A Novel Two-party Scheme against Off-line Password Guessing Attacks using New Theorem of Chaotic maps

Hongfeng Zhu
Software College, Shenyang Normal University
No.253, HuangHe Bei Street, HuangGu District, Shenyang, P.C 110034 - China
zhuhongfeng1978@163.com

Received May 9, 2017; revised July 8, 2017; accepted August 1, 2017;
Published December 31, 2017

Abstract

Over the years, more password-based authentication key agreement schemes using chaotic maps were susceptible to attack by off-line password guess attack. This work approaches this problem by a new method--new theorem of chaotic maps: $T_{a+b}(x)+T_{a-b}(x)=2T_a(x)T_b(x),(a>b)$. In fact, this method can be used to design two-party, three-party, even in N-party intelligently. For the sake of brevity and readability, only a two-party instance: a novel Two-party Password-Authenticated Key Agreement Protocol is proposed for resisting password guess attack in this work. Compared with the related literatures recently, our proposed scheme can be not only own high efficiency and unique functionality, but is also robust to various attacks and achieves perfect forward secrecy. For capturing improved ratio of security and efficiency intuitively, the paper firstly proposes a new parameter called security/efficiency ratio(S/E Ratio). The higher the value of the S/E Ratio, the better it is. Finally, we give the security proof and the efficiency analysis of our proposed scheme.

Keywords: Key agreement, mutual authentication, password-guessing attack, chaotic maps

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

https://doi.org/10.3837/tiis.2017.12.027 **ISSN : 1976-7277**

1. Introduction

Mutual authentication key agreement (MAKA) is one of the most important cryptographic components which is used for establishing an authenticated and confidential communication channel. The mutual authentication and the key agreement are impartible and the reasons are: (1) A protocol only has the attribute of key agreement will lead the man-in-the-middle attacks at least, just like the first key agreement scheme Deffie–Hellman (D–H) key agreement [1]. (2) A protocol only has the attribute of mutual authentication will bring about some function loss. For example, you can use mutual authentication scheme for acquiring E-mail service, but you cannot only use mutual authentication scheme for getting Instant Messaging service, because there is no session key to protect transmissive information. Unlike digtal signature needing the third party for arbitration and many other properties, MAKA protocols are only related with the involving participants, so naturally the efficient chaotic cryptosystem is the first candidate.

Many researchers make some comparisons with other cryptosystem systems to find that chaotic system has many advantages, for example, unpredictability, deterministic random-like process and so on. In the past few years, cryptography systems based on chaos theory have been studied widely [2-15], such as two-party AKA protocols [3-5], three-party AKE protocols [6], *N*-party AKE protocols [7], random number generating [8], hash functions [11], symmetric encryption [9], asymmetric encryption [2,10], digtal signature [12], anonymity scheme [13], Multi-server Environment (Centralized Model) [14, 23], Multiple Servers to Server Architecture (Distributed Model) [15].

To further give the better user experience, that using cryptosystem systems to construct password-based MAKA (called PAKA) protocols [3-5, 14, 15, 21-23] are popular recently. But these protocols introduce password as a trust authenticator will lead off-line password guess attack, such as the works [3, 5]. For resisting off-line password guess attack, almost all protocols adopt carefully designed methods with hash, chaotic maps, XOR, symmetric/asymmetric encryption and so on. Some papers [21-23] adopt two or three factors to protect password for avoiding password guess attack, such as smardcard with password (two factors), smardcard, biometric with password (three factors) and so on. But multi-factors authentication will lead to great amount of calculation and increased Cost.

In this paper, we find a new way to solve this problem—new theorem of chaotic maps: $T_{a+b}(x)+T_{a-b}(x)=2T_a(x)T_b(x),(a>b)$. In our scheme, using the transmitting messages, anyone cannot construct a function which only including one input variable *password* and a related output. So in this paper, we give a new instance of two-party PAKA protocol, and based on the two-party instance, it is easy to expand to many application fields, such as three-party environment, smartcard with password environment and so on. The main contribution in the paper is not only the new instance of two-party PAKA protocol, but also by this instance, there is a new method or a new direction for resisting off-line password guess attack.

The rest of the paper is organized as follows: mathematical preliminaries of chaotic maps are given in Section 2. Next, a novel chaotic maps-based password-authentication key agreement scheme is described in Section 3. Then, the security proof and efficiency analysis about our proposed scheme are given in Section 4 and Section 5. This work is finally summarized in Section 6.

2. Mathematical Preliminaries

2.1 Chebyshev chaotic maps

Let n be an integer and let x be a variable with the interval $[-1,1]$. The Chebyshev polynomial [18] $T_n(x):[-1,1]\to[-1,1]$ 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:

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

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

$$T_r(T_s(x))=T_{rs}(x). \quad (2)$$

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

$$T_r(T_s(x))=T_s(T_r(x)). \quad (3)$$

In order to enhance the security, Zhang [19] proved that semi-group property holds for Chebyshev polynomials defined on interval (-∞,+∞). The enhanced Chebyshev polynomials are used in the proposed protocol:

$$T_n(x)=(2xT_{n-1}(x)-T_{n-2}(x))(\mathrm{mod}\ N), \quad (4)$$

where $n\geq 2$, $x\in(-\infty,+\infty)$, and N is a large prime number. Obviously,

$$T_{rs}(x)=T_r(T_s(x))=T_s(T_r(x)). \quad (5)$$

Definition 2.1. Semi-group property of Chebyshev polynomials:

$T_{rs}(x)=T_r(T_s(x))=\cos(r\cos^{-1}(s\cos^{-1}(x)))=\cos(rs\cos^{-1}(x))=T_s(T_r(x))=T_{sr}(x)$.

Definition 2.2. Given *x* and *y*, it is intractable to find the integer *s*, such that $T_s(x)=y$. It is called the Chaotic Maps-Based Discrete Logarithm problem (CMBDLP or CDL).

Definition 2.3. Given *x*, $T_r(x)$ and $T_s(x)$, it is intractable to find $T_{rs}(x)$. It is called the Chaotic Maps-Based Diffie-Hellman problem (CMBDHP or CDH).

2.2 Theorems of Chaotic maps problems [12]

Let P and Q be integers and p be a prime. The general second-order linear recurrence relation is of the form:

$$T_a(x)=P\times T_{a-1}(x)+Q\times T_{a-2}(x) \quad (a\geq 2) \quad (6)$$

Where $T_a(x)\in GF(p)$ for all *a*.

The recurrence relation function of chaotic maps is defined to be Eq. (4), with initial conditions $T_0(x)=1$ and $T_1(x)=x$ It is easy to see that the chaotic maps function is a special type of second-order linear recurrence relation as defined in Eq. (6) with $P=2x$ and $Q=-1$.

Theorem 2.1 Let $f(x)=t^2-2xt+1$ and α,β be two roots of $f(x)$. If $x=1/2(\alpha+\beta)$, then the number of solutions satisfy

$$T_a(x)=\frac{\left(x+\sqrt{x^2-1}\right)^a+\left(x-\sqrt{x^2-1}\right)^a}{2} \bmod p .$$

Proof Since α and β are the roots of the characteristic polynomial $f(x)$ of the recurrence Eq. (1) defined by

$$f(x)=t^2-2xt+1 \tag{7}$$

we get two different solutions from Eq. (7), i.e.

$$\alpha=x+\sqrt{x^2-1}, \quad \beta=x-\sqrt{x^2-1} \tag{8}$$

Assuming c_1 and c_2 are two random numbers, we can get the following properties according to Eq. (6):

$$P\left(c_1\alpha^{n-1}+c_2\beta^{n-1}\right)-Q\left(c_1\alpha^{n-2}+c_2\beta^{n-2}\right)=c_1\alpha^n+c_2\beta^n \tag{9}$$

From this, when $T_0=c_1+c_2$, $T_1=c_1\alpha+c_2\beta$, any recurrence relation of $T_a(x)$ that can satisfy Eq. (6) is of the form $c_1\alpha^n+c_2\beta^n$. So the recurrence relation of $T_a(x)$ is defined as Eq. (10) with the coefficient $c_1=c_2=1/2$:

$$T_a(x)=\frac{\alpha^a}{2}+\frac{\beta^a}{2} \tag{10}$$

Therefore,

$$T_a(x)=\frac{\left(x+\sqrt{x^2-1}\right)^a+\left(x-\sqrt{x^2-1}\right)^a}{2} \bmod p \tag{11}$$ □

Theorem 2.2 If a and b are two positive integers and $a>b$, then $T_{a+b}(x)+T_{a-b}(x)=2T_a(x)T_b(x)$.

Proof Based on Eq. (11), we can prove the theorem 2.2 as follows:

$$T_a(x)\times T_b(x)=\left[\frac{\left(x+\sqrt{x^2-1}\right)^a+\left(x-\sqrt{x^2-1}\right)^a}{2}\right]\times\left[\frac{\left(x+\sqrt{x^2-1}\right)^b+\left(x-\sqrt{x^2-1}\right)^b}{2}\right]$$

$$=\frac{1}{4}\left[\left(x+\sqrt{x^2-1}\right)^{a+b}+\left(x-\sqrt{x^2-1}\right)^{a+b}+\left(x+\sqrt{x^2-1}\right)^a\left(x-\sqrt{x^2-1}\right)^b+\left(x-\sqrt{x^2-1}\right)^a\left(x+\sqrt{x^2-1}\right)^b\right]$$

$$=\frac{1}{4}\left[\begin{array}{l}\left(x+\sqrt{x^2-1}\right)^{a+b}+\left(x-\sqrt{x^2-1}\right)^{a+b}+\dfrac{\left(x+\sqrt{x^2-1}\right)^a\left(x-\sqrt{x^2-1}\right)^b\left(x+\sqrt{x^2-1}\right)^b}{\left(x+\sqrt{x^2-1}\right)^b}\\ +\dfrac{\left(x-\sqrt{x^2-1}\right)^a\left(x+\sqrt{x^2-1}\right)^b\left(x-\sqrt{x^2-1}\right)^b}{\left(x-\sqrt{x^2-1}\right)^b}\end{array}\right]$$

$$
\begin{aligned}
&= \frac{1}{4}\left[\begin{array}{l}\left(x+\sqrt{x^2-1}\right)^{a+b}+\left(x-\sqrt{x^2-1}\right)^{a+b}+\left(x+\sqrt{x^2-1}\right)^{a-b}\left(x-\sqrt{x^2-1}\right)^{b}\left(x+\sqrt{x^2-1}\right)^{b} \\ +\left(x-\sqrt{x^2-1}\right)^{a-b}\left(x+\sqrt{x^2-1}\right)^{b}\left(x-\sqrt{x^2-1}\right)^{b}\end{array}\right] \\
&= \frac{1}{4}\left[\begin{array}{l}\left(x+\sqrt{x^2-1}\right)^{a+b}+\left(x-\sqrt{x^2-1}\right)^{a+b}+\left(x+\sqrt{x^2-1}\right)^{a-b}\left(x^2-\left(x^2-1\right)\right)^{b} \\ +\left(x-\sqrt{x^2-1}\right)^{a-b}\left(x^2-\left(x^2-1\right)\right)^{b}\end{array}\right] \\
&= \frac{1}{4}\left[\left(x+\sqrt{x^2-1}\right)^{a+b}+\left(x-\sqrt{x^2-1}\right)^{a+b}+\left(x+\sqrt{x^2-1}\right)^{a-b}1^{b}+\left(x-\sqrt{x^2-1}\right)^{a-b}1^{b}\right] \\
&= \frac{1}{2}\left[T_{a+b}(x)+T_{a-b}(x)\right] \qquad (12) \qquad \square
\end{aligned}
$$

2.3 Threat Model

The widely accepted security assumptions about password based authentication schemes [16, 17] should be adopted as the threat model.

(1) The user$_i$ holds the uniformly distributed low-entropy password from the small dictionary. The server keeps the private key. At the time of registration, the server sends the personalized security parameters to the user$_i$ by secure channel and the user$_i$ should keep the personalized security parameters safe.

(2) An adversary and a user$_i$ interact by executing oracle queries that enables an adversary to perform various attacks on authentication protocols.

(3) The communication channel is controlled by the adversary who has the capacity to intercept, modify, delete, resend and reroute the eavesdropped messages.

In the password authenticated protocol Π, each participant is either a user $u_i \in U$ or a trusted server S interact number of times (If the two participants are both users, the S may represent a user). Only polynomial number of queries occurs between adversary and the participant's interaction. This enables an adversary to simulate a real attack on the authentication protocol. The possible oracle queries are as follows:

Execute (Π_U^i, Π_S^j): This query models passive attacks against the protocol which is used to simulate the eavesdropping honest execution of the protocol. It prompts an execution of the protocol between the user's instances Π_U^i and server's instances Π_S^j that outputs the exchanged messages during honest protocol execution to A.

Send (Π_U^i, m): This query sends a message m to an instance Π_U^i, enabling adversary A for active attacks against the protocol. On receiving m, the instance Π_U^i continues according to the protocol specification. The message output by Π_U^i, if any, is returned to A.

Reveal (Π_U^i): This query captures the notion of known key security. The instance Π_U^i, upon receiving the query and if it has accepted, provides the session key, back to A.

Corrupt (Π_U^i, m): These queries together capture the notion of two-factor security. The former returns the password of U_i while the latter returns the information stored in the smart card of U_i.

Test(Π_U^i): This query is used for determining whether the protocol achieves authenticated key exchange or not. If Π_U^i has accepted, then a random bit $b \in \{0,1\}$ chosen by the oracle, *A* is given either the real session key if $b=1$, otherwise, a random key drawn from the session key space.

We say that an instance Π_U^i is said to be open if a query Reveal(Π_U^i) has been made by adversary, and unopened if it is not opened. We say that an instance Π_U^i has accepted if it goes into an accept mode after receiving the last expected protocol message.

Definition 4. Two instances Π_U^i and Π_S^i are said to be partnered if the following conditions hold:

- Both Π_U^i and Π_S^i accept;
- Both Π_U^i and Π_S^i share the same session identifications(sid);
- The partner identification for Π_U^i and Π_S^i and vice-versa.

Definition 5. We say an instance Π_U^i is considered fresh if the following conditions are met:

- It has accepted;
- Both Π_U^i and its partner Π_S^i are unopened;
- They are both instances of honest clients.

Definition 6. Consider an execution of the authentication protocol Π by an adversary *A*, in which the latter is given access to the Execute, Send, and Test oracles and asks at most single Test query to a fresh instance of an honest client. Let b' be his output, if $b'=b$, where *b* is the hidden bit selected by the Test oracle. Let *D* be user's password dictionary with size |*D*|. Then, the advantage of *A* in violating the semantic security of the protocol Π is defined more precisely as follows:

$$Adv_{\Pi,D}(A) = [2\Pr[b' = b] - 1]$$

The password authentication protocol is semantically secure if the advantage $Adv_{\Pi,D}(A)$ is only negligibly larger than $O(q_s)/|D|$, where q_s is the number of active sessions.

3. The novel two-party PAKA protocol

In this section, we give a novel chaotic maps-based password-authentication key agreement scheme which consists of three sections: share the password, the novel two-party PAKA, password changing.

3.1 Notations

The concrete notations used hereafter are shown in **Table 1**.

Table 1. Notations

Symbol	Definition
ID_A, ID_B	The identities of the users (Alice and Bob), respectively
PW	The shared password of the users
a, b, c	Random numbers
$(x, T_k(x))$	Public key based on Chebyshev chaotic maps
k	Secret key based on Chebyshev chaotic maps
H	A secure one-way hash function. $H: \{0,1\}^* \rightarrow \{0,1\}^l$ for a constant l
$\|$	Concatenation operation

3.2 Share the password

In this section, we give two main architectures for sharing the password instead of some concrete methods. The two logical architectures for sharing the password are shown in **Fig. 1**. Without loss of generality, let $U = \{Alice, Bob\}$ be a set of two users, S be a trusted server.

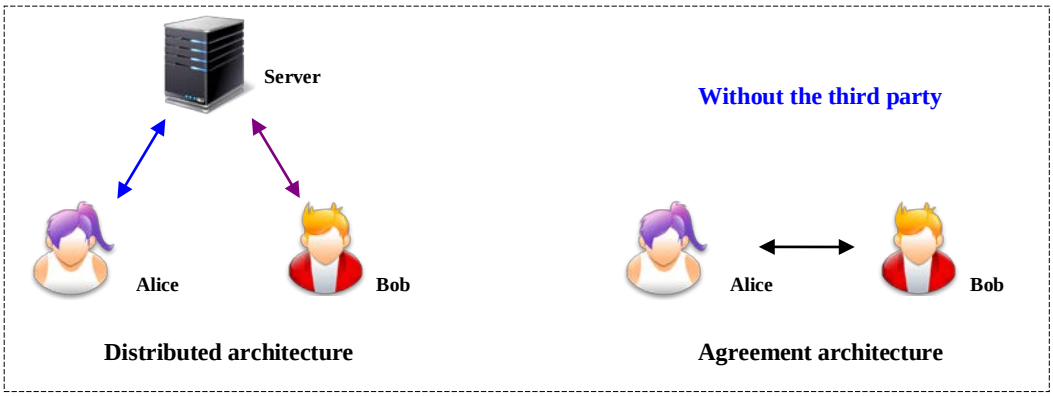

Fig. 1. Two logical architectures for sharing the password

(1) Distributed architecture: The trusted server defines system parameters and generates his private/public key-pair. Then, the trusted server publishes the system parameters and keeps private key secret. Next, each user must register in trusted server before PAKE. Finally, the trusted server cooperates with the registering user to generate the shared password between the registering users.

Due to space limitations, this section just gives an instance for sharing the password in distributed architecture: a) any user must take his/her identities card as the authenticator and transfer it to the server by a secure channel; b) the server uses his private/public key-pair to sign some messages for authenticaing itself; c) after mutual authentication, a user must leave his/her private cell-phone number as a secure receiver for receive any temporary shared password which is sent by the server.

(2) Agreement architecture: In this architecture, there is no the trust third party involved. The two users will exchange the shared password by a secure channel. The main methods are: public-key cryptosystem, phone calls or secure instant messaging software, or exchange password face to face, and so on.

3.3 The novel two-party PAKA

This concrete process is presented in the following **Fig. 2**.

(1) User A → User B: $\{ID_A, T_b(x), E_A, V_A\}$

If Alice wishes to consult some personal issues establish with Bob in a secure way, she will input *password* and choose two random integer numbers a, b $(a > HPW)$. Then, she computes $T_a(x), T_b(x)$, $E_A = T_a(x)T_{HPW}T_b(x)$ and $V_A = T_{a+HPW}(x) + T_{a-HPW}(x)$. After that, Alice sends $\{ID_A, T_b(x), E_A, V_A\}$ to Bob.

(2) User B → User A: $\{E_B, V_B\}$

After receiving the message $\{ID_A, T_b(x), E_A, V_A\}$, Bob firstly must use the shared password to get $T_a(x) = E_A / T_{HPW}T_b(x)$. Next, Bob computes $2T_a(x)T_{HPW}(x)$ and verifies $2T_a(x)T_{HPW}(x) = V_A$?. If above equation holds, that means Alice is a legal user, or Bob will abort this process. After authenticating Alice, Bob chooses a random c $(c > HPW)$ and computes $T_c(x), E_B = T_c(x)T_{HPW}T_b(x)$ and $V_B = T_{c+HPW}(x) + T_{c-HPW}(x)$. Finally Bob computes the session key $SK = H(T_cT_a(x))$ locally and sends $\{E_B, V_B\}$ to Alice.

(3) Because $T_{HPW}T_b(x)$ has already computed before, Alice can get $T_c(x) = E_B / T_{HPW}T_b(x)$ directly. Next, Alice computes $2T_c(x)T_{HPW}(x)$ and verifies $2T_c(x)T_{HPW}(x) = V_B$?. If above equation holds, that means Bob is a legal user, or Alice will abort this process. After authenticating Bob, Alice computes the session key $SK = H(T_aT_c(x))$ locally.

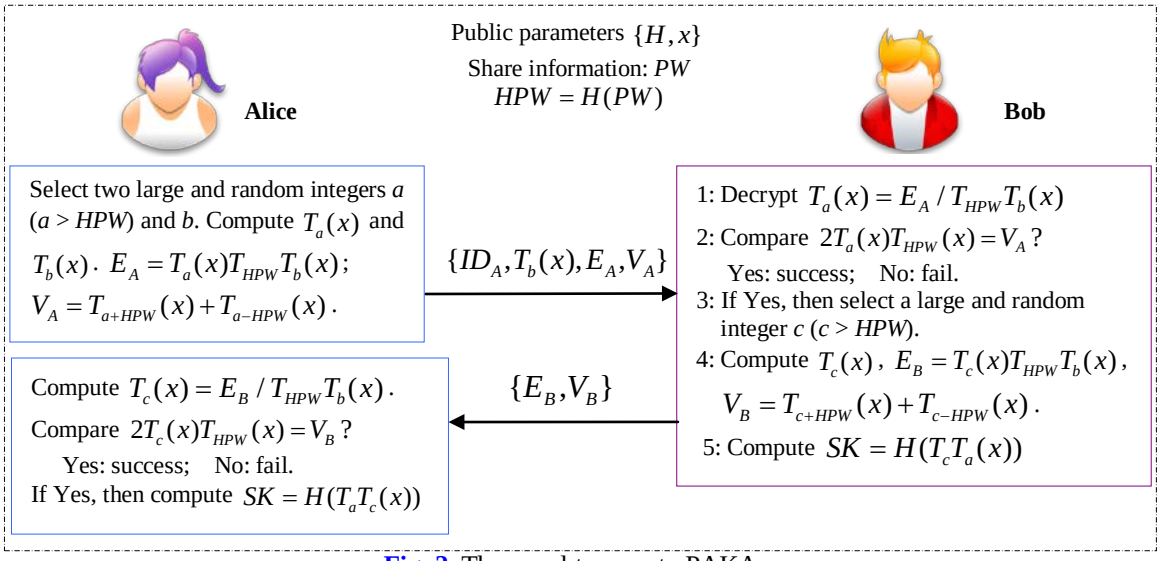

Fig. 2. The novel two-party PAKA

3.4 Password changing

Fig. 3 illustrates the password changing phase.

(1) User A → User B: $\{ID_A, T_b(x), E_A, V_A, C_A\}$

When Alice wants to change her password, she chooses PW' ($HPW' = H(PW')$), two random numbers a,b ($a > HPW$), and computes $T_a(x), T_b(x)$, $E_A = T_a(x)T_{HPW}T_b(x)$, $V_A = (T_{a+HPW}(x) + T_{a-HPW}(x))HPW'$ and $C_A = T_aT_{HPW}(x)PW'$. Then Alice sends $\{ID_A, T_b(x), E_A, V_A, C_A\}$ to Bob.

(2) User B $\rightarrow$ User A: $\{E_B, V_B\}$

Upon receiving $\{ID_A, T_b(x), E_A, V_A, C_A\}$ from Alice, Bob firstly must use the old shared password PW to get $T_a(x) = E_A / T_{HPW}T_b(x)$. Next, Bob computes $T_{HPW}T_a(x)$ to get the new password $PW' = C_A / T_{HPW}T_a(x)$. Then, Bob computes $2T_a(x)T_{HPW}(x)HPW'$ and verifies $2T_a(x)T_{HPW}(x)HPW' = V_A$?. If above equation holds, that means Alice is a legal user, or Bob will abort this process. After authenticating Alice, Bob chooses a random c ($c > HPW$) and computes $T_c(x), E_B = T_c(x)T_{HPW}T_b(x)$ and $V_B = (T_{c+HPW}(x) + T_{c-HPW}(x))HPW'$. Finally Bob uses the new password PW' instead of PW and sends $\{E_B, V_B\}$ to Alice.

(3) Because $T_{HPW}T_b(x)$ has already computed before, Alice can get $T_c(x) = E_B / T_{HPW}T_b(x)$ directly. Next, Alice computes $2T_c(x)T_{HPW}(x)HPW'$ and verifies $2T_c(x)T_{HPW}(x)HPW' = V_B$?. If above equation holds, that means Bob is a legal user, or Alice will abort this process. After authenticating Bob, Alice uses the new password PW' instead of PW.

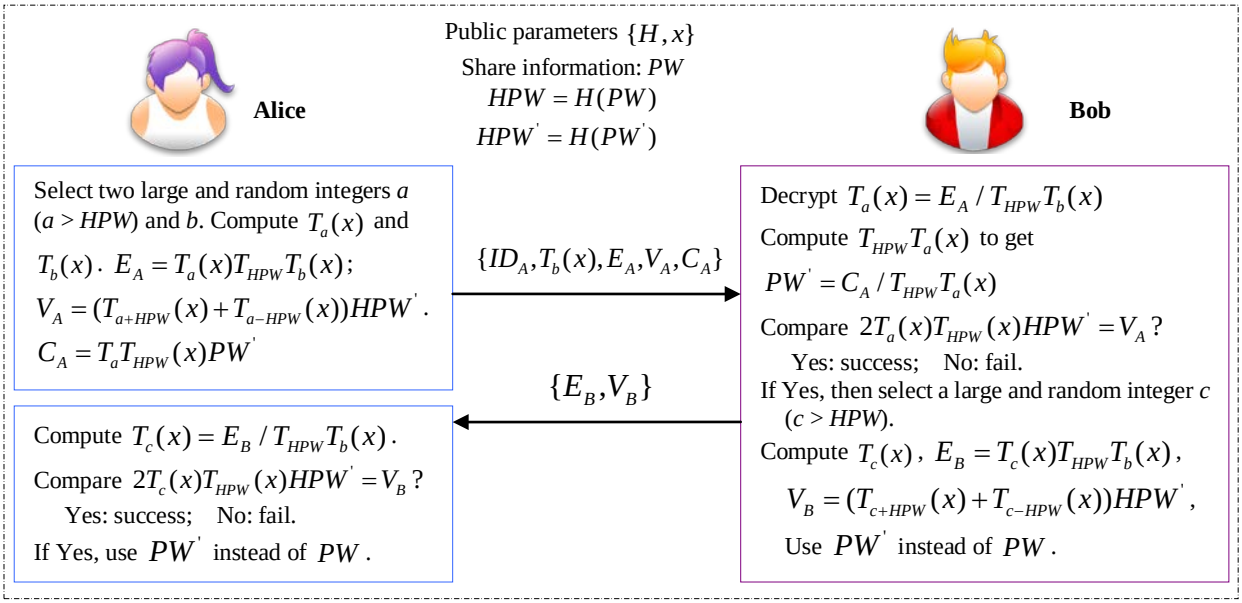

Fig. 3. Password changing phase

4. Security Analysis

4.1 Formal Security Analysis of the Proposed Scheme [16, 17]

First of all, we transform the process of our proposed novel two-party PAKA phase to the

following two simulation Algorithms.

Algorithm 1 Simulation of send query

1: On a query *Send* $(\Pi_U^i, start)$, assume that U_i is in correct state, then we proceed as follows:

2: Choose two numbers $a,b \in_R Z_p^*(a > HPW)$, compute $T_a(x), T_b(x)$, $E_A = T_a(x)T_{HPW}T_b(x)$ and $V_A = T_{a+HPW}(x) + T_{a-HPW}(x)$. This query returns $< ID_A, T_b(x), E_A, V_A >$ as answer.

3: On a query *Send* $(U_B, < ID_A, T_b(x), E_A, V_A >)$, assume that U_B is in correct state, we continue as follows:

4: Compute $T_a(x) = E_A / T_{HPW}T_b(x)$.

5: **if** $2T_a(x)T_{HPW}(x) \neq V_A$ **then**

6: Reject the message.

7: **else** compute $B^* = H(ID_i \| k)$ and $H(B^* \| C_{i_1})$.

8: **if** $H(B^* \| C_{i_1}) \neq C_{i_2}$ **then**

9: Reject the message.

10: **else** select a large and random integer $c \in_R Z_p^*$ ($c > HPW$), and compute $T_c(x)$, $E_B = T_c(x)T_{HPW}T_b(x)$ and $V_B = T_{c+HPW}(x) + T_{c-HPW}(x)$ and $SK = H(T_cT_a(x))$. The query $< E_B, V_B >$ returns as answer.

11: **end if**

12: **end if**

13: On a query *Send* $< E_B, V_B >$, assume that U_A is in correct state, then we proceed as follows:

14: Compute $T_c(x) = E_B / T_{HPW}T_b(x)$.

15: **if** $2T_c(x)T_{HPW}(x) \neq V_B$ **then**

16: Reject the message.

17: **else** compute $SK = H(T_aT_c(x))$.

18: **end if**

Algorithm 2 Simulation of Execute query

On a query *Reveal* (Π_U^i), we proceed as follows:

If The instance Π_U^i is accepted **then**

This query answered the session key.

End if

Theorem 1 L*et D be a uniformly distributed dictionary of possible passwords with size |D|, Let P be the improved authentication protocol described in Algorithm* 1 *and* 2*. Let A be an adversary against the semantic security within a time bound t. Suppose that CDH assumption holds, then,*

$$Adv_{\Pi,D}(A) \leq \frac{2q_h^2}{2^{l+1}} + \frac{(q_s + q_e)^2}{p^2} + 2q_h Adv_G^{cdh}(A) + \frac{2q_h}{p} + \frac{2q_s^2}{D}$$

where $Adv_G^{cdh}(A)$ *is the success probability of A of solving the chaotic maps-based computational Diffie–Hellman problem.* q_s *is the number of Send queries,* q_e *is the number of Execute queries and* q_h *is the number of random oracle queries.*

Proof This proof defines a sequence of hybrid games, starting at the real attack and ending up in game where the adversary has no advantage. For each game $G_i (0 \leq i \leq 4)$, we define an event $Succ_i$ corresponding to the event in which the adversary correctly guesses the bit *b* in the test-query.

Game G_0 This game correspond to the real attack in the random oracle model. In this game, all the instances of U_A and U_B are modeled as the real execution in the random oracle. By definition of event $Succ_i$ in which the adversary correctly guesses the bit b involved in the Test-query, we have

$$Adv_{\Pi,D}(A) = 2 \mid \Pr[Succ_0] - \frac{1}{2} \mid \tag{1}$$

Game G_1 This game is identical to the game G_0, except that we simulate the hash oracles h by maintaining the hash lists $List_h$ with entries of the form (*Inp*, *Out*). On hash query for which there exists a record (*Inp*, *Out*) in the hash list, return *Out*. Otherwise, randomly choose $Out \in \{0,1\}$, send it to A and store the new tuple (*Inp*, *Out*) into the hash list. The Execute, Reveal, Send, Corrupt, and Test oracles are also simulated as in the real attack where the simulation of the different polynomial number of queries asked by A. From the viewpoint of A, we identify that the game is perfectly indistinguishable from the real attack. Thus, we have

$$\Pr[Succ_1] = \Pr[Succ_0] \tag{2}$$

Game G_2 In this game, the simulation of all the oracles is identical to game G_1 except that the game is terminated if the collision occurs in the simulation of the partial transcripts $\{E_A, V_A\}$ or $\{E_B, V_B\}$ and on hash values. According to the birthday paradox, the probability of collisions of the simulation of hash oracles is at most $q_h^2 / 2^{l+1}$. Similarly, the probability of collisions in the transcripts simulations is at most $\frac{(q_h + q_e)^2}{2p^2}$. Since a,b,c were selected uniformly at random. Thus, we have

$$\Pr[Succ_2] - \Pr[Succ_1] = \frac{q_h^2}{2^{l+1}} + \frac{(q_h + q_e)^2}{2p^2} \tag{3}$$

Game G_3 In this game, the session key is guessed without asking the corresponding oracle h so that it become independent of password and ephemeral keys a,c which are protected by the chaotic maps-based computational Diffie–Hellman problem. We change the way with earlier game unless A queries h on the common value $SK = H(T_a T_c(x))$. Thus, $Adv_G^{cdh}(A) \geq \frac{1}{q_h} \mid \Pr[Succ_3] - \Pr[Succ_2] \mid - \frac{1}{p}$, that is, the difference between the game G_3 and the game G_2 is as follows:

$$\mid \Pr[Succ_3] - \Pr[Succ_2] \mid \leq q_h Adv_G^{cdh}(A) + \frac{q_h}{p} \tag{4}$$

Game G_4 This game is similar to the game G_3 except that in Test query, the game is aborted if A asks a hash function query with $SK = H(T_a T_c(x))$. A gets the session key SK by hash function query with probability at most $\frac{q_h^2}{2^{l+1}}$. Hence, we have

$$\mid \Pr[Succ_4] - \Pr[Succ_3] \mid \leq \frac{q_h^2}{2^{l+1}} \tag{5}$$

If A does not make any h query with the correct input, it will not have any advantage in distinguishing the real session key from the random once. Moreover, if the corrupt query Corrupt (U, 2) is made that means the password-corrupt query Corrupt (U, 1) is not made, and the password is used once in local computer to authenticate user for getting some important information and no more used in the process of the protocol Π. Thus, the probability of A

made off-line password guessing attack is at most $\frac{q_s^2}{D}$. Combining the Eqs. 1-5 one gets the announced result as:

$$Adv_{\Pi,D}(A) \le \frac{2q_h^2}{2^{l+1}} + \frac{(q_s+q_e)^2}{p^2} + 2q_h Adv_G^{cdh}(A) + \frac{2q_h}{p} + \frac{2q_s^2}{D}$$

4.2 Further Security Discussion of the Proposed Scheme

Proposition 1 *The proposed scheme could resist password guessing attack.*

Proof In this attack, an adversary may try to guess a legal user U_i's password PW_i using the transmitted messages. Password guessing attack can only crack a function with one low entropy variable (password), so if we at least insert one large random variable which can resist this attack. Based on the new theorem of chaotic maps $T_{a+b}(x)+T_{a-b}(x)=2T_a(x)T_b(x),(a>b)$, and our protocol has some high entropy variables a,b,c with *HPW* to makeup two kinds of functional expressions $\{E_A,V_A\}$ or $\{E_B,V_B\}$.

- For $E_A = T_a(x)T_{HPW}T_b(x)$, there are two large random variables (*a*, *b*) to covered the low entropy variable (*password*). Based on CDL problem, anyone cannot compute *b* by $T_b(x)$. And then based on CDH problem, you cannot compute $T_{HPW}T_b(x)$. Furthermore, the $T_a(x)$ is secret information for all the process of our scheme. Finally we can get a conclusion that an adversary cannot guess three input variables ($password^{'},a^{'},b^{'}$) to construct a function $T_{a^{'}}(x)T_{HPW^{'}}T_{b^{'}}(x)=E_A$? for judging the equation is equal or not, because *a*, *b* are two large and randomly selected values. It has the same proof process for E_B.
- For $V_A = T_{a+HPW}(x)+T_{a-HPW}(x)$, there is a large random variable (*a*) to covered the low entropy variable (*password*). Based on CDH problem, only the party owns the *HPW* can compute $T_{HPW}T_b(x)$ for getting $T_a(x)$ further. Finally we can get a conclusion that an adversary cannot guess two input variables ($password^{'},a^{'}$) to construct a function $T_{a+HPW}(x)+T_{a-HPW}(x)=V_A$? for judging the equation is equal or not, because *a* is a large and randomly selected value. On the other side, the legal party can authenticate this message by $2T_a(x)T_{HPW}(x)=V_A$? It has the same proof process for V_B.
- Combine $\{ID_A,T_b(x),E_A,V_A,E_B,V_B\}$ to launch password guessing attack. Any combination of these messages $\{ID_A,T_b(x),E_A,V_A,E_B,V_B\}$ cannot construct a function that only one low input variable (*password* or *HPW*). Additionally, no message part is repeated in consecutive communications. This shows that our scheme can resist password guessing attack.

Proposition 2 *The proposed scheme could resist stolen verifier attack.*

Proof In the proposed scheme, any party stores nothing about the legal users' information. All the en/decrypted messages can be deal with the user's password which is stored in the user's brain, so the proposed scheme withstands the stolen verifier attack.

Proposition 3 *The proposed scheme could withstand replay and man-in-the-middle attacks*.

Proof The verification messages include the temporary random numbers. More important thing is that all the temporary random numbers are protected by CDH problem in chaotic maps which only can be uncovered by the legal users (using *HPW*). So our proposed scheme resists the replay and man-in-the-middle attacks.

Proposition 4 *The proposed scheme could resist user impersonation attack.*

Proof In such an attack, an adversary may try to masquerade as a legitimate user U_i to cheat another legitimate user. For any adversary, there are two ways to carry this attack:

- The adversary may try to launching the replay attack. However, the proposed scheme resists the replay attack.
- The adversary A may try to generate a valid authenticated message $\{ID_A, T_b(x), E_A, V_A\}$ for two random values *a*, *b*. Howerer, the adversary cannot compute E_A, V_A as computation of E_A, V_A requires *HPW* which is only known to legal users.

This shows that the proposed scheme resist user impersonation attack.

Proposition 5 *The proposed scheme could withstand server impersonation attack.*

Proof In this attack, an adversary can masquerade as the server and try to respond with a valid message to the user U_i. For any adversary, this attack cannot be happened because there is no any server involved in the proposed scheme.

Proposition 6 *The proposed scheme could support mutual authentication.*

Proof In our scheme, the user B verifies the authenticity of user A's request by verifying the condition $2T_a(x)T_{HPW}(x) = V_A$? during the proposed phase. To compute E_A, V_A, the shared password is needed. Therefore, an adversary cannot forge the message. Additionally, E_A, V_A includes large random nubmers *a* and *b*, the adversary cannot replay the old message. This shows that the user B can correctly verify the message source. It is the same way for the user A authenticating the user B. Hence, mutual authentication can successfully achieve in our scheme.

Proposition 7 *The proposed scheme could have Key freshness property*.

Proof Note that in our scheme, each established session key $SK = H(T_aT_c(x))$ includes random values *a* and *c*. The unique key construction for each session shows that proposed scheme supports the key freshness property.

Proposition 8 *The proposed scheme could have known key secrecy property.*

Proof In our scheme, if a previously established session key $SK = H(T_aT_c(x))$ is compromised, the compromised session key reveals no information about other session keys due to following reasons:

- Each session key is hashed with one-way hash function. Therefore, no information can be retrieved from the session key.
- Each session key includes two nonces, which ensures different key for each session.

Since no information about other established group session keys from the compromised session key is extracted, our proposed scheme achieves the known key secrecy property.

Proposition 9 *The proposed scheme could have forward secrecy.*

Proof Forward secrecy states that compromise of a legal user's long-term secret key does not become the reason to compromise of the established session keys. In our proposed scheme, the session key has not included the user's long-term secret key: Password. This shows that our scheme preserves the forward secrecy property.

Proposition 10 *The proposed scheme could have perfect forward secrecy.*

Proof A scheme is said to support perfect forward secrecy, if the adversary cannot compute the established session key, using compromised secret key k of any server. The proposed scheme achieves perfect forward secrecy. In our proposed scheme, the session key has not included the server's long-term secret key k because there is no any server involved. This shows that our scheme provides the perfect forward secrecy property.

From the **Table 2**, we can see that the proposed scheme can provide resistance to guessing attacks, perfect forward secrecy and so on. As a result, the proposed scheme is more secure and has much functionality compared with the recent related scheme.

Table 2. Security of our proposed protocol

Security requirements	**[3](2010)**	**[4](2015)**	**[5](2015)**	**Our Proposed Scheme**
Authentication	Mutual	Mutual	Mutual	Mutual
Update password phase	No	No	No	**YES**
forward secrecy property	Yes	Yes	Yes	Yes
Perfect forward secrecy	Yes	Yes	Yes	Yes
Resistance to stolen-verifier attack	No	No	No	**YES**
Resistance to guessing attacks (On-line or off-line) (Including Prevent Password Guessing Attacks for privileged-insider or for any adversary)	Yes	Yes	Yes	Yes
Resistance to man-in-the-middle attack and replay attack	Yes	Yes	Yes	Yes
Resistance to impersonation attack	Yes	Yes	Yes	Yes
Key freshness property	Yes	Yes	Yes	Yes
Known key secrecy property	Yes	Yes	Yes	Yes
Formal security proof	Yes	No	No	**YES**
New Theorem of chaotic maps	No	No	No	**YES**
Notes: Yes/No: Support/Not support				

5. Efficiency Analysis

Compared to RSA and ECC, Chebyshev polynomial computation problem offers smaller key sizes, faster computation, as well as memory, energy and bandwidth savings. To be more precise, on an Intel Pentium4 2600 MHz processor with 1024 MB RAM, where n and p are 1024 bits long, the computational time of a one-way hashing operation, a symmetric encryption/decryption operation, an elliptic curve point multiplication operation and Chebyshev polynomial operation is 0.0005s, 0.0087s, 0.063075s and 0.02102s separately [20]. Moreover, the computational cost of XOR operation could be ignored when compared with other operations. **Table 3** shows performance comparisons between our proposed scheme and the literatures of [3-5]. we sum up these formulas into one so that it can reflect the relationship among the time of algorithms intuitively. $T_p \approx 10T_m \approx 30T_c \approx 72.6T_s \approx 1263.24T_h$, where: T_p:

Time for bilinear pair operation, T_m: Time for a point scalar multiplication operation, T_c: The time for executing the $T_n(x)$ mod p in Chebyshev polynomial, T_s: Time for symmetric encryption algorithm, T_h: Time for Hash operation.

For capturing integrate performance for both security and efficiency at the same time, the paper firstly proposes a new parameter called security/efficiency ratio(S/E Ratio), where Efficiency = Computation × Rounds. The value of S/E Ratio is higher that means better. In S/E Ratio, molecule represents security which can be divided two parts: **(1)** If the protocol can resist one kind of attack, the value will plus 1; or the value is zero. **(2)** If the protocol can provide the formal security proof or BAN logic analysis, the value will plus 2; or the value is zero. In S/E Ratio, denominator represents efficiency which is the specific time consumption approximately. As in **Table 2** and **Table 3**, we can see that the value of S/E Ratio is more than the values of the related literatures (see **Fig. 4**), so we can draw a conclusion that the proposed scheme has achieved the improvement of both efficiency and security.

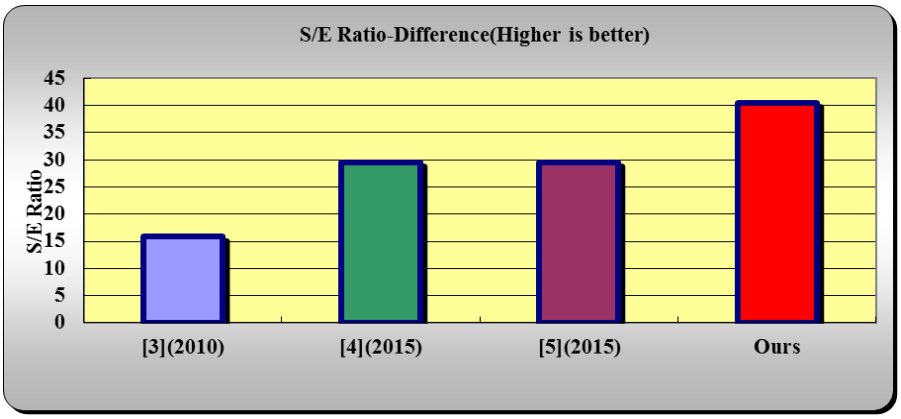

Fig. 4. Comparison of S/E Ratio

Table 3. Comparisons between our proposed scheme and the related literatures

Protocols (Authentication phase)		**[3] (2010)**	**[4] (2015)**	**[5] (2015)**	**Ours**
Computation	$User_A$	$11T_h + 2T_c + 6T_{xor}$	$6T_h + 2T_c + 1T_{xor}$	$6T_h + 2T_c + 3T_{xor}$	$1T_h + 3T_c$
	Server or $User_B$	$11T_h + 2T_c + 5T_{xor}$	$6T_h + 2T_c + 1T_{xor}$	$6T_h + 2T_c + 4T_{xor}$	$1T_h + 4T_c$
	Total	$22T_h + 4T_c + 11T_{xor}$ ≈190.432 T_h	$12T_h + 4T_c + 2T_{xor}$ ≈180.432 T_h	$12T_h + 4T_c + 7T_{xor}$ ≈180.432 T_h	$2T_h + 7T_c$ ≈296.756T_h
Communication	Messages	6	2	3	2
	rounds	6	3	3	2
S/E Ratio (security/efficiency) We assume T_h ≈ 0.5ms (related with Table2) Efficiency = Computation × Rounds		9/(190.432 T_h × 6) ≈15.754	8/(180.432 T_h × 3) ≈29.559	8/(180.432 T_h × 3) ≈29.559	12/(296.756 T_h × 2) ≈40.437
Design	Concise design	No	No	Yes	Yes
	Number of nonces	4	3	4	3
	Model	Random Oracle	Random Oracle	Random Oracle	Random Oracle

T_h: Time for Hash operation T_{xor}: Time for XORed operation
T_c: The time for executing the $T_n(x)$ mod p in Chebyshev polynomial using the algorithm in literature [20]

6. Conclusion

The study presented a novel Two-party Password-Authenticated Key Agreement Protocol using a new theorem of chaotic maps. After giving the proof process of the theorem, the paper sets an instance in detail. Subsequently, we firstly propose a new parameter called security/efficiency ratio(S/E Ratio) for capturing integrate performance for both security and efficiency simultaneously. The security analysis and performance analysis of our new scheme demonstrates that it is secure and efficient one-round PAKA scheme by the new theorem of chaotic maps which will lead to many new schemes arise in the future. Next, the proposed protocol in three aspects will be extended: (1) Bringing in the smart card or biometric to strength of the security level. (2) From the view of functionality, it is meaningful to research the fairness or entanglement and so on. (3) From the perspective of complex, diversified algorithms, especially for constructing new cryptocurrency/blockchain, are our interests.

References

[1] Whitfield Diffie, Martin E.Hellman, "New directions in cryptography," *IEEE Trans. Inf. Theory*, vol. 22, no. 6, pp. 644-654, November, 1976. Article (CrossRef Link).
[2] M. S. Baptista, "Cryptography with chaos," *Physics Letters A*, vol. 240, no. 1, pp. 50-54, Mar. 1998. Article (CrossRef Link).
[3] X.F.Guo and J.Zhang, "Secure group key agreement protocol based on chaotic Hash," *Information Sciences*, vol. 180, no. 20, pp. 4069-4074, Otc. 2010. Article (CrossRef Link).
[4] Tian-Fu Lee, "Enhancing the security of password authenticated key agreement protocols based on chaotic maps," *Inf. Sci.* vol. 290, pp. 63–71 , January, 2015. Article (CrossRef Link).
[5] Yu Liu, Kaiping Xue, "An improved secure and efficient password and chaos-based two-party key agreement protocol," *Nonlinear Dyn.* vol. 84, no. 2, pp. 549-557, November, 2015. Article (CrossRef Link).
[6] Hongfeng Zhu, "A Provable One-way Authentication Key Agreement Scheme with User Anonymity for Multi-server Environment," *KSII Transactions on Internet & Information Systems*, vol. 9, no. 2, pp. 811-829, Feb. 2015. Article (CrossRef Link).
[7] Hongfeng Zhu, "Sustained and Authenticated of a Universal Construction for Multiple Key Agreement Based on Chaotic Maps with Privacy Preserving," *Journal of Internet Technology*, vol. no.5, pp. 1-10, September, 2016. Article (CrossRef Link).
[8] Özkaynak Fatih, "Cryptographically secure random number generator with chaotic additional input," *Nonlinear Dynamics*, vol. 78, no. 3, pp. 2015-2020, Jul. 2014. Article (CrossRef Link).
[9] Chen Jianyong, J. Zhou, and K. W. Wong. "A Modified Chaos-Based Joint Compression and Encryption Scheme." *IEEE Transactions on Circuits & Systems II Express Briefs*, vol. 58, no. 2, pp. 110-114, Feb. 2011. Article (CrossRef Link).
[10] P. Bergamo, P. DArco, A. De Santis, and L. Kocarev, "Security of public-key cryptosystems based on Chebyshev polynomials," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 7, pp. 1382–1393, Jul. 2005. Article (CrossRef Link).
[11] S.J. Xu, X.B. Chen, R. Zhang, Y.X. Yang, and Y.C. Guo, "An improved chaotic cryptosystem based on circular bit shift and XOR operations," *Physics Letters A*, vol. 376, no. 10–11, pp. 1003–1010, Feb. 2012. Article (CrossRef Link).
[12] K. Chain and W.C. Kuo, "A new digital signature scheme based on chaotic maps," *Nonlinear Dynamics*, vol. 74, no. 4, pp. 1003–1012, Aug. 2013. Article (CrossRef Link).
[13] Z. Tan, J. Ning, Y. Liu, X. Wang, G. Yang, and W. Yang, "ECRModel: An Elastic Collision-Based Rumor-Propagation Model in Online Social Networks," *IEEE Access*, vol. 4, pp. 6105–6120, September, 2016. Article (CrossRef Link).

[14] Hongfeng Zhu, "A provable privacy-protection system for multi-server environment," *Nonlinear Dynamics*, vol. 82, no. 1–2, pp. 835–849, Jun. 2015.Article (CrossRef Link).
[15] Hongfeng Zhu, "Flexible and Password-Authenticated Key Agreement Scheme Based on Chaotic Maps for Multiple Servers to Server Architecture," *Wireless Personal Communications*, vol. 82, no. 3, pp. 1697–1718, Jan. 2015. Article (CrossRef Link).
[16] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol.29, no. 2, pp.198–208, Oct. 1981. Article (CrossRef Link).
[17] S. H. Islam, "Provably secure dynamic identity-based three-factor password authentication scheme using extended chaotic maps," *Nonlinear Dynamics*, vol. 78, no. 3, pp. 2261–2276, Jul. 2014. Article (CrossRef Link).
[18] X. Wang and J. Zhao, "An improved key agreement protocol based on chaos," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 12, pp. 4052–4057, Dec. 2010. Article (CrossRef Link).
[19] L. Zhang, "Cryptanalysis of the public key encryption based on multiple chaotic systems," *Chaos, Solitons & Fractals*, vol. 37, no. 3, pp. 669–674, Aug. 2008. Article (CrossRef Link).
[20] L. Kocarev and S. Lian, *Chaos-Based Cryptography*. Springer Berlin Heidelberg, 2011. Article (CrossRef Link).
[21] D. Mishra and S. Mukhopadhyay, "Cryptanalysis of Pairing-Free Identity-Based Authenticated Key Agreement Protocols," *Lecture Notes in Computer Science*, vol. 8303, pp. 247–254, 2013. Article (CrossRef Link).
[22] D. Mishra, A. K. Das, A. Chaturvedi, and S. Mukhopadhyay, "A secure password-based authentication and key agreement scheme using smart cards,"*Journal of Information Security and Applications*, vol. 23, pp. 28–43, Aug. 2015. Article (CrossRef Link).
[23] J. Srinivas, S. Mukhopadhyay, and D. Mishra, "A Self-Verifiable Password Based Authentication Scheme for Multi-Server Architecture Using Smart Card," *Wireless Personal Communications*, pp. 1-25, May, 2017. Article (CrossRef Link).

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 and international conference papers on the above research fields.