An Efficient RFID Authentication Protocol Supporting Tag Ownership Transfer

1He Jialiang,2Ouyang Dantong,3Xu Youjun,4Shang Yu
11College of Computer Science and Technology, Jilin University
2Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University
E-mail: urchin2012@sina.com
22College of Computer Science and Technology, Jilin University
2Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University
E-mail: ouyangdantong@163.com
31College of Computer Science and Information Technology, Daqing Normal University
E-mail: xu_youjun@sohu.com
41College of Computer Science and Technology, Jilin University
2Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University
E-mail: acm_icpc@qq.com

Abstract
Except common security and privacy requirements, well scalability and supporting tag ownership transfer are important requests for RFID systems. In this paper, an efficient RFID mutual authentication protocol supporting tag ownership transfer is proposed, this protocol only requires O(1) work to identify and authenticate a tag in the backend server and is suitable for the low-cost RFID systems. The security and performance of the proposed protocol are analyzed as well.

Keywords: RFID; Authentication Protocol; Tag Ownership Transfer

1. Introduction
Radio Frequency Identification (RFID) is a technology which is used to identify remote objects embedded with RFID tags by wireless scanning without manual intervention[1]. It has been used in various real-life applications such as supply chain management, security monitoring, e-payment system, citizen management, medical management. Because of gradually descending cost and more advantages, its use will increase with high speed in the future.

RFID technology can provide so many benefits for automatically identifying object items, however, based on special physical character such as wireless communication, signal broadcasting, there are many security and privacy problems in the existing RFID systems. However, due to strictly limited calculation resources, small storage capacity and faint power supply of low-cost tags, it is difficult to apply an ordinary and complicated but safe cryptographic algorithm to an RFID system and these factors are hindering the rapid spread of this technology[2]. So designing an efficient and low-cost RFID security scheme is a difficult and challenging object, it has attracted many researchers embarking on this task.

Except common security and privacy requirements, scalability is a desirable property in almost any system, enabling it to handle growing amounts of work in a graceful manner[3]. A scalable RFID system should have the ability of handling large numbers of tags without undue strain. However, for most of the proposed RFID authentication protocols, the backend server must perform a linear search(record by record lightweight encryption calculation and comparation) of its database to identify and authenticate a tag, such a linear search runs in O(n) time(n is the number of elements in the backend database). Such a costly search function will potentially cause scalability issues as the tag population increases[4], so how to reduce time complexity of performing lightweight encryption calculation in the backend server becomes a significant object for the researchers.
Another requirement for RFID systems is secure tag ownership transfer. In some real-life applications, the owner of a RFID tag may change a number of times during its lifetime. Ownership transfer means that the server of the new owner takes over tag authorization, and so needs to be given the necessary private information to securely interact with and identify the tag. Thus all information associated with the tag will need to be passed from the old to the new owner[4]. Such transfer should meet requirement as follows, after tag ownership transfer, the older owner cannot read the tag using remained tag information so as to trace the new owner’s behaviors, while the new owner cannot trace the old owner’s past behaviors by using received tag information.

Especially, there is no universal applicable solution for RFID systems, the security levels depend on specific application. That is to say, RFID systems should attain balanced target between security and performance. The research object of RFID authentication protocols is using fewer resources to satisfy the same security requirements or achieving higher security levels by using the equal resources[5].

The main contribution of this paper is to propose an efficient RFID mutual authentication protocol supporting tag ownership transfer. In addition, this protocol only requires O(1) work to identify and authenticate a tag in the backend server and is suitable for the low-cost RFID systems because only one-way hash function, XOR operation and concatenation operation are needed in tags.

The rest of this paper is organized as follows. In the second section, the related research work is introduced; in the third section, a new efficient RFID authentication protocol supporting tag ownership transfer is proposed; in the fourth section, security and performance properties of the proposed protocol are analyzed; finally, the conclusion of this paper is generalized in the fifth section.

2. Related work

Presently, lightweight encryption methods such as Hash, PRNG(Pseudo Random Number Generator) and CRC(Cyclic Redundancy Code) are used wildly in design of RFID protocols. Especially, for achieving the balance between security and performance. Hash-based methods have been researched and used actively. For the aspect of security, a qualified RFID authentication protocol should meet common security requirements such as resisting spoofing attack, replay attack, DoS attack and meeting tag untraceability, tag information protection, forward security, and backward security. For the aspect of performance, because constraint limited resources on tag side, so storage cost, computation cost, traffic cost of a tag in a qualified RFID authentication protocol should be minimally controlled as minimally as possible.

According to whether the backend server and a tag update the identifier or not in an authentication access, RFID authentication protocols are divided into dynamic ID mechanism and static ID mechanism. Dynamic ID mechanism is always used in the circumstance that the ownership of ID is needed to transfer. Our research work in this paper is based on dynamic ID mechanism. Many authentication protocols based on dynamic ID mechanism have been proposed recently[4,6-12].

In 2010, Zhou et al. proposed a lightweight anti-desynchronization RFID authentication protocol[12], Let’s introduce this authentication protocol detailedly as follows:

An Efficient RFID Authentication Protocol Supporting Tag Ownership Transfer
He Jialiang, Ouyang Dantong, Xu Youjun, Shang Yu
Table 1. The notations used in [12]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>The tag’s unique serial number</td>
</tr>
<tr>
<td>IDS</td>
<td>The tags’ unique index-pseudonym</td>
</tr>
<tr>
<td>Data</td>
<td>Information of the corresponding tag</td>
</tr>
<tr>
<td>H()</td>
<td>An hash function, H: {0,1}^l \rightarrow {0,1}^l</td>
</tr>
<tr>
<td>PRNG()</td>
<td>The pseudo random number generator</td>
</tr>
<tr>
<td>XOR</td>
<td>XOR operator</td>
</tr>
<tr>
<td>M_L</td>
<td>The left part of the message M</td>
</tr>
<tr>
<td>M_R</td>
<td>The right part of the message M</td>
</tr>
<tr>
<td>r</td>
<td>The random number generated by the reader</td>
</tr>
<tr>
<td>R</td>
<td>The random number generated by the backend server</td>
</tr>
<tr>
<td>Pre-x</td>
<td>The previous value of x</td>
</tr>
<tr>
<td>Cur-x</td>
<td>The current value of x</td>
</tr>
<tr>
<td>x_i</td>
<td>The x value in the (i)th session of the protocol</td>
</tr>
<tr>
<td>T</td>
<td>Temporary value</td>
</tr>
<tr>
<td>A→B:M</td>
<td>A sends message M to B</td>
</tr>
</tbody>
</table>

Figure 1. The protocol proposed in [12]

The (i+1)th authentication access is shown in Figure 1, and detailed process as follows:
Phase 1: The reader generates a random number r and query tags with r.
Phase 2: After receiving r, a tag calculates \(M = H(T_i \oplus r)\) and \(P = H(\text{key}_i \oplus r \oplus M \oplus C)\), then sends IDS, M and P_l back to the reader, subsequently the tag should calculate \(T_{i+1} = H(T_i \oplus \text{PRNG})\) and save \(T_{i+1}\) in its memory. Especially, this protocol uses \(H(T_i \oplus r)\) to substitute pseudo random number of the tag.
Phase 3: After receiving IDS, M and P_l from the tag, the reader would send r, IDS, M and P_l to the backend server.
Phase 4: After receiving authentication message from the reader, the backend server would search whether there exists certain IDS* in the column ‘Cur-IDS’ of the database, which could make \(H(\text{Cur-key} \oplus r \oplus M \oplus C) = P_l\). If there exists such record, the tag would be considered as a legitimate tag, then the backend server would generate a random number R and calculate \(N = H(\text{key}_i \oplus R \oplus M)\), subsequently send R, N_R and Data* to the reader, then the backend server should calculate \(\text{key}_{i+1} = \text{key}_i \oplus N_R\) and update Pre-key = Cur-key, Cur-key = \(\text{key}_{i+1}\), Pre-IDS = Cur-IDS, Cur-IDS = \(H(\text{key}_{i+1})\).
An Efficient RFID Authentication Protocol Supporting Tag Ownership Transfer
He Jialiang, Ouyang Dantong, Xu Youjun, Shang Yu

If there not exists certain IDS* in the column ‘Cur-IDS’ of the database, which could make H(Cur-key* ⊕ r ⊕ M ⊕ C)* = P_L, the backend server should search whether there exists certain IDS* in column ‘Pre-IDS’ of the database, which could make H(Pre-key* ⊕ r ⊕ M ⊕ ID*) = P_L. If there exists such record, the tag would be considered as a legitimate tag, but in the last authentication access, the tag has not updated its key and IDS successfully for some reason, so the backend server should generate a random number R and calculate N = H(key_i ⊕ R ⊕ M), then send R, N_R and Data* to the reader, then the backend server should calculate key_{i+1} = key_i ⊕ N_L, and update Cur-key = key_{i+1}, Cur-IDS = H(key_{i+1}) but Pre-key and Pre-IDS would keep unaltered.

If there not exists certain IDS* in column ‘Cur-IDS’ and column ‘Pre-IDS’ of the database like that, the tag would be considered as an illegitimate tag, the authentication is failed, failure information would be sent to the reader.

Phase5: After receiving R, N_R and Data* from the backend server, the reader would store Data* in its memory and send R, N_R to the tag. After receiving R, N_R from the tag, the reader would calculate H(key_r ⊕ R ⊕ M). If the right part of calculation outcome equals to the received N_R, then the object of mutual authentication achieves, the tag should update key_{i+1} = key_i ⊕ N_L and IDS_{i+1} = H(key_{i+1}), otherwise, the authentication is failed.

This protocol is an efficient and clear RFID authentication protocol, which only requires O(1) work to identify and authenticate a tag in the backend server by using RFID pseudonym scheme. However, there are some shortcomings of security and performance as follows:

(1) It does not meet tag untraceability. An adversary could recognize and locate a tag as follows, in the phase2, the adversary could eavesdrop the responding message IDS_m, M and P_L through the backward channel, in the phase5, the adversary could intercept the message R, N_R through the forward channel, so the tag could not update key_{i+1} = key_i ⊕ N_L and IDS_{i+1} = H(key_{i+1}) successfully in this access. Then the adversary could impersonate the legal reader and generate a random number r’ and query tags with r’, the tag would respond this query for its inherent nature and calculate M’ = H(T_i ⊕ r'), P’ = H(key_i ⊕ r' ⊕ M ⊕ C), then respond the query with IDS_m, M’ and P’ through the backward channel, because the adversary has acquired IDS in the last authentication access, so the adversary could compare IDS, with the responding message from tags, if the message from any tag comprise IDS, then the tag would be recognized and located, so this protocol does not meet tag untraceability that it has claimed. Tag untraceability is the most important requirement for a RFID system, so this protocol is not a qualified authentication protocol for this reason.

(2) Because this protocol is based on pseudonym scheme, so the pseudonym of each tag should be uniquely. In the process of generating IDS_{i+1} in the backend server, this protocol does not check new IDS_{i+1} whether equals to some value in column ‘Cur-IDS’ and column ‘Pre-IDS’. If the value has existed in column ‘Cur-IDS’ or column ‘Pre-IDS’, new IDS_{i+1} being stored in the database without checking would lead to intending confusion.

(3) RFID tag has constraint requirements of limited resources, storage cost, computation cost, and traffic cost of each tag should be controlled as minimally as possible. This protocol stores key in a tag, it not only wastes storage resource of the tag but also leads to potential security risk. In addition, in the phase2, this protocol uses H(T_i ⊕ r) to substitute pseudo random number of a tag, and calculates T_{i+1} = H(T_i ⊕ P_L) and saves T_{i+1} in the tag’s memory, as we know, if an adversary acquires H(), T_{i+1} would be calculated by the adversary easily, so the operation H() is not necessary for the calculation of T_{i+1}, it would increase computation cost of the tag.

In 2010, Luo et al. proposed a RFID security protocol based on dynamic ID mechanism[11], which could support tag ownership transfer, however, there two problems in the protocol as follow:

(1) In the protocol, the backend server must perform the exhaustive linear search (record-by-record hash function calculation) to find the matching item. In each authentication access, the backend server averagely performs (n+1)/2 times hash operations to verify and authenticate a tag, such a linear search runs in O(n) time, where n is the number of elements in the database. Such a costly search function would potentially cause scalability issues as the amount of tags increases enormously.

(2) The protocol needs seven rounds to perform a whole verification and secret update process for each tag, it would not increase security of this protocol but lead to extra traffic cost.

Based on these two protocols, we proposed a new efficient RFID authentication protocol as follows:
3. A new efficient RFID authentication protocol supporting tag ownership transfer

3.1 Notation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>The unique index code of a tag (The length is (l))</td>
</tr>
<tr>
<td>IDS</td>
<td>The tags’ unique index-pseudonym (The length is (l))</td>
</tr>
<tr>
<td>Info</td>
<td>Information of the corresponding tag stored in the backend server</td>
</tr>
<tr>
<td>H()</td>
<td>An one-way hash function, (H : {0,1}^* \rightarrow {0,1}^l) (The length of output is (l))</td>
</tr>
<tr>
<td>E(_k)((\cdot))</td>
<td>Symmetry encryption function (The length of output is (l))</td>
</tr>
<tr>
<td>PRNG()</td>
<td>The pseudo random number generator (The length of output is (l_k), usually (l_k &lt; l))</td>
</tr>
<tr>
<td>R</td>
<td>The random number generated by the reader (The length is (l_R))</td>
</tr>
<tr>
<td>T</td>
<td>Temporary value (The length is (l))</td>
</tr>
<tr>
<td>F</td>
<td>Failure information of authentication</td>
</tr>
<tr>
<td>Pre-(x)</td>
<td>The previous value of (x)</td>
</tr>
<tr>
<td>Cur-(x)</td>
<td>The current value of (x)</td>
</tr>
<tr>
<td>(x_i)</td>
<td>The (x) value in the (i)th session of this protocol</td>
</tr>
<tr>
<td>(A \rightarrow B : M)</td>
<td>A sends message (M) to (B)</td>
</tr>
</tbody>
</table>

3.2 Initialization stage

The backend server and tags store information that required performing authentication. The backend server should generate a random secret key \(k_j\) and calculate \(E_{kj}(ID_j)\) for each tag, it initially stores \(ID_j, Info_j, Pre-key_j\) (initial value is ‘0’), \(Cur-key_j\) (initial value is \(k_j\)), \(Pre-IDS_j\) (initial value is ‘0’), \(Cur-IDS_j\) (initial value is \(E_{kj}(ID_j)\)) of each tag in its database. \(ID_j, IDS\) (initial value is \(E_{kj}(ID_j)\)) and \(T_0\) (initial value may be ‘0’) would be set in the corresponding tag.

3.3 The \((i+1)\)th authentication access

The new proposed protocol is shown in Figure 2:
The \((i+1)\)th authentication access as follows:

**Phase 1:** The reader generates a random number \(R\) and query tags with \(R\).

**Phase 2:** After receiving \(R\), a tag calculates \(M = H(T_i \oplus R)\) and \(P = H(M_L \|| R) \oplus IDS_i\), then sends \(H(M_L \|| R) \oplus IDS_i\) and \(M_L\) back to the reader, subsequently the tag should calculate \(T_{i+1} = M \oplus P\) and save \(T_{i+1}\) in its memory.

**Phase 3:** After receiving \(H(M_L \|| R) \oplus IDS_i\) and \(M_L\) from the tag, the reader should send \(R, H(M_L \|| R) \oplus IDS_i\) and \(M_L\) to the backend server.

**Phase 4:** After receiving authentication message from the reader, the backend server would calculate \(IDS = H(M_L \|| R) \oplus P\), and search whether there exists certain \(IDS^*\) in column ‘Cur-IDS’ of the database, which could make \(IDS = IDS^*\). If there exists such record, the tag should be considered as a legitimate tag, then the backend server could calculate \(key_{i+1} = Key_{i+1}(R, || M_L)\) and \(IDS_{i+1} = E_{key_{i+1}}(ID^*)\). In particular, after accomplishing these two calculations, the backend server must search whether there exists \(IDS^*\) in column ‘Pre-IDS’ and column ‘Cur-IDS’, which could make \(IDS^* = E_{key_{i+1}}(ID^*)\). If there exists such record, the backend server would calculate \(H(L(IDS_{i+1} || IDS_i)) \oplus \text{N} = H(M_L \|| R) || IDS_i \|| ID_{Si+1}\), then send Info*, \(H(L(IDS_{i+1} || IDS_i)) \oplus \text{N}\) to the reader and update \(Pre-key, Pre-IDS, Cur-key = key_{i+1}, \text{Pre-IDS} = Cur-IDS, Cur-IDS = IDS_{i+1}\); if there exists such record, the backend server must generate such a random key’ that could make the value which equals to \(E_{key_{i+1}}(ID^*)\) could not be found in column ‘Pre-IDS’ and column ‘Cur-IDS’, then the backend server would calculate \(H(L(IDS_{i+1} || IDS_i)) \oplus \text{N} = H(M_L \|| R) || IDS_i \|| ID_{Si+1}\), then send Info*, \(H(L(IDS_{i+1} || IDS_i)) \oplus \text{N}\) to the reader and update \(Pre-key, Pre-IDS, Cur-key = key_{i+1}, Cur-IDS = ID_{Si+1}\); if there exists such record, the backend server would calculate \(H(L(IDS_{i+1} || IDS_i)) \oplus \text{N} = H(M_L \|| R) || IDS_i \|| ID_{Si+1}\), then send Info*, \(H(L(IDS_{i+1} || IDS_i)) \oplus \text{N}\) to the reader and update \(Pre-key, Pre-IDS, Cur-key = key_{i+1}, Cur-IDS = E_{key_{i+1}}(ID^*)\).

If there does not exist certain \(IDS^*\) in column ‘Cur-IDS’ of the database, which could make \(IDS = IDS^*\). The backend server would search whether there exists certain \(IDS^*\) in column ‘Pre-IDS’ of the database, which could make \(IDS = IDS^*\). If there exists such record, the tag should be considered as a legitimate tag, but in the last authentication access, the tag has not updated key and IDS successfully for some reasons, so the backend server calculates \(key_{i+1} = Pre-key^* \oplus (R, || M_L)\) and \(IDS_{i+1} = E_{key_{i+1}}(ID^*)\). In particular, after accomplishing these two calculations, the backend server must search whether there exists \(IDS^*\) in column ‘Pre-IDS’ and column ‘Cur-IDS’, which could make \(IDS^* = E_{key_{i+1}}(ID^*)\). If there does not exist such record, the backend server would calculate \(H(L(IDS_{i+1} || IDS_i)) \oplus \text{N} = H(M_L \|| R) || IDS_i \|| ID_{Si+1}\), then send Info*, \(H(L(IDS_{i+1} || IDS_i)) \oplus \text{N}\) to the reader and update \(Cur-key = key_{i+1}, Cur-IDS = E_{key_{i+1}}(ID^*)\) but Pre-key and Pre-IDS would keep unaltered; if there exists such record, the backend server would calculate \(H(L(IDS_{i+1} || IDS_i)) \oplus \text{N} = H(M_L \|| R) || IDS_i \|| ID_{Si+1}\), then send Info*, \(H(L(IDS_{i+1} || IDS_i)) \oplus \text{N}\) to the reader and update \(Cur-key = key_{i+1}, Cur-IDS = E_{key_{i+1}}(ID^*)\) but Pre-key and Pre-IDS would keep unaltered.
server must generate such a random key’ that would make the value which equals to $E_{\text{key}}'(\text{ID}^*)$ could not be found in column ‘Pre-IDS’ and column ‘Cur-IDS’, then the backend server would calculate $H_i(\text{IDS}_{i+1} || \text{IDS}_i)$ and $N = H(M_i || R || \text{IDS}_i)$, and send Info*, $H_i(\text{IDS}_{i+1} || \text{IDS}_i)$, N to the reader, then update Cur-key = $\text{key}_{i+1} = \text{key'}$, Cur-IDS = $E_{\text{key}}'(\text{ID}^*)$ but Pre-key and Pre-IDS would keep unaltered.

If there not exists certain IDS* in column ‘Cur-key’ and column ‘Pre-key’ of the database, which could make $\text{IDS} = \text{IDS}^*$, the authentication is failed, F(failure information) would be sent to the reader.

Praiseworthy, in this phase, only one hash operation would be needed in verifying and authenticating a tag, so time complexity of hash function calculation achieves $O(1)$.

Phase5: After receiving Info*, $H_i(\text{IDS}_{i+1} || \text{IDS}_i)$, N from the backend server, the reader stores Info* in its memory and sends $H_i(\text{IDS}_{i+1} || \text{IDS}_i)$, N to the tag. After receiving $H_i(\text{IDS}_{i+1} || \text{IDS}_i)$, N from the reader, the tag would calculate $\text{IDS}'' = (M_i \oplus R \oplus \text{IDS}_i)$, then calculate $H(L(\text{IDS}'' || \text{IDS})$. If $H_i(\text{IDS}'' || \text{IDS})$ equals to received $H_i(\text{IDS}_{i+1} || \text{IDS}_i)$, then the object of mutual authentication achieves, the tag should update $\text{IDS}_{i+1} = \text{IDS}''$, otherwise, the authentication is failed.

3.4 The process of tag ownership transfer

Ownership transfer means that the server of the new owner takes over tag authorization, and securely interacts with and authenticates the tag instead of the server of the old owner. The process of tag ownership transfer as follows.

(1) The old owner of the tag should update $k(\text{Cur-key})$ to $k_1$, $\text{IDS}(\text{Cur-IDS})$ to $E_{k_1}(\text{ID})$ by executing an authentication access.

(2) The old owner sends (ID, Info, $k_1$) to the new owner in security channel, because the new owner may use different encryption function from $E_{k_1}(\text{ID})$, so $E_{k_1}(\text{ID})$ is not necessary being transmitted from the older owner to the new owner. In addition, this transmission is implemented between a backend server to another backend server usually, so security scheme can use traditional symmetry encryption method.

The older owner(A) should confer with the new owner(B) on symmetry secret key, A and B can perform security protocol like revised Needham-Schroeder protocol[13] to acquire symmetry secret key $K_{AB}$.

The old owner encrypts (ID, Info, $k_1$) using $K_{AB}$ and sends encryption information to the new owner.

The new owner decrypts encryption information using $K_{AB}$ and gets (ID, Info, $k_1$).

(3) The new owner should update $k_1(\text{Cur-key})$ to $k_2$ and corresponding pseudonym by executing an authentication access. As a result, the old owner is no longer able to read the information of the tag, the ownership of the tag is transferred from the old owner to the new owner successfully. So it would protect the privacy of the new owner from the old owner of the tag. On the other hand, the new owner cannot trace previous transactions between the old owner and the tag since it only knows the updated $k_1$, so it would protect the privacy of the old owner from the new owner of the tag.

4. Analysis

4.1 Security analysis

(1) Tag untraceability

An adversary can intercept the response message $(H(M_i \oplus R \oplus \text{IDS}_i, M_i))$ from a tag, and analyze the information carefully and try to detect the user location privacy by tracking the tag. Because the tag generates a new substitute random number $M = H(T_i \oplus R)$ during each authentication access, and updates $T_{i+1} = M \oplus H(M_i \oplus R \oplus \text{IDS}_i)$ in the phase2, so the adversary cannot determine which tag does the response from the message $(H(M_i \oplus R \oplus \text{IDS}_i, M_i))$. So this protocol can meet tag untraceability.

(2) Tag information protection

Because the information of an ID(Info) is stored in the backend server and is transmitted through the secure channel from the backend server to the reader, an adversary cannot acquire Info. So this protocol can meet tag information protection.

(3) Spoofing attack

An adversary feigns a legitimate reader that sends a query with R to tags through the forward channel, and obtains the response of a tag $(H(M_i \oplus R \oplus \text{IDS}_i, M_i))$. In the next authentication access, when a
An Efficient RFID Authentication Protocol Supporting Tag Ownership Transfer
He Jialiang, Ouyang Dantong, Xu Youjun, Shang Yu

legitimate reader sends query with R’, the adversary feigns the tag and responds the legitimate reader with the obtained message \( (H(M_L \parallel R) \oplus \text{IDS}_t, M_L) \) through the backward channel. However, the reader generates a new random number during each authentication access, namely \( R \neq R' \), so the adversary cannot perform tag impersonation.

(4) Replay attack
Replay attack can be prevented in this protocol due to the message transmitted for each session is different. Different value of \( H(M_L \parallel R) \) is utilized in individual session and \( T \) that stored in a tag plays a key role in providing different value of \( H(M_L \parallel R) \) to conceal pseudonym IDS of the tag. An adversary cannot hold \( H() \) and then acquire \( H(M_L \parallel R) \), so it is impossible for an adversary to apply replay attack.

(5) Denial of Service (DoS) attack
As pseudonym IDS of a tag is mutative, even if loss of message, power failure or loss of connection with the backend server happens during an authentication access, it will lead to dy-synchronization between the backend server and the tag, this protocol can solve this problem in the next authentication access by searching pseudonym IDS in column ‘Pre-IDS’ and continuing the verification process. So this protocol can shield DoS attack well.

(6) Forward security.
Both the backend server and a tag store pseudonym IDS, and update them in each authentication session. So forward security is assured because IDS has no relationship with previous sessions.

(7) Backward security.
Both the backend server and the tag store pseudonym IDS, and update them in each authentication session, key is stored in the backend server only, and only the backend server hold \( E_k() \), an adversary cannot acquire \( E_k(ID) \) and forecast the behavior of the tag in the intending authentication access. So this protocol can meet backward security.

Table 3 indicates a comparison of results among our protocol and [11], [12] in terms of security.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag untraceability</td>
<td>O</td>
<td>X</td>
<td>O</td>
</tr>
<tr>
<td>Tag information protection</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Spoofing attack</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Replay attack</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>DoS attack</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Forward security</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Backward security</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
<tr>
<td>Mutual authentication</td>
<td>O</td>
<td>O</td>
<td>O</td>
</tr>
</tbody>
</table>

‘O’ denotes satisfied, ‘X’ denotes not satisfied

4.2 Performance evaluations
Storage cost: a tag only stores ID, IDS and temporary value \( T \) in its memory, so this protocol can meet the storage constraints in a low-cost RFID environment.

Computational cost: in this protocol, only hash operation, XOR calculation and concatenation operation are used in tags, without pseudo random number generator used because of hardware cost, so this protocol can satisfy the restriction of the computational cost very well.

Traffic cost: this protocol needs five rounds to perform an authentication access, and the efficiency of the transmission and communication cost has been economized well.

Workload of the backend server: in the backend server, only one hash operation is needed for verifying and authenticating a tag, the backend server performs a constant time search to find the match item, so this protocol has well scalability and is very suitable to managing large population of tags.

Table 4 indicates a comparison of results among our protocol and [11], [12] in terms of performance.
Table 4. Comparison of performance

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Storage cost</td>
<td>Tag 2I</td>
<td>4I</td>
<td>3I</td>
</tr>
<tr>
<td>Computation cost</td>
<td>Tag 5h</td>
<td>4h</td>
<td>4h</td>
</tr>
<tr>
<td></td>
<td>Reader r</td>
<td>r</td>
<td>r</td>
</tr>
<tr>
<td></td>
<td>Server (((n/2+1))*h+3h)</td>
<td>3h</td>
<td>3h</td>
</tr>
<tr>
<td>Traffic cost</td>
<td>T to R 3l</td>
<td>2.5l</td>
<td>1.5l</td>
</tr>
<tr>
<td></td>
<td>R to T 2l</td>
<td>1l</td>
<td>1.5l</td>
</tr>
<tr>
<td></td>
<td>Total 5l</td>
<td>3.5l</td>
<td>3l</td>
</tr>
<tr>
<td>Rounds</td>
<td>7</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>Hardware cost</td>
<td>Tag H</td>
<td>H</td>
<td>H</td>
</tr>
<tr>
<td>Time complexity</td>
<td>Server O(n)</td>
<td>O(1)</td>
<td>O(1)</td>
</tr>
</tbody>
</table>

5. Conclusion

RFID systems suffer from many privacy and security problems, and scalability becomes an important requirement gradually. Therefore, in this paper, a new efficient RFID mutual authentication protocol supporting tag ownership transfer is proposed, this protocol only requires O(1) work to identify and authenticate a tag in the backend server, it alleviates workload of the backend server sharply, so this protocol has well scalability and is very suitable to managing large population of tags. The careful security analysis shows that this protocol can resist spoofing attack, replay attack, DoS attack, and meet tag untraceability, tag information protection, forward security and backward security; the performance evaluations show that this protocol is suitable for low-cost RFID systems well.

6. Acknowledgement

This work was supported in part by NSFC under Grant Nos. 60973089, 60873148, 60773097, 61003101; Jilin Province Science and Technology Development Plan under Grant Nos. 20101501, 20100185, 20090108, 20080107, 201101039; Doctoral Fund of Ministry of Education of China (20100061110031); Opening Fund of Top Key Discipline of Computer Software and Theory in Zhejiang Provincial Colleges at Zhejiang Normal University; Zhejiang Provincial Natural Science Foundation under Grant Nos. Y1100191; Erasmus Mundus External Cooperation Window’s Project (EMECW): Bridging the Gap, 155776-EM-1-2009-1-IT-ERAMUNDUS-ECW-L12. The Opening Foundation of Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education under Grant No. 93K-17-2009-K05.

7. References


