k) = \left| pr(T_b^{`} = T_b) - \frac{1}{2} \right| = \left| 1 - \frac{1}{2} \right| = \frac{1}{2}$$

Proof: It is obvious that the executed round between DB and T_b has a transcript with three flows. Also, an adversary has the ability to put a tag T_b into *Case* 2 or

Case 3. From the Learning phase, A puts the state of T_1 in the beginning of *Case* 1, because it's related transcript involves two information flows at this time. Therefore, $Adv_A^{UPriv}(k)$=1.

3 The proposed scheme

The different delegated readers receiving the same messages in the W0 scheme may result in some potential problems: if one of the delegated readers is compromised, it is difficult for the authority to find which reader is untrustworthy.

Each of the offline reader must have a unique IDR^- , different secrets and pseudonyms. Moreover, the δ calculation formulas contain delegated reader IDR^- used in the tag's response. For a delegation protocol, it is important to distinguish whether its corresponding entity is a legitimate tag or another delegated reader. The revised protocol differentiate between two entities by embedding a unique secret δ, which is shared by the server and the tag but not by the reader.

The procedure of the authentication phase is depicted in Fig. 2.

The proposed scheme involves two different phases. The first phase is set up phase. *DB* randomly selects a key k and values m and x_0 and computes identifiers x_i as $x_i = e_k(x_{i-1})$ for $1 \le i \le m$ for each tag. *DB* generates a random string r_R, IDR^- and $\delta = d_z(IDR^- \| k)$. Then, *DB* sends r_R, IDR^- and δ to the offline reader R^-.

The second phase is authentication phase. This phase is divined in two different cases. *DB* performs the tag identification in constant for a normal *case* 1 using *Req*1 and accomplishes a linear search in an abnormal *case* 2 using *Req*2.

The procedure of the authentication phase is depicted as follows.

Case 1 **(Delegation Request)**

1. Upon reception of r_R and IDR^-, the tag checks whether c equals to zero. If not, then the following steps are performed:

(a) Then it allows the tag to use hash functionality and calculates $\delta = d_z(IDR^- \| k)$, $M_T = f_k(r_R \| x \| \delta)$ and updates x to $e_k(x \| z)$ and c to $c-1$. Then it transmits (r_R, M_T, x) to the offline reader.

(b) Upon reception of $(M_T, x, r_R, Req1)$, *DB* performs the following steps.

i. Upon reception of *Req*1, *DB* uses the received x to quickly search for x_i and identify the tag in constant time.

ii. Checks that $f_k(r_R \| x_{i-1} \| \delta)$ equals to the received M_T, where k and δ are the key belonging to the identified tag. If the equation is verified, *DB* authenticated the tag. Next, it reduces the corresponding tags' identifiers.

iii. Calculates $M_s = g_z(k \| r_R \| M_T)$ and transmits (r_R, M_s) to the tag.

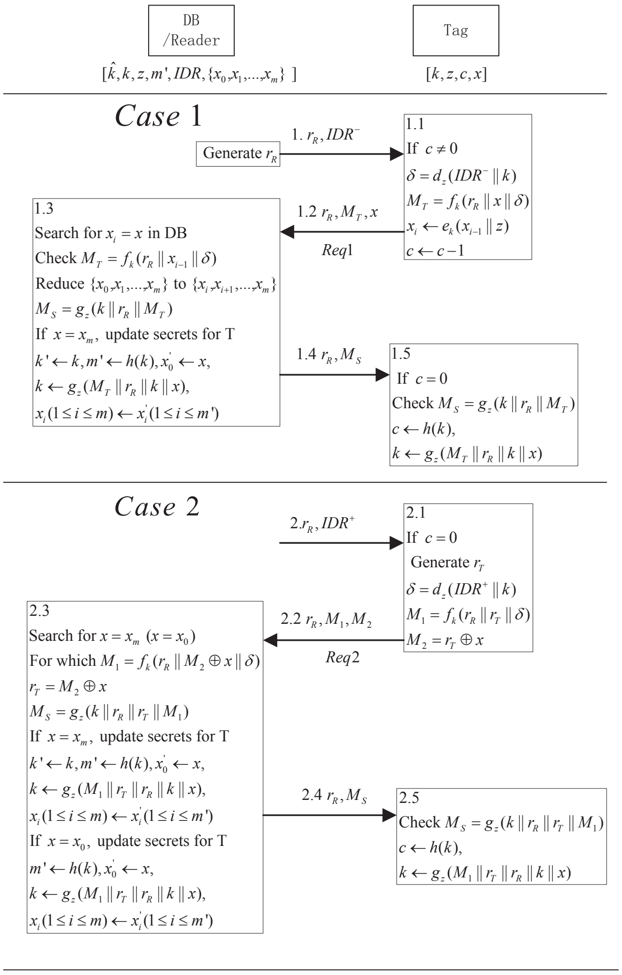
DB
/Reader
$[\hat{k}, k, z, m', IDR, \{x_0, x_1, \ldots, x_m\}\ ]$
Tag
$[k, z, c, x]$
Case 1
Generate r_R
1. r_R, IDR^-
1.1
If $c \ne 0$
$\delta = d_z(IDR^- \| k)$
$M_T = f_k(r_R \| x \| \delta)$
$x_i \leftarrow e_k(x_{i-1} \| z)$
$c \leftarrow c-1$
1.2 r_R, M_T, x
*Req*1
1.3
Search for $x_i = x$ in DB
Check $M_T = f_k(r_R \| x_{i-1} \| \delta)$
Reduce $\{x_0, x_1, \ldots, x_m\}$ to $\{x_i, x_{i+1}, \ldots, x_m\}$
$M_S = g_z(k \| r_R \| M_T)$
If $x = x_m$, update secrets for T
$k' \leftarrow k, m' \leftarrow h(k), x_0' \leftarrow x,$
$k \leftarrow g_z(M_T \| r_R \| k \| x),$
$x_i(1 \le i \le m) \leftarrow x_i'(1 \le i \le m')$
1.4 r_R, M_S
1.5
If $c = 0$
Check $M_S = g_z(k \| r_R \| M_T)$
$c \leftarrow h(k),$
$k \leftarrow g_z(M_T \| r_R \| k \| x)$
Case 2
2.r_R, IDR^+
2.1
If $c = 0$
Generate r_T
$\delta = d_z(IDR^+ \| k)$
$M_1 = f_k(r_R \| r_T \| \delta)$
$M_2 = r_T \oplus x$
2.2 r_R, M_1, M_2
*Req*2
2.3
Search for $x = x_m$ $(x = x_0)$
For which $M_1 = f_k(r_R \| M_2 \oplus x \| \delta)$
$r_T = M_2 \oplus x$
$M_S = g_z(k \| r_R \| r_T \| M_1)$
If $x = x_m$, update secrets for T
$k' \leftarrow k, m' \leftarrow h(k), x_0' \leftarrow x,$
$k \leftarrow g_z(M_1 \| r_T \| r_R \| k \| x),$
$x_i(1 \le i \le m) \leftarrow x_i'(1 \le i \le m')$
If $x = x_0$, update secrets for T
$m' \leftarrow h(k), x_0' \leftarrow x,$
$k \leftarrow g_z(M_1 \| r_T \| r_R \| k \| x),$
$x_i(1 \le i \le m) \leftarrow x_i'(1 \le i \le m')$
2.4 r_R, M_S
2.5
Check $M_S = g_z(k \| r_R \| r_T \| M_1)$
$c \leftarrow h(k),$
$k \leftarrow g_z(M_1 \| r_T \| r_R \| k \| x)$

Fig. 2: The proposed scheme.

iv. If $x = x_m$, the tag's secrets are renewed: $k' \leftarrow k$, $k \leftarrow g_z(M_T \| r_R \| k \| x)$, $m' \leftarrow h(k)$, $x_0' \leftarrow x$, $x_i(1 \le i \le m) \leftarrow x_i'(1 \le i \le m')$ where $x_i' = e_k'(x_{i-1}' \| z)$.

(c) Upon reception of (r_R, M_s), the tag checks whether c equals to zero.

i. If $c = 0$, T checks $M_s = g_z(k \| r_R \| M_T)$ using k, r_R and M_T. If T authenticates *DB* successfully, then its secrets are renewed: $k \leftarrow g_z(M_T \| r_R \| k \| x), c \leftarrow h(k)$.

ii. If $c \ne 0$, suspends the protocol.

Case 2 **(Delegation Update)**

2. Upon reception of r_R and IDR^+, the tag produces a random nonce r_T and checks whether c equals to zero. If yes, a step by step description is also given below:

(a) The tag calculates the messages $\delta = d_z(IDR^+ \| k)$, $M_1 = f_k(r_R \| r_T \| \delta)$ and $M_2 = r_T \oplus x$. Then it transmits $(r_R, M_1, M_2, Req2)$ to *DB*.

(b) Upon reception of M_1, M_2 and *Req*2, *DB* implements the following steps.

i. Searches for a value $x = x_m$ or $x = x_0$ for which $M_1 = f_k(r_R \| M_2 \oplus x \| \delta)$. Then *DB* finds the match and authenticates the tag.

ii. Calculates $M_s = g_z(k \| r_R \| r_T \| M_1)$ and transmits (r_R, M_s) to the tag.

iii. If $x = x_m$, then the secrets are renewed: $k' \leftarrow k$, $k \leftarrow g_z(M_1 \| r_T \| r_R \| k \| x), m^{'} \leftarrow h(k), x_0^{'} \leftarrow \ x, x_i(1 \leq i \leq m) \leftarrow x_i^{'}(1 \leq i \leq m^{'})$ where $x_i^{'} = e_k^{'}(x_{i-1}^{'} \| z)$.

iv. If $x = x_0$, its secrets are renewed:

$k \leftarrow \ g_z(M_1 \| r_T \| r_R \| k \| x), \ m \leftarrow \ h(k), x_0 \leftarrow \ x, \ x_i \ (1 \leq \ i \leq \ m) \leftarrow \ x_i \ (1 \leq \ i \leq \ m)$ where $x_i = e_k(x_{i-1} \| z)$.

(c) Upon reception of (r_R, M_s), the tag checks whether $M_s = g_z(k \| r_R \| r_T \| M_1)$ equals to the received M_s, then its secrets are renewed:

$k \leftarrow \ g_z(M_1 \| r_T \| r_R \| k \| x), c \leftarrow \ h(k)$.

4 Security analysis and performance analysis

Before giving security analysis of our revised protocol, we want to stress three critical distinctions between our protocol and other schemes.

(a) Notice that compared with the original W0 protocol, our proposed protocol needs one less execution of the message computation from the reader, while the computational complexity of message M_s becomes simple for the delegated reader.

On the other side, our revision uses the same secret update procedure as that in the W2 scheme except that M_{s1} is replaced by M_s, and instead of sending messages M_{s1} and M_{s2} in the last step, message M_s is transmitted. The reader do not perform a bitwise XOR operation of the data M_s, thus the computational complexity is reduced. Consequently, on the tag side, T need not verify the integrity of the message M_s.

(b) A solution to reducing the maximum complexity of the search can be that add two different flags to the tags response in order to distinguish delegation request from delegation update. The tag then sends r_R, M_T, and x back to DB with a request $Req1$ for an normal update state in case1.

On the other hand, two flags of the requests are slightly different if the tag is in a rare abnormal state as is in *case* 2 when DB receives the request $Req2$.

In the W0 and W2 protocol, such a flag *SecReq* gives a hint to an adversary to distinguish the tag's state from the different cases.

In [3], the Erguler's theorem demonstrates an impossibility in achieving any form of untraceability as long as the attacker decides its unbalanced states by checking whether the tag response includes *SecReq* message or not.

In the W1 protocol, the messages $\{r_R, M_T, x\}$ of *case* 1 and $\{r_R, M_1, M_2\}$ of *case* 2 are similar, DB can not distinguish and do not know how to perform the following proceedings.

(c) The number of exchanged flows during different cases of our revision is same. It is impossible for the attacker to identify and trace the state of the target tag.

A. Security analysis

Although the W2 scheme is vulnerable to traceability attack due to different numbers of message flows, this problem has been resolved in the revised protocol. To repel the cited attack, there are three information flows in both cases of the revised scheme.

Table 2: security properties comparisons

Attack types	W1	W2	Ours
Traceability attack	N	N	Y
Tag impersonation attack	N	Y	Y
Desynchronization attack	N	Y	Y

(a) **Resistance to impersonation attack**

If we analyse the revised scheme, it can be seen that an authority can distinguish compromised entity from all readers by using the previously described reader identifiers IDR^- and δ. The reason is that only the registered tag can compute M_1 which directly depends on freshly generated r_T and δ values.

Therefore, a unique secret z can be used to differentiate between tags and other delegated readers in order to protect against tag impersonation attack. Moreover, the reader identifier should be taken as an input in computing the tag's response message in order to find quickly which reader is untrustworthy for the authority.

(b) **Resistance to DoS attack**

The number of tag is scalable when the reader verify the tag. The revised scheme has the advantage of controlled delegation with embedded couter both on the tag and the reader, whereas we prefer to use m on reader's side to protect against DoS attack.

(c) **Resistance to tracking attack**

Although the W1 and W2 protocols might hide the tags' identifiers from adversaries,it is not difficult for an attacker to decide the state of case using eavesdropped communications.

The proposed protocol protects against the tag tracking attack because of the use of security parameter δ appropriately. Hence, an adversary can not convince the reader to believe that the fake tags are legitimate, even if he knows the secrets.

Hence, when an adversary tries to drop the target tag in the case $c = 0$ by querying it more than m times in the Execute phase, it has no way to trace the past rounds of the target tag. Therefore, the revision provides forward untraceability.

Furthermore, the backward traceability attack is impossible with respect to the existence of the look-up table which keeps corresponding hash values of the records x.

In addition, the message structures in the revised scheme are similar to the original W2 scheme that make the proposed protocol resistant to other attacks according to the presented proofs.

Table 2 summarizes the security properties comparisons, where Y denotes that the protocol resists such an attack and N denotes that the protocol does not protect against such an attack.

B. $Performance$ analysis

(a) **Computational aspect**

Regarding the computation on server's side, Song's scheme takes four hash operations, and Habibi's scheme takes six while our scheme takes five.

The complexity of this search can also be globally reduced because the look-up table keeps the corresponding hash value of each tag's identifier *x*.

(b) **Scalability**

The revised protocol takes $O(1)$ work to authenticate a tag in *case* 1 and implements $O(n)$ work to update the secrets of the tag in *case* 2. With this mechanism, it is possible to achieve minimum complexity of $O(1)$ for tag identification.

(c) **Communication**

Furthermore, our scheme verifies the message integrity M_S and mitigates the computational load of the server by reducing the number of exchanged message in *step* 2 of *case* 2, while there are three messages in the W2 protocol under identical conditions.

The above performance comparisons show that our scheme has a reasonable execution efficiency, which is superior to the W0, W1 and W2 protocol. However, the computation of low-cost tags can not afford more hash operations.

5 Conclusions and further work

This article has demonstrated the security weaknesses of previous schemes. Additionally, we propose SRDP scheme to enhance the security and protect against tag tracing attacks by adding only one more hash function. The delegation of the reader is controlled depending on the counter value c and m' respectively stored in the tag and reader. As a consequence, an adversary can not impersonate a tag. Our protocol can distinguish malicious readers from all other readers, because different offline readers have different rights and IDR^{-}.

Therefore, we conclude that the security requirements can be met by adding two different flags to distinguish two different cases. Furthermore, our future work is to improve the schemes without any more hash functions, because the computation of low cost tags can not afford more hash operations.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (Grant No. 61202478 and Grant No. 61303263).

References

[1] B. Alomair, A. Clark, J. Cuellar, R. Poovendran. Scalable RFID Systems: A Privacy-Preserving Protocol with Constant-Time Identification. IEEE Transactions on Parallel and Distributed Systems, **23**, 677-691 (2012).

[2] C. F. Lee, H.Y. Chien, C.S. Laih. Server-less RFID authentication and searching protocol with enhanced security. International Journal of Communication Systems, **25**, 376-385 (2012).

[3] I. Erguler, E. Anarim, G. Saldamli. Unbalanced states violates RFID privacy. Journal of Intelligent Manufacturing, 1-9 (2012).

[4] S. Piramuthu. Vulnerabilities of RFID protocols proposed in ISF. Information Systems Frontiers, **14**, 647-651 (2012).

[5] M. Burrows, M. Abadi and R. M. Needham, A Logic of Authentication, Proc. R. Soc., **426**, 233-271 (1989).

[6] Hung-Min Sun; Wei-Chih Ting; King-Hang Wang; On the Security of Chien's Ultralightweight RFID Authentication Protocol, Dependable and Secure Computing, IEEE Transactions on, **8**, 315-317 (2011).

[7] B. Song, C. J. Mitchell. Scalable RFID security protocols supporting tag ownership transfer. Journal of Computer Communications, **34**, 556-566 (2011).

[8] M. H. Habibi, M. R. Aref. Security and Privacy Analysis of SongCMitchell RFID Authentication Protocol. Journal of Wireless Personal Communications, **69**, 1583-1596 (2013).

[9] I. Erguler, E. Anarim. Security flaws in a recent RFID delegation protocol. Journal of Personal and Ubiquitous Computing, **16**, 337-349 (2012).

[10] M. Safkhani, N. Bagheri, P. Peris-Lopez, et al. On the traceability of tags in SUAP RFID authentication protocols. proceedings of the RFID-Technologies and Applications (RFID-TA), 2012 IEEE International Conference on, IEEE, (2012).

[11] I. Erguler, E. Anarim. Scalability and security conflict for RFID authentication protocols [J]. Wireless Personal Communications, **59**, 43-56 (2011).

[12] K. Osaka, T. Takagi, K. Yamazaki, et al. An efficient and secure RFID security method with ownership transfer [M]. RFID Security. Springer, 147-176 (2009).

[13] R. C-W Phan,J. Wu , K. Ouafi, et al. Privacy analysis of forward and backward untraceable RFID authentication schemes [J]. Wireless Personal Communications, **61**, 69-81 (2011).

[14] M. Sharaf. RFID Mutual Authentication and Secret Update Protocol for Low-Cost Tags. proceedings of the Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on, IEEE (2012).

[15] W. Stevens, TCP /IP Illustrated, Volume 1: The Protocols,Addison-Wesley, ISBN 0-201-63346-9, (1994).

[16] D. Molnar, A. Soppera, and D. Wagner. A scalable, delegatable pseudonym protocol enabling ownership transfer of RFID tags. In Selected Areas in Cryptography, Springer, 276-290 (2006).

[17] A. Fernndez-Mir, R. Trujillo-Rasua, J. Castella-Roca, et al. A scalable RFID authentication protocol supporting ownership transfer and controlled delegation [M]. RFID Security and Privacy. Springer, 147-162 (2012).

[18] G. Avoine, C. Lauradoux, T. Martin. When compromised readers meet RFID. In: Workshop on information security applicationsłWISA09. Lecture Notes in Computer Science, Springer-Verlag, 36-50 (2009).

[19] G. Avoine, X. Carpent, B. Martin. Privacy-friendly synchronized ultralightweight authentication protocols in the storm [J]. Journal of Network and Computer Applications, **35**, 826-843 (2012).

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 applied computer Technology from the China University of Mining and Technology. Her research interests include security protocols and network security.

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.

Yu Guo received his bachelor's degree from the China University of Mining and Technology. He has been a master degree candidate in materials science and Engineering from the China University of Mining and Technology. His research interests include fiber-reinforced composites and numerical simulation.