An Improved RFID Authentication Protocol with Privacy Protection Based on Chaotic Maps

Hongfeng Zhu, Yan Zhang

Software College, Shenyang Normal University, China
zhuhongfeng1978@163.com, 1505733680@qq.com

Abstract

Considering the convenience of reusability and identifiability of Radio frequency identification (RFID) technology, increasingly industries tend to use these properties, such as in the area of transportation, logistics, medical treatment, military, and so on. Many researchers have proposed RFID authentication protocols that conform to the EPCglobal Class 1 Generation 2 (EPC C1-G2) standard. Recently, Akgun et al. have proposed an RFID authentication protocol which utilized the Chebyshev chaotic map hard problem to realize flexibility RFID application. However, we point out three security flaws in their protocol. Firstly, we find that their protocol lack of RFID tag verification. Secondly, the protocol is vulnerable to tag impersonation attack and de-synchronization attack. Therefore, in this paper, we propose an improved RFID authentication protocol based on chaotic maps hard problem. Our protocol has the feature of privacy protection and robust to many malicious attacks. Moreover, the protocol also enhances the security of the RFID system.

Keywords: RFID, De-synchronization attack, Chaotic maps, Authentication

1 Introduction

With the rapid development of RFID technology, its application has been extended to every corner of our the society and our daily life. Due to the sharp decline in the cost and the improvement of the function, the concerns about RFID technology in the manufacturing, logistics [1], retail, medical [2] and defense fields have been increasing rapidly. With the aid of RFID technology, the logistics enable achieve objects tracking and the society prevent the infant from stealing in the hospital, the airline realize real-name authentication of passengers' identity. Moreover, considering the accuracy of recognition and reusability, RFID technology devotes to build a convenient platform to meet all devices which need identification. However, due to the widespread use of the RFID technology, the weaknesses are revealed gradually.

Generally, Therefore, RFID tags can be divided into four types: ultralight tags, lightweight tags, simple tags and high cost tags. In the process of these applications, we found some defects in RFID technology. Firstly, RFID mechanism includes transmitter, reader and encoder, but the costs of these combinations are not cheap, if our society intends to promote use scope, the cost is worth considering. Secondly, RFID mechanism is mainly devotes to identification, so we should clear that RFID device can easily access to our privacy without our knowing. Similarly, once the tags close to the RFID reader within certain limits, the tag enable transfer information to the reader unconditional. Recently, for the purpose of fulfill tag redeployment and reduce tag cost, many researchers and organizations with a view to achieve EPC C1-G2 [3-4]. Unfortunately, research showed that this tag standard cannot satisfy every different RFID mechanism, and then some researchers also point out that there were security vulnerabilities in the standard.

According to all of these proposed protocols, the common goal is to establish a low-cost and reusable RFID tags. In 2005, the *Electronic News* [5] published a novel periodical that Royal Philips intends to produce RFID chip conforming to EPCglobal. In 2007, Chien and Chen [3] proposed a mutual authentication protocol for passive tags. Based on the function of cyclic redundancy code (CRC), the protocol failed to realize identification completely. In contrast, it exposed some weaknesses like tag forgery attack, denial-of-service attack and backward security attack. Meanwhile, considering the authentication process adopts exhaustive search, the method reduce the efficiency of scheme. In 2010, Yeh et al. [6] proposed a new RFID system conforming to this standard and the protocol uses PENG function instead of CRC algorithm. Yoon [7] point out some weaknesses of Yeh et al. scheme including forward attack and data integrity, later, Yoon proposed an improved protocol to eliminate these attacks.

Chaotic maps algorithm has widely used in cryptography [8], including group key exchange for group session [9], identity authentication based on smart card [10] and composing image encryption for

[*]Corresponding Author: Hongfeng Zhu; E-mail: zhuhongfeng1978@163.com

DOI: 10.3966/160792642018111906014

multimedia communication [11]. Chaos theory is an evolutionary theory that makes the system from order to disorder, which is the method to research the form of “random process” mechanism. Pseudo-randomness and non-periodicity are two main characteristics of chaotic cryptosystem, and it also has the nature of sensibility to initial parameters and owns unpredictability. So on account of these properties this algorithm enable achieve encryption system based on hard problem. Therefore, chaos theory has been widely noted and used in cryptographic articles.

In 2013, Wang et al. [12] proposed a RFID security mechanism based on chaotic maps, the protocol intends to solve authentication problems by chaotic algorithm. However, in 2014, Benssalah et al. [13] also proposed a chaotic map-based RFID authentication protocol, and claimed that their protocol enable eliminate the weaknesses which appeared in Cheng et al. Unfortunately, in 2015, Akgun et al. [14] analyzed the security of Benssalah et al. and found that it was vulnerable to tracking, tag impersonation and de-synchronization attacks. So they put forward an improved protocol to overcome these problems. In this paper, we find that though Akgun et al. has correct some weaknesses in previous agreements, there are still have some weaknesses in their protocols such as de-synchronization attack and impersonation attack. Therefore, we propose an improved RFID authentication protocol, our protocol based on the Chebyshev chaotic map hard problems and hash function.

The rest of paper is organized as follows: some preliminaries are introduced in Section 2. We display the authentication protocol of Akgun et al. and analyze its vulnerabilities in Section 3. In Section 4, we present our improved RFID authentication protocol. We present security analysis and efficiency analysis of our protocol in Sections 5 and Section 6. Finally, we conclude the paper in Sections 7.

2 Chebyshev Chaotic Maps

Let n be an integer and let x be a variable value with the interval $[-1,1]$. The Chebyshev polynomial $T_n(x):[-1,1]\to[-1,1]$ is defined as:

$$T_n(x)=\cos(n\cos^{-1}(x)).$$

Chebyshev polynomial map $T_n:R\to R$ of degree n is defined using the following recurrent relation:

$$T_n(x)=2xT_{n-1}(x)-T_{n-2}(x),$$

where $n\geq 2$, $T_0(x)=1$, and $T_1(x)=x$.

The first few Chebyshev polynomials are:

$$T_2(x)=2x^2-1,$$
$$T_3(x)=4x^3-3x,$$
$$T_4(x)=8x^4-8x^2+1.$$

One of the most important properties is that Chebyshev polynomials are the so-called semi-group property which establishes that $T_r(T_s(x))=T_{rs}(x)$.

An immediate consequence of this property is that Chebyshev polynomials commute under composition

$$T_r(T_s(x))=T_s(T_r(x)).$$

Definition 1. (Enhanced Chebyshev polynomials)

The enhanced Chebyshev maps of degree $n\,(n\in N)$ are defined as:

$T_n(x)=(2xT_{n-1}(x)-T_{n-2}(x))(\bmod p)$, where $n\geq 2$, $x\in(-\infty,+\infty)$, p is a large prime number. Obviously, $T_{rs}(x)=T_r(T_s(x))=T_s(T_r(x))=T_{sr}(x)$.

Definition 2. (Chaotic Maps-based Discrete Logarithm problem, DLP)

Given x and y, it is intractable to find the integer s, such that $T_s(x)=y$.

Definition 3. (Chaotic Maps-based Diffie-Hellman problem, CDH)

Given x, $T_r(x)$ and $T_s(x)$, it is intractable to find $T_{rs}(x)$, such that $T_r(T_s(x))=T_{rs}(x)$ or $T_s(T_r(x))=T_{rs}(x)$.

It is widely believed that there is no polynomial time algorithm could solve DLP, CDH hard problems with a non-negligible probability.

3 Review of Akgun et al.’s Scheme

In 2015, Akgun et al. [14] showed vulnerabilities in Benssalah et al.’s [13] protocol, and proposed an RFID authentication protocol based on Chebyshev chaotic map. Akgun et al. claimed that their protocol enable eliminate the weaknesses which appeared in [13] and could achieve mutual authentication and resist de-synchronization attack. Next, we provide the detailed analysis of the protocol. This section is composed of two phases: initialization phase and authentication phase.

3.1 Notations

The notations used in the protocol of Akgun et al. are introduced in Table 1.

Table 1. Notations

r , t , s	Generate by reader, tag and server
ID	
RID	
x	Current shared session key
x_{old}	Last session key
$T.(\cdot)$	Enhanced Chebyshev polynomial
$\in$	Random choice operation
$\oplus$	XOR operation
$\|\|$	Concatenation operation
$h(\cdot)$	
$\leftarrow$	

3.2 Initialization Phase

The back-end server generates a secret key x and stores $\{ID, x, c_i\}$ in tag memory, c_i is an index value of database. The server stores $\{ID, x_{old}, x_{new}, c_{old}, c_{new}\}$ in database where c_{old} and c_{new} are index values. The reader stores its identifier $\{RID\}$. First, set $x_{new} = x_{old} = x$ and $c_{new} = c_{old} = 0$.

3.3 Authentication Phase

The process of authentication in Akgun et al.'s scheme is shown as Figure 1.

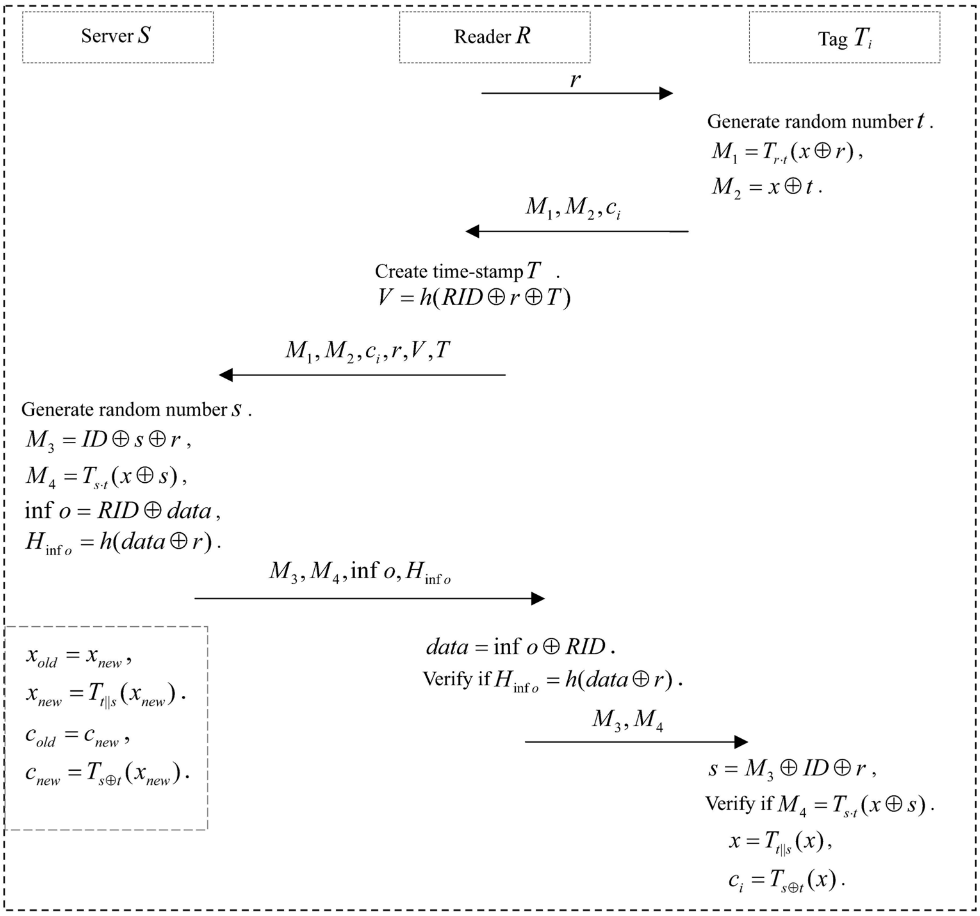
Server S
Reader R
Tag T_i
r
Generate random number t .
$M_1 = T_{r \cdot t}(x \oplus r)$,
$M_2 = x \oplus t$.
M_1, M_2, c_i
Create time-stamp T .
$V = h(RID \oplus r \oplus T)$
M_1, M_2, c_i, r, V, T
Generate random number s .
$M_3 = ID \oplus s \oplus r$,
$M_4 = T_{s \cdot t}(x \oplus s)$,
$\inf o = RID \oplus data$,
$H_{\inf o} = h(data \oplus r)$.
$M_3, M_4, \inf o, H_{\inf o}$
$x_{old} = x_{new}$,
$x_{new} = T_{t \| s}(x_{new})$.
$c_{old} = c_{new}$,
$c_{new} = T_{s \oplus t}(x_{new})$.
$data = \inf o \oplus RID$.
Verify if $H_{\inf o} = h(data \oplus r)$.
M_3, M_4
$s = M_3 \oplus ID \oplus r$,
Verify if $M_4 = T_{s \cdot t}(x \oplus s)$.
$x = T_{t \| s}(x)$,
$c_i = T_{s \oplus t}(x)$.

Figure 1. Authentication phase of Akgun et al.'s scheme

Step 1. The reader generates a random number r and sends it to tag.

Step 2. After receiving the number r , the tag generates a random number t , computes:

$$M_1 = T_{r \cdot t}(x \oplus r) , \; M_2 = x \oplus t .$$

Then sends (M_1, M_2, c_i) to the reader.

Step 3. The reader creates a time-stamp T and computes:

$$V = h(RID \oplus r \oplus T) .$$

Then sends (M_1, M_2, c_i, r, V, T) to the server.

Step 4. After receiving (M_1, M_2, c_i, r, V, T) , the server checks $V \overset{?}{=} h(RID \oplus r \oplus T)$. If holds, the server performs the following operations:

(1) If $c_i = 0$. The server executes an exhaustive search on its database to search qualified records as follows:

$$T_{old} = T_{(M_2 \oplus x_{old}) \cdot r}(x_{old} \oplus r)$$
$$T_{new} = T_{(M_2 \oplus x_{old}) \cdot r}(x_{new} \oplus r) .$$

Then, verify whether $M_1 \overset{?}{=} T_{old}$ or $M_1 \overset{?}{=} T_{new}$, if holds, finds the corresponding records and sets x to x_{old} or x_{new}.

(2) If $c_i \neq 0$. It means c_i is index of corresponding database entry. The server finds the value that meets the demand and sets x to x_{old} or x_{new}. Then verifies the validity of M_1. If not holds, the server rejects the tag.

After finishing the above operations, the server computes:

$$M_3 = ID \oplus s \oplus r\ ,\ M_4 = T_{s\cdot t}(x \oplus s)\ ,$$
$$\inf o = RID \oplus data\ ,\ H_{\inf o} = h(data \oplus r)\ .$$

Then sends $(M_3, M_4, \inf o, H_{\inf o})$ to the reader.

The server conducts key updating as follows:

$$x_{old} = x_{new}\ ,\ x_{new} = T_{t\|s}(x_{new})\ .$$
$$c_{old} = c_{new}\ ,\ c_{new} = T_{s\oplus t}(x_{new})\ .$$

Step 5. After receiving $(M_3, M_4, \inf o, H_{\inf o})$, the reader computes $data = \inf o \oplus RID$ and verifies whether $H_{\inf o} \overset{?}{=} h(data \oplus r)$. If holds, sends (M_3, M_4) to the tag, otherwise, the reader terminates the transmission.

Step 6. After receiving (M_3, M_4), the tag retrieves s from M_3 as $s = M_3 \oplus ID \oplus r$. Then the tag checks the validity of M_4. If valid, the tag conducts key updating:

$$x = T_{t\|s}(x)\ ,\ c_i = T_{s\oplus t}(x)\ .$$

3.4 Weaknesses of Akgun et al.'s Scheme

In this protocol, the fresh random number r, t and s cannot be reused over and over again. So the messages in previous sessions unable reuse in other sessions.

Akgun et al. claimed that the protocol enable eliminate the vulnerabilities in Benssalah et al.'s protocol, it resist secure disclosure attack and de-synchronization attack. Unfortunately, we find that their protocol fails to resist these attacks.

3.4.1 Tag Impersonation Attack

Tag impersonation attack means that a forge tag could be identified by the reader, and this attack could threat the security of protocol.

(1) Supposed that an adversary A eavesdrops the last successful authentication session between a tag T_i and the reader R, and A records the messages $\{r, M_1, M_2, c_i\}$:

$$M_1 = T_{r\cdot t}(x \oplus r)\ ,\ M_2 = x \oplus t\ .$$

(2) R sends a new challenger $r^{'}$ to the tag to start a session.

(3) A receives the messages as follows:

$$M_1^{'} = T_{\frac{r^{'}}{r}}(M_1) = T_{\frac{r^{'}}{r}}(T_{r\cdot t}(x \oplus r)) = T_{r^{'}\cdot t}(x \oplus r)\ ,$$
$$M_2^{'} = M_2 = x \oplus t\ ,\ C_i^{'} = C_i\ .$$

(4) After receiving $\{r^{'}, M_1^{'}, M_2^{'}, c_i^{'}\}$, the server searches the record. And retrieve t with $t = M_3^{'} \oplus x$. Then verifies the validity of $M_1^{'}$ as follows:

$$M_1^{'} = T_{r^{'}\cdot t}(x \oplus r)\ . \qquad \textbf{(*)}$$

(5) We believed that A has authenticated by the server. It means the server compute M_3 and M_4 to the reader.

The original message transmission from a tag to the reader, so without restriction of tag identity (ID), any unqualified tags enable achieve authentication with the server.

3.4.2 Privacy Disclose Attack

In this protocol, the server S and tag T_i store tag identity in their database. And the protocol utilizes random numbers t and s to ensure secure messages transmission. However, considering there is no verification about tag identity between T_i and S in the first interaction, so each tag can be authenticated by S. While after verification, S calculates $M_3 = ID \oplus s \oplus r$ and $M_4 = T_{s\cdot t}(x \oplus s)$ to T_i. Consequently, the adversary enable simulate this process and retrieves a tag's identifier.

Suppose that there is an adversary A intends to obtain an identifier of effective tag, the interaction process is as follows:

Round 1: After receiving r from R, T_a selects a random number t, computes $M_1^a = T_{r\cdot t}(x \oplus r)$ and $M_2^a = x \oplus t$, then T_a transfers $\{M_1, M_2, c_a\}$ to S. According to M_1^a and M_2^a, S sends M_3^a and M_4^a back to T_a as follows:

$$M_3^a = ID \oplus s_a \oplus r\ ,\ M_4^a = T_{s_a\cdot t}(x \oplus s_a)$$

Round 2: A eavesdrops $\{r, M_1^a, M_2^a, c_a M_3^a, M_4^a\}$ by "**Round 1**". Then, A disguises the last eligible tag T_a to initiate a new session, after receiving the new challenger $r^{'}$ from the reader, A sends $\{M_1^{a'}, M_2^{a'}, c_a^{'}\}$ to S, where

$$M_1^b = T_{\frac{r^{'}}{r}}(M_1^a) = T_{\frac{r^{'}}{r}}(T_{r\cdot t}(x \oplus r)) = T_{r^{'}\cdot t}(x \oplus r)\ ,$$
$$M_2^b = M_2^a = x \oplus t\ ,\ c_b = c_a\ .$$

Then S computes $\{M_3^b, M_4^b\}$ and returns to T_a as follows:

$$M_3^b = ID \oplus s_b \oplus r', \; M_4^b = T_{s_b \cdot t}(x \oplus s_b). \qquad \textbf{(2)}$$

Round 3: A intercepts $\{M_3^b, M_4^b\}$ and repeats "**Round 2**". Then the server returns $\{M_3^c, M_4^c\}$ to the eligible tag T_a:

$$M_3^c = ID \oplus s_c \oplus r'', \; M_4^c = T_{s_c \cdot t}(x \oplus s_c). \qquad \textbf{(3)}$$

Round 4: Repeat the operation as "**Round 3**" in limit times, then A gets $\{M_3^*, M_4^*\}$ as follows:

$$M_3^* = ID \oplus s_* \oplus r^*, \; M_4^* = T_{s_* \cdot t}(x \oplus s_*). \qquad \textbf{(*)}$$

Considering ID is a fixed value for all authentication rounds, and A imitates the process with the eligible tag T_a. So, A utilizes $\{M_3^a, M_3^b, M_3^c, ..., M_3^*\}$ and XOR to retrieve ID as follows:

$$M_3^a \oplus M_3^b \oplus M_3^c \oplus ... \oplus M_3^* = (ID \oplus s_a \oplus r) \oplus$$
$$(ID \oplus s_b \oplus r') \oplus (ID \oplus s_c \oplus r'') \oplus ... \oplus (ID \oplus s_* \oplus r^*).$$

Therefore, although the protocol based on the random number, it still unable protects the privacy of eligible tags. Firstly, the server cannot achieve identifier authentication on tag. Secondly, the server computes the value of M_3 with a plaintext ID. These are the loopholes for the adversary retrieves the tag identifier (ID) from the transmitted messages.

3.4.3 De-synchronization Attack

The de-synchronization attack means that an adversary forces one party updates its session key in this process but another party still keeps the previous key. As a result, the reader and the tag cannot authenticate each other in subsequent sessions.

(1) A eavesdrops the last successful authentication session between T_i and R, A sends r to start a session.

(2) T_i transfers the messages $\{M_1, M_2, c_i\}$ to R.

(3) After intercepting $\{M_1, M_2, c_i\}$, A disguises R with the tag ID and computes messages $\{M_3', M_4'\}$ as follow:

$$M_3' = ID \oplus s' \oplus r,$$
$$M_4' = T_{\frac{s'}{s}}(M_4) = T_{\frac{s'}{s}}(T_{s \cdot t}(x \oplus s)) = T_{s' \cdot t}(x \oplus s).$$

(4) After receiving $\{M_3', M_4'\}$, T_i authenticates S with M_4'. If success, T_i updates $x = T_{t\|s}(x)$ and $c_i = T_{s \oplus t}(x)$.

However, since the passive adversary launches the de-synchronization attack and impersonate the server to perform these operations. Therefore, although the tag updates the session key with $T_{t\|s}(x)$ and $T_{s \oplus t}(x)$, the server still stores the previous values. As a result, S unable authenticates T_i in the following sessions.

4 The Improved Scheme

In this section, we propose an improved RFID protocol based on chaotic maps to achieve information transmission. After analyzed Akgun et al.'s [14] scheme we find that their scheme lacks of tag identifier in M_1 and M_2, causing any unqualified tags which access to the reader could read internal information. Meanwhile, the server computes M_3 with the plaintext *ID*, an adversary could derive eligible tag *ID* from the transmitted messages. Therefore, we use $h(ID)$ to protect tag identity and support the server to verify tags. The protocol is generated by two phases: initialization phase and authentication phase.

4.1 Notations

The notations in our protocol are shown in Table 2.

Table 2. Notations

r, t, s	Generate by reader, tag and server
ID	Tag identifier, secret in the tag and server
$h(ID)$	The hash value of the tag identity
RID	Reader identifier, secret in the reader and server
x	The current shared session key
x_{old}	The last session key
$\oplus$	XOR operation
$\|\|$	Concatenation operation
$h(\cdot)$	Hash function

4.2 Initialization Phase

Firstly, we put $\{ID, h(ID), x_{old}, x_{new}, c_{old}, c_{new}, RID\}$ into the back-end server. Meanwhile, each tag stores $\{ID, h(ID), x, c_i\}$ in its memory. The reader stores its identifier $\{RID\}$. We set $x_{new} = x_{old} = x$, $c_{new} = c_{old} = 0$.

4.3 Authentication Phase

The process of our improved protocol is shown in Figure 2.

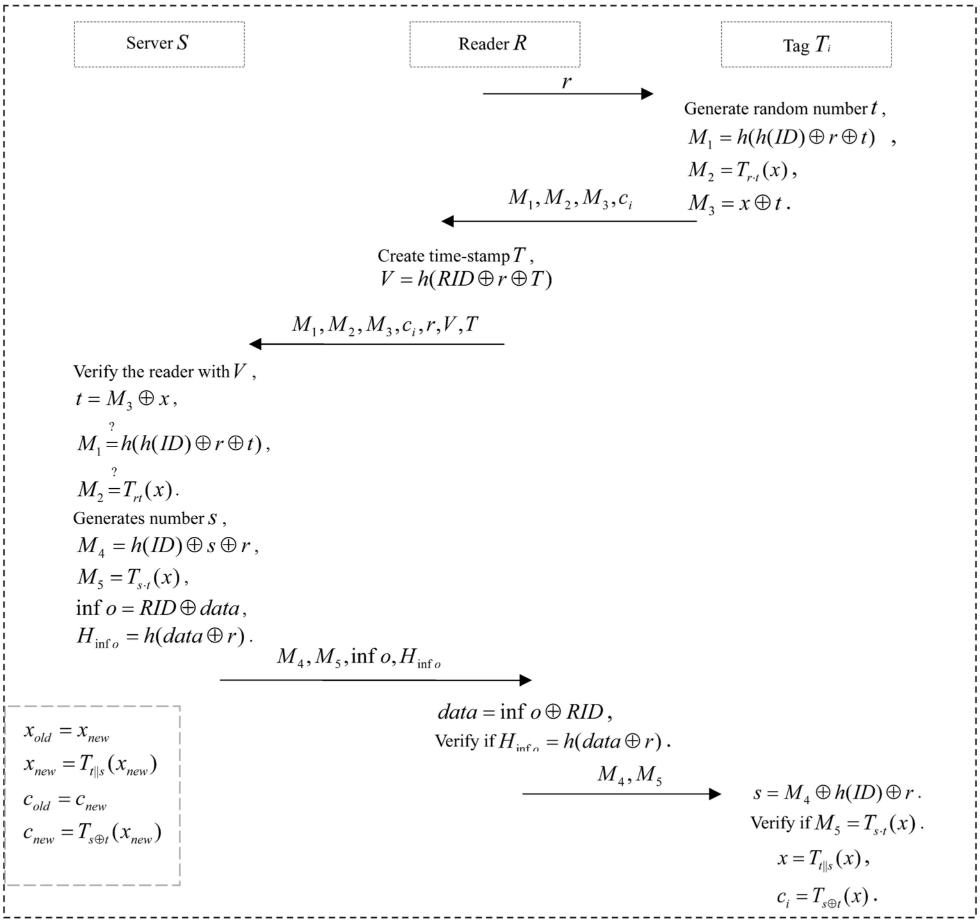
Server S
Reader R
Tag T_i
r
Generate random number t,
$M_1 = h(h(ID) \oplus r \oplus t)$,
$M_2 = T_{r \cdot t}(x)$,
$M_3 = x \oplus t$.
M_1, M_2, M_3, c_i
Create time-stamp T,
$V = h(RID \oplus r \oplus T)$
$M_1, M_2, M_3, c_i, r, V, T$
Verify the reader with V,
$t = M_3 \oplus x$,
$M_1 \overset{?}{=} h(h(ID) \oplus r \oplus t)$,
$M_2 \overset{?}{=} T_{rt}(x)$.
Generates number s,
$M_4 = h(ID) \oplus s \oplus r$,
$M_5 = T_{s \cdot t}(x)$,
$\inf o = RID \oplus data$,
$H_{\inf o} = h(data \oplus r)$.
$M_4, M_5, \inf o, H_{\inf o}$
$x_{old} = x_{new}$
$x_{new} = T_{t\|s}(x_{new})$
$c_{old} = c_{new}$
$c_{new} = T_{s \oplus t}(x_{new})$
$data = \inf o \oplus RID$,
Verify if $H_{\inf o} = h(data \oplus r)$.
M_4, M_5
$s = M_4 \oplus h(ID) \oplus r$.
Verify if $M_5 = T_{s \cdot t}(x)$.
$x = T_{t\|s}(x)$,
$c_i = T_{s \oplus t}(x)$.

Figure 2. The process of our improved protocol

Step 1: R generates a random number r and sends to T_i.

Step 2: After receiving r, T_i generates t and computes:

$$M_1 = h(h(ID) \oplus r \oplus t), \quad \textbf{(1)}$$

$$M_2 = T_{r \cdot t}(x), \quad \textbf{(2)}$$

$$M_3 = x \oplus t. \quad \textbf{(3)}$$

Then sends $\{M_1, M_2, M_3, c_i\}$ to the reader.

Step 3: The reader creates a timestamp T and computes:

$$V = h(RID \oplus r \oplus T). \quad \textbf{(4)}$$

Then sends $\{M_1, M_2, M_3, c_i, r, V, T\}$ to the server.

Step 4: After receiving $\{M_1, M_2, M_3, c_i, r, V, T\}$, the server performs the following operations:

(1) Verify whether $V \overset{?}{=} h(RID \oplus r \oplus T)$, if holds, the reader becomes a trusted party.

(2) Check the index c_i as follows:

■If $c_i = 0$;

S performs the exhaustive search as follows:

$$T_{old} = T_{(M_3 \oplus x_{old}) \cdot r}(x_{old}), \quad \textbf{(5)}$$

$$T_{new} = T_{(M_3 \oplus x_{old}) \cdot r}(x_{new}). \quad \textbf{(6)}$$

If M_2 matches T_{old} or T_{new}, S finds the corresponding records and sets x to x_{old} or x_{new}. Then S retrieves t by $t = M_3 \oplus x_i$ (i =old or new), and verifies whether $M_1 \overset{?}{=} h(h(ID) \oplus r \oplus t)$.

■If $c_i \neq 0$;

It means that c_i is index of the entry, S finds the entry with c_{old} or c_{new} and sets x to x_{old} or x_{new}.

Then S retrieves t by $t = M_3 \oplus x_i$. Verifies whether $M_1 \overset{?}{=} h(h(ID) \oplus r \oplus t)$ and $M_2 \overset{?}{=} T_{r \cdot t}(x_i)$. If holds, the server continue to perform the operation.

(3) S generates a random number s, computes:

$$M_4 = h(ID) \oplus r \oplus s, \quad \textbf{(7)}$$

$$M_5 = T_{s \cdot t}(x), \quad \textbf{(8)}$$

$$\inf o = RID \oplus data, \quad \textbf{(9)}$$

$$H_{\inf o} = h(data \oplus r). \quad \textbf{(10)}$$

Then sends $\{M_4, M_5, \inf o, H_{\inf o}\}$ to the reader.

(4) S performs the key updating as follows:

$$x_{old} = x_{new},\ x_{new} = T_{t\|s}(x_{new}).$$

$$c_{old} = c_{new},\ c_{new} = T_{s\oplus t}(x_{new}).$$

Step 5: After receiving the messages, the reader computes:

$$data = \inf o \oplus RID,\ H_{\inf o} \overset{?}{=} h(data \oplus r).$$

If holds, sends the messages $\{M_4, M_5\}$ to the tag.

Step 6: After receiving the messages $\{M_4, M_5\}$, T_i retrieves s by $s = M_4 \oplus h(ID) \oplus r$, then verifies whether $M_5 \overset{?}{=} T_{s\cdot t}(x)$, if holds, S performs key updating:

$$x = T_{t\|s}(x), \tag{11}$$

$$c_i = T_{t\oplus s}(x). \tag{12}$$

5 Security Consideration

In this section, we provide a formal security model for our improved RFID authentication protocol based on chaotic maps. We give a detailed analysis to prove that our protocol meets the security requirements under the random oracle model.

5.1 Formal Security Model

Players. We denote a tag T and a server S that participate in the authentication protocol P. Each of them may have several instances involved in distinct, possibly concurrent, executions of P. We denote the tag instances and the server instances by T^i, S^j, R^k ($i, j, k \in Z$), and denote any kind of instance by I.

Queries. The interaction between A and the participants occurs in oracle queries, which denotes A intends to break the authentication. Several queries are avaliable to A.

- Execute (T^i, S^j, R^k) : This oracle query models passive attack, A eavedrops the honest executions between the T^i and S^j.
- Send (I, m) : This oracle query models active attack, A sends a message to instance I and receives the response I back generates in processing the message m according to P. The query sends (T^i, Start) initializes the key change protocol, where Start denotes the message. Therefore, A receives the tag should send to the server.
- Text (I) : This oracle query is used to define the semantic security of session key. Define a private coin c is flipped, if $c = 1$ then the session key is returned to adversary A, or else returned to a same size random key.
- Reveal (I) : This oracle query models the misuse of session key. If the pointed instance holds a session key and I was not models by the text query, A obtain the session key x in instance I.
- Corrupt (I, a) : This oracle query models the corruption capability of A. A enable steal/break tag's any one of the two authentication factors, but not all of both.
  — If a=1, it outputs the identity ID of tag T.
  — If a=2, it outputs $\{ID, r, t\}$ select by tag.
  — If a=3, it outputs c_i and last session key x_{old}.

Authentication. The significance of authentication phase is to resist any adversary impersonate the tag or the server. We denote by $Adv_P^{auth}(A)$ that A impersonates a participant as an instance of either T or S in protocol P.

Semantic Security. In the execution of P, an adversary launch a Execute (T^i, S^j) -Query, Send (I, m) -Query, Reveal (I) -Query and Corrupt (I, a) -Query, enable launch a singleText (I) -query to a instance. In the Text-Query, A outputs a guess bit c'. If $c' = c$, A win the game, and we denote it by $Succ$. We define $Adv_P^{auth}(A) = 2\Pr[Succ(A)] - 1 = 2\Pr[c' = c] - 1$ as the probability of adversary A breaks the semantic security. And protocol P is said to be semantically secure if the advantage of A is negligible in the security parameter.

Computational Chaotic-based D-H Assumption (CCDH). Suppose that p be a large prime, $n \in N$, $x \in Z_p$ and $T_n \bmod p$ was generated by Chebyshev polynomial. A is provided with $T_r(x)$, $T_s(x)$ and $T_{rs}(x)$ in the experiment $Exp_{x,p}^{ccdh}(A)$. So define an CCDH assumption as follows:

$$Adv_{x,p}^{ccdh}(A) = \max\{\Pr[Exp_{x,p}^{ccdh}(T_r(x), T_s(x)) = T_{rs}(x)]\}.$$

5.2 Formal Security Proof

Theorem1. Let D be a uniformly distributed dictionary of size $|D|$, let P be the improved authentication scheme showed in **Section 4**. Let A be an adversary against the semantic security, so,

$$Adv_{P,D}^{ccdh}(A) \le \frac{q_H^2}{2^l} + \frac{(q_s + q_e)^2}{p^2} + 2q_e \cdot Adv_{x,p}^{ccdh}(A) + 2\max\left\{\frac{q_H}{p}, \frac{q_s}{D} + \frac{q_s}{2^l}\right\},$$

q_s denotes the Send-query, q_e denotes the Execure-query, q_H denotes the Hash-query.

Proof: We give a sequence of games to support this theorem start at G_0 to G_4, where A has no advantage.

Game **G_0 :** This game in the random oracle model that corresponding to the real attack. Define $Succ_i$ as the adversary guesses the bit b involved in the Test-query. So,

$$Adv_P^{auth}(A) = 2\,|\Pr[Succ_0] - \frac{1}{2}|.$$

Game G_1**:** In this game, we simulate the hash oracles H_i as usual by maintaining hash list H_Λ. We also simulate all the instances for the Send-query, Execure-query, Reavel-query, Test-query and Corrupt-query. Therefore, we find that the game is perfectly indistinguishable from the real attack. So,

$$|\Pr[Succ_1] - \Pr[Succ_0]| = 0.$$

Game G_2 **:** We simulate all the oracles in game G_1 except that we cancle the game where collisions appear on $((M_1, M_2, M_3, c_i), (M_4, M_5))$ and hash values.

The probability bounded by birthday paradox is:

$$|\Pr[Succ_2] - \Pr[Succ_1]| \le \frac{(q_s + q_e)^2}{2p^2} + \frac{q_H^2}{2^{l+1}},$$

where $l = \min\{l_i\}, i = 0,1,2,3$.

Game G_3**:** We simulate the random oracle on Send-query. We start the simulation with T, S, R, and this game is perfectly indistinguishable from the previous game G_2, so,

$$|\Pr[Succ_3] - \Pr[Succ_2]| = 0.$$

Game G_4**:** We define an adversary may have lucky in guessing the correct parameter and impersonate the server or the tag. Then, we have,

$$|\Pr[Succ_4] - \Pr[Succ_3]| \le q_e \cdot Adv_{x,p}^{ccdh}(A),$$

$$\Pr[Succ_4] = \frac{1}{2} + \max\left\{\frac{q_H}{p}, \frac{q_s}{|D|} + \frac{q_s}{2^l}\right\}.$$

Integrating all the above equations, we get:

$$\begin{aligned} Adv_{p,D}^{ccdh}(A) &= 2\,|\Pr[Succ_0] - \frac{1}{2}| \\ &= 2\,|\Pr[Succ_0] - \Pr[Succ_4] + \max\left\{\frac{q_H}{p}, \frac{q_s}{|D|} + \frac{q_s}{2^l}\right\}| \\ &\le 2(|\Pr[Succ_1] - \Pr[Succ_2]| + |\Pr[Succ_3] - \Pr[Succ_4]| \\ &\quad + \max\left\{\frac{q_H}{p}, \frac{q_s}{|D|} + \frac{q_s}{2^l}\right\}) \le \frac{(q_s + q_e)^2}{2p^2} + \frac{q_H^2}{2^{l+1}} + \\ &\quad 2q_e \cdot Adv^{cddh}(A) + 2\max\left\{\frac{q_H}{p}, \frac{q_s}{|D|} + \frac{q_s}{2^l}\right\}. \end{aligned}$$

5.3 Achieve Mutual Authentication

The improved protocol provides mutual authentication between T_i and S. It means that only each tag achieve the authentication from tag and server can be regarded as a eligible tag. The protocol depends on random number and chaotic maps hard problem. It also indicates that our improved protocol enable resist some malicious attacks.

Proof. Mutual authentication is a critical step for the server validate the tag identity. In our protocol, after receiving r from R, T_i computes $\{M_1, M_2, M_3\}$ and then sends it to S. Then S retrieves the random number that generates by T_i and verify the identity with $M_1 \overset{?}{=} h(h(ID) \oplus r \oplus t)$. Meanwhile, S checks whether $M_2 \overset{?}{=} T_{r \cdot t}(x)$ based on the hard problem (**Definition 2**). If M_1 and M_2 meet the requirements, T_i is considered achieve verification by S. Then S computes $\{M_4, M_5\}$ and sends the messages to T_i. Then, T_i retrieves s by $s = M_4 \oplus h(ID) \oplus t$, and only eligible tag owns the same identifier $h(ID)$ with the server. Then, T_i verifies whether $M_5 \overset{?}{=} T_{s \cdot t}(x)$ with the value s. If holds, T_i consider S holds the common secret messages. Therefore, T_i believes that it certifies S successful, then T_i conducts tag updates. After achieve the operations, we believe that the server S and the tag T_i achieve mutual authentication successful.

Theorem 5.3.1 On the basis of Chebyshev hard problem in **Definition 2** and **Definition 3**, and since the protocol achieves the mutual authentication. We can prove that our protocol defects tag impersonation attack.

Proof. Assume that an adversary A intends to impersonate a tag T_i to communicate with the server S, so A imitates the operations as T_i. Adversary A eavesdrops the last successful authentication session between T_i and R, A records the messages $\{r, M_1^n, M_2^n, M_3^n, c_i\}$. Next, A intends to initiate a new session $n+1$ and computes $M_1^{n+1}, M_2^{n+1}, M_3^{n+1}$. Firstly, A needs to retrieve $h(ID)$ from the last session to compute M_1^{n+1}. However, A cannot obtain an eligible tag identity from the previous operations because of the identity is secret stored in the server. Meanwhile, in order to calculate authentication message M_2^{n+1}, A needs to obtain the random number t^n. But it is difficult and ineffective to the adversary retrieve t^n from M_2^n. Therefore, A unable achieve authentication from the server without knowing the tag identity $h(ID)$ and the integer t^n. As a result, the adversary fails to launch tag impersonation attack.

Theorem 5.3.2 The improved protocol enable defect server impersonation attack based on hard problem in **Definition 2** and **Definition 3**.

Proof. Assume that an adversary A intends to impersonate the server to the tag T_i, so A needs to imitate the process with T_i. After receiving $\{M_1, M_2, M_3, c_i\}$, A computes $M_4 = h(ID) \oplus r \oplus s$ and $M_5 = T_{s \cdot t}(x)$. Since s is a random number which could generate by A, and the value r enable obtain in public channel, suppose that A eavesdrops the last successful authentication session, but an eligible tag identity is stored both in the tag and the server. Therefore, only the server S with the tag T_i can calculate M_4. Meanwhile, the secret value t is generates by the eligible tag, and based on hard problem (CMBDLP), it is difficult for A to retrieve t from the previous $T_{s \cdot t}(x)$. It indicates that A cannot compute M_5 in this process. Only verify the message M_5 successful that T_i believe the authenticity of S, and convince that S holds the same messages with itself. From this analysis we conclude that the adversary cannot impersonate the server, namely, our improved protocol resist the server impersonation attack.

5.4 Realize Anonymity

Proof. Anonymity is an important property for privacy protection in RFID system. Protect tag identity help the tag hide it trace and location information. Meanwhile, in RFID system, guarantee the tag identity anonymity assist the protocol defects masquerade attack. In our protocol, in order to preserve the tag identity, no one knows the tag information $(ID, h(ID))$ except for the tag and the server. Similarly, the reader identity RID is only hold by the reader and the server. Since our improved protocol provides mutual authentication and based on Chebyshev hard problem. If an adversary wants to initiate a malicious query to the legitimate tag and tries to obtain the tag identity from $\{M_1, M_2, M_3\}$, it would teminate by the unknown secret value. The messages M_1, M_2 and M_3 are composed of the random number t and secret session key x, and they are updated in every communication interaction. Therefore, the tag identity cannot be obtained by the adversary with the limit queries. So the improved protocol protect the tag identity and realize anonymity.

5.5 Resist Replay Attack

Proof. Replay attack means that an adversary intends to use the messages which obtained from the previous sessions to impersonate an eligible tag or a legitimate server. Suppose that an adversary tries to carry out a replay attack in order to deception the RFID system. Therefore, the adversary must reuse the previous messages $\{M_1, M_2, M_3, c_i, r\}$ and $\{M_4, M_5\}$. Unfortunately, these messages cannot be accept in the following operations, because each message is composed of random number t, s and the secret session key x. Above all, the nonce t and s are generated by the tag and server, based on the hard problem (**Definition 2** and **Definition 3**), the random number cannot be computed within the limit calculation. Hence, we believe that the adversary unable reuses the previous messages to start replay attack.

5.6 Resist De-synchronization Attack

Proof. If an attacker intends to launch de-synchronization attack, the server and tag should update their session key at different times. In order to prevent this attack, our protocol stores the session key in the tag and server at the same time. Suppose an adversary A intercepts $\{M_1, M_2, M_3, c_i, r\}$ to the server, aiming to force the tag updates its secret values x_i and c_i, the adversary should impersonate the server to compute $\{M_4, M_5\}$ to the tag. As mentioned above, the secret messages ID and x are protected by the system, therefore, the adversary cannot calculate $\{M_4, M_5\}$ and the tag cannot authenticate the illegal tag. As a result, our protocol enable prevent the adversary launch de-synchronization attack.

According to the analysis as above, we conclude that this protocol enable resist various attacks. Any adversary cannot obtain the secret values because of $T(\cdot)$ and we utilize its hard problem in this protocol. In order to display the security in detail, we compare our agreement with few recent protocols in Table 3.

Table 3. Security analysis and comparison with other works

Security Requirements	[13]	[14]	Ours
Tag anonymity	Yes	No discussion	Yes
Mutual authentication	Yes	Yes	Yes
Forward secrecy	Yes	Yes	Yes
Resist replay attack	Yes	Yes	Yes
Resist impersonation attack	No	No	Yes
Resist secret disclose attack	No	No	Yes
Resist de-syn attack	No	No	Yes

6 Efficiency Consideration

Compared to RSA and ECC, Chebyshev polynomial computation problem offers smaller key sizes, faster computation, as well as memory, energy and bandwidth savings. Chaotic maps encryption algorithm utilizes the unique semi-group nature of Chebyshev chaotic maps, based on two difficult problems-the chaotic maps discrete logarithm problem and the chaotic maps Diffie-Hellman problem, puts forward a kind of encryption algorithm. From analysis in Table 4, you could find that our protocol achieves high efficient operation.

Table 4. Communication costs comparison

Performance	Cheng et al. [11]	Benssalah et al. [13]	Akgun et al. [14]	Our protocol
Tag	1T*r*+1T*h*+3T*xor*+1T*c*	1T*r*+1T*h*+2T*xor*+1T*c*	1T*r*+2T*xor*+1T*c*	1T*r*+2T*h*+2T*xor*+1T*c*
Reader	1T*r*	1T*r*	1T*r*	1T*r*
Back-end server	1T*r*+1T*h*+2T*xor*+1T*c*	1T*r*+1T*h*+2T*xor*+1T*c*	1T*r*+3T*xor*+1T*c*	1T*r*+1T*h*+2T*xor*+1T*c*
Total	3T*r*+2T*h*+5T*xor*+2T*c*	3T*r*+2T*h*+7T*xor*+2T*c*	3T*r*+5T*xor*+2T*c*	3T*r*+3T*h*+4T*xor*+2T*c*
Rounds	5	5	5	5

Note. T_r : Time for random number generation; T_h : Time for hash function; T_{xor} : Time for XOR operation; T_c : Time for executing $T_n(x) \bmod p$.

7 Conclusion

The emergence of RFID technology has greatly convenience our lives, such as in the library book lending, logistics, public transportation and so on. Therefore, the security should be put into attention. In 2015, Akgun et al. proposed an improved RFID authentication protocol based on chaotic maps to overcome the weakness which appear in the previous schemes. Unfortunately, according to analysis we find that Akgun et al.'s protocol is vulnerable to some malicious attacks, such as secret disclose attack, impersonation attack and de-synchronization attack. Therefore, in this paper, we propose an improved RFID authentication protocol with privacy protection based on chaotic maps hard problem to eliminate these weaknesses. Meanwhile, the efficiency of our improved protocol is also considerable. We expect the agreement to be applied in practice, and we believe that our RFID protocol is suitable for the application in our society.

Acknowledgements

This work is supported by the Liaoning Provincial Natural Science Foundation of China (Grant No. 201602680).

References

[1] R. R. Oliveira, I. M.G. Cardoso, J. L.V. Barbosa, C. A. da Costa, M. P. Prado, An Intelligent Model for Logistics Management based on Geofencing Algorithms and RFID Technology, *Expert Systems with Applications*, Vol. 42, No. 15-16, pp. 6082-6097, September, 2015.

[2] S. D. Kaul, A. K. Awasthi, RFID Authentication Protocol to Enhance Patient Medication Safety, *Journal of Medical Systems*, Vol. 37, No. 6, pp. 1-6, December, 2013.

[3] H.-Y. Chien, C.-H. Chen, Mutual Authentication Protocol for RFID Conforming to EPC Class 1 Generation 2 Standards, *Computer Standards and Interfaces*, Vol. 29, No. 2, pp. 254-259, February, 2007.

[4] C.-L. Chen, Y.-Y. Deng, Conformation of EPC Class 1 Generation 2 Standards RFID System with Mutual Authentication and Privacy Protection, *Engineering Applications of Artificial Intelligence: The International Journal of Intelligent Real-Time Automation*, Vol. 22, No. 8, pp. 1284-1291, December, 2009.

[5] EDN Staff, *Philips Intros RFID Chip for EPCglobal Standard*, Electronic News, 2005.

[6] T.-C. Yeh, Y.-J. Wang, T.-C. Kuo, S.-S. Wang, Securing RFID Systems Conforming to EPC Class 1 Generation 2 Standard, *Expert Systems with Applications*, Vol. 37, No. 12, pp. 7678-7683, December, 2010.

[7] E.-J. Yoon, Improvement of the Securing RFID Systems Conforming to EPC Class 1 Generation 2 Standard, *Expert Systems with Applications*, Vol. 39, No. 1, pp. 1589-1594, January, 2012.

[8] H.-F. Zhu, Sustained and Authenticated of a Universal Construction for Multiple Key Agreement Based on Chaotic Maps with Privacy Preserving, *Journal of Internet Technology*, Vol. 17, No. 5, pp. 869-878, September, 2016.

[9] X.-F. Guo, J.-S. Zhang, Secure Group Key Agreement Protocol based on Chaotic Hash, *Information Sciences*, Vol. 180, No. 20, pp. 4069-4074, October, 2010.

[10] C. Guo, C.-C. Chang, Chaotic Maps-based Password-authenticated Key Agreement Using Smart Cards, *Communications in Nonlinear Science and Numerical Simulation*, Vol. 18, No. 6, pp. 1433-1440, June, 2013.

[11] Z.-Y. Cheng, Y. Liu, C.-C. Chang, S.-C. Chang, Authenticated RFID Security Mechanism based on Chaotic Maps, *Security and Communication Networks*, Vol. 6, No. 2, pp. 247-256, February, 2013.

[12] H.-jun Wang, H. Zhang, J.-x. Li, C. Xu, A (3, 3) Visual Cryptography Scheme for Authentication, *Journal of Shenyang Normal University (Natural Science Edition)*, Vol. 31, No. 3, pp. 397-400, July, 2013.

[13] M. Benssalah, M. Djeddou, K. Drouiche, Security Enhancement of the Authenticated RFID Security Mechanism based on Chaotic Maps, *Security and Communication Networks*, Vol. 7, No. 12, pp. 2356-2372, December, 2014.

[14] M. Akgun, A. O. Bayrak, M. U. Caglayan, Attacks and Improvements to Chaotic Map-based RFID Authentication Protocol, *Security and Communication Networks*, Vol. 8, No. 18, pp. 4028-4040, December, 2015.

Biographies

Hongfeng Zhu obtained his Ph.D. degree in Information Science and Engineering from Northeastern University. Hongfeng Zhu is a full professor of the Kexin software college at Shenyang Normal University. He is also a master's supervisor. He has research interests in wireless networks, mobile computing, cloud computing, social networks, network security and quantum cryptography. Dr. Zhu had published more than 60 international journal papers on the above research fields.

Yan Zhang, 27 years old, an undergraduate from Shenyang Normal University. She has research interests in wireless networks, mobile computing, cloud computing, social networks, network security and quantum cryptography. In the four years of college, after completing her studies, she enjoys reading the book related to this major. Under the guidance of the teacher, she has published five articles in EI journals and three articles in SCI journals.