Journal of Information Hiding and Multimedia Signal Processing ©2017 ISSN 2073-4212
Ubiquitous International Volume **8**, Number **1**, January **2017**

A Novel One-round Scheme against Off-line Password Guessing Attacks for Three-party Instance

Yu Bai and Dongmin Jiang

College of Electronic Information Engineering, Shenyang Aerospace University
No.37, DaoYi South Street, ShenBei New District, Shenyang, P.C 110136, China
1779683174@qq.com; 2834861086@qq.com

Received August, 2016; revised November, 2016

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 designed in two-party, three-party, even in N-party intelligently. For the sake of brevity and readability, only a three-party instance: a novel Three-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 not only own high efficiency and unique functionality, but is also robust to various attacks and achieves perfect forward secrecy. Finally, we give the security proof and the efficiency analysis of our proposed scheme.*
Keywords: Key agreement, Mutual authentication, Password-guessing attack, Chaotic maps

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. For achieving security, efficiency and convenience at the same time, another key element-password should be involved. Password-authenticated key agreement protocol (PAKA) allows communicating parties to authenticate each other via insecure network using their human memorable passwords (low-entropy) and establishes a secure session key for their subsequent communications.

On the one side, we should a secure and efficient algorithm to design PAKA. 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, 21–23], such as two-party AKA protocols [3, 4], three-party AKE protocols [5, 6], N-party AKE protocols [7], random number generating [8], symmetric encryption [9], asymmetric encryption [2, 10], hash functions [11], digtal signature [12], anonymity scheme [13], Multi-server Environment (Centralized Model) [14], Multiple Servers to Server Architecture (Distributed Model) [15].

On the other side, PAKA schemes have a fatal weakness: these protocols introduce password as a trust authenticator will lead off-line password guess attack, such as the works [3]. For resisting off-line password guess attack, almost all protocols adopt carefully designed methods with hash, chaotic maps, XOR, symmetric encryption and so on. 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 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:

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

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

The first few Chebyshev polynomials are:

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

One of the most important properties is that Chebyshev polynomials are the so-called semi-group property which establishes that

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

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

$$T_r(T_s(x)) = T_s(T_r(x)). \tag{3}$$

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

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

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

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

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:

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

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.

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

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:

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

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 $b \in \{0, 1\}$; 1g 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 2.4. *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 2.5. *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 2.6. *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 three-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, ID_C	The identities of the users (Alice, Bob and Carl), respectively
PW	The shared password of the users
a, a', b, b', c, 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. There are two logical architectures for sharing the password. Without loss of generality, let $U = \{Alice, Bob, Carl\}$ be a set of three users, S be a trusted server.

(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.

(2) Agreement architecture: In this architecture, there is no the trust third party involved. The three 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 three-party PAKA.** This concrete process is presented in the following **Fig.1**.

(1) Round 1:

User A $\rightarrow$ User B and User C: $m_A = \{ID_A, T_{a'}(x), E_A, V_A\}$;

User B $\rightarrow$ User A and User C: $m_B = \{ID_B, T_{b'}(x), E_B, V_B\}$;

User C $\rightarrow$ User A and User B: $m_C = \{ID_C, T_{c'}(x), E_C, V_C\}$;

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

(2) Round 2: Local computation

For Alice: After receiving the messages $\{m_B, m_C\}$, Alice firstly must use the shared password to get $T_b(x) = E_B/T_{HPW}T_{b'}(x)$ and $T_c(x) = E_C/T_{HPW}T_{c'}(x)$. Next, Alice computes $4T_b(x)T_c(x)(T_{HPW}(x))^2$ and verifies $4T_b(x)T_c(x)(T_{HPW}(x))^2 = V_BV_C$?. If above equation holds, which means Bob and Carl are two legal users, or Alice will abort this process. After authenticating Bob and Carl, Alice computes the session key $SK = T_{H(T_a(x)T_b(x)T_c(x))}(x)$ locally. Bob and Carl will do the similar processes.

(3) Correctness of $4T_b(x)T_c(x)(T_{HPW}(x))^2 = V_BV_C$

Proof: Based on the **Theorem 2.2** $T_{a+b}(x) + T_{a-b}(x) = 2T_a(x)T_b(x)$, we have

$$4T_b(x)T_c(x)(T_{HPW}(x))^2 = 2T_b(x)T_{HPW}(x)2T_c(x)T_{HPW}(x)$$
$$= (T_{b+HPW}(x) + T_{b-HPW}(x))(T_{c+HPW}(x) + T_{c-HPW}(x)) = V_BV_C$$

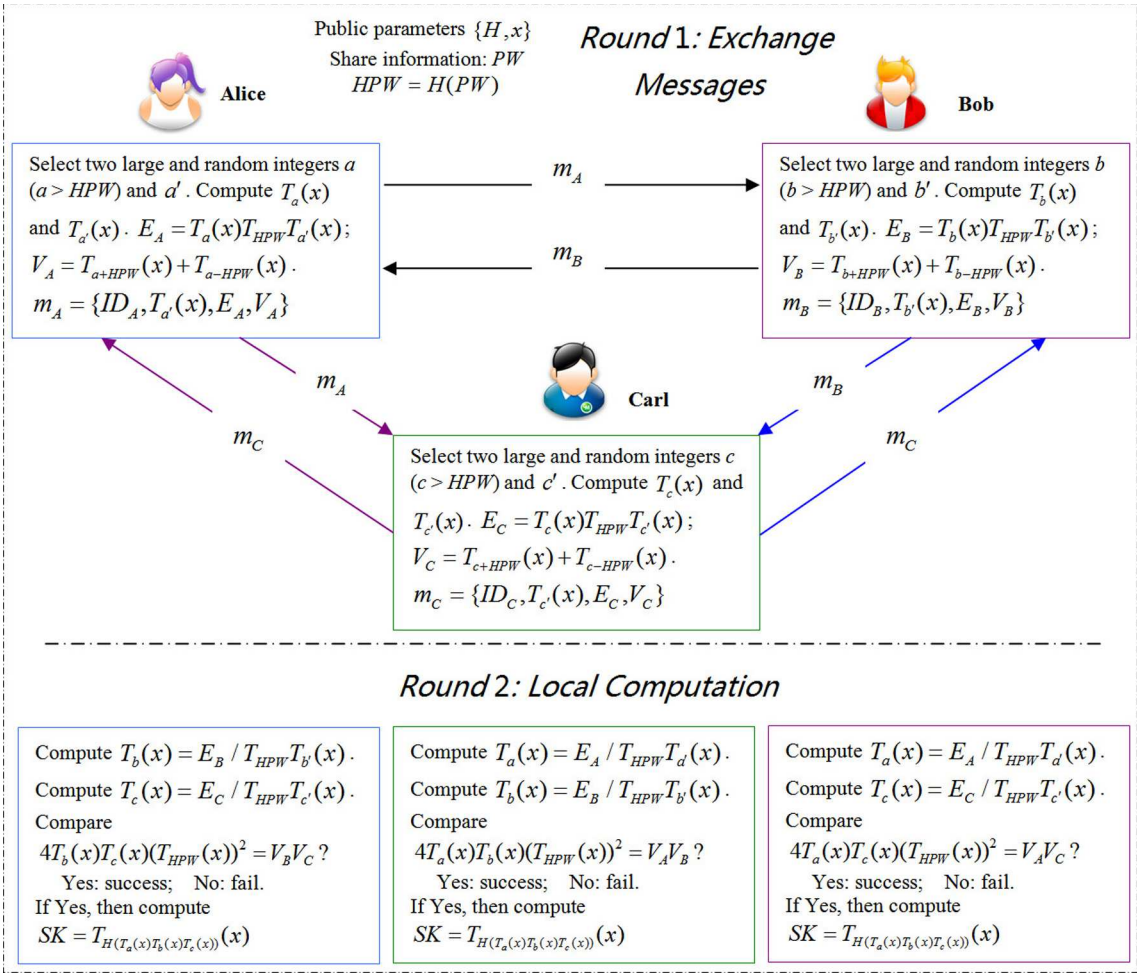

FIGURE 1. The novel three-party PAKA

3.4. **Password changing. Fig.2** illustrates the password changing phase.

(1) User A → User B and User C: $m_A = \{ID_A, T_{a'}(x), E_A, C_A, V_A\}$;

We assume that Alice is the sponsor of changing password, and she chooses PW' $(HPW' = H(PW'))$, two random numbers $a, a'(a > HPW)$, and computes $T_a(x), T_{a'}(x)$, $E_A = T_a(x)T_{HPW}T_{a'}(x)$, $C_A = T_aT_{HPW}(x)PW'$ and $V_A = (T_{a+HPW}(x)+T_{a-HPW}(x))HPW'$. Then Alice sends $m_A = \{ID_A, T_{a'}(x), E_A, C_A, V_A\}$ to Bob and Carl.

(2) User B → User A: $m_B = \{ID_B, E_B, V_B\}$; **User C → User A**: $m_C = \{ID_C, E_C, V_C\}$;

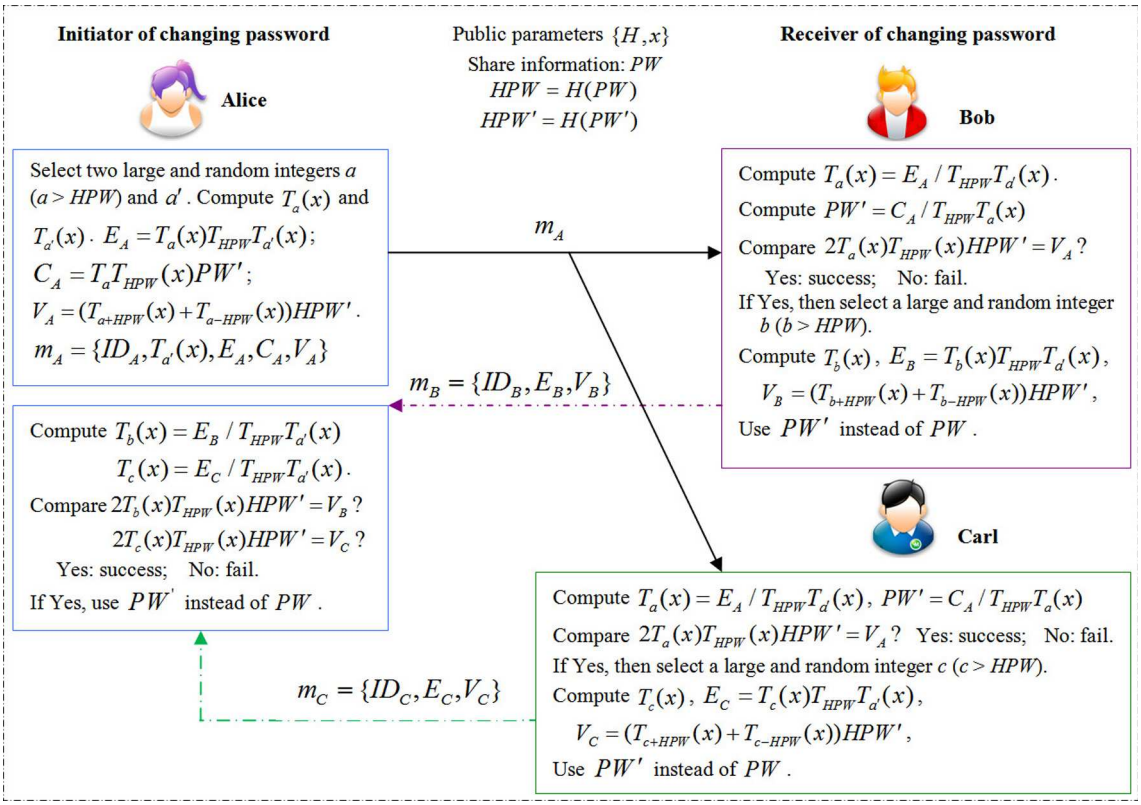

FIGURE 2. Password changing phase

For Bob: Upon receiving $m_A = \{ID_A, T_{a'}(x), E_A, C_A, V_A\}$ from Alice, Bob firstly must use the old shared password PW to get $T_a(x) = E_A/T_{HPW}T_{a'}(x)$. 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 $b(b > HPW)$ and computes $T_b(x), E_B = T_b(x)T_{HPW}T_{a'}(x)$ and $V_B = (T_{b+HPW}(x) + T_{b-HPW}(x))HPW'$. Finally Bob uses the new password PW' instead of PW and sends $m_B = \{ID_B, E_B, V_B\}$ to Alice.

Carl will do the similar processes.

(3) Because $T_{HPW}T_{a'}(x)$ has already computed before, Alice can get $T_b(x) = E_B/T_{HPW}T_{a'}(x)$ and $T_c(x) = E_C/T_{HPW}T_{a'}(x)$ directly. Next, Alice computes $2T_b(x)T_{HPW}(x)HPW'$, $2T_c(x)T_{HPW}(x)HPW'$ and verifies $2T_b(x)T_{HPW}(x)HPW' = V_B$? and $2T_c(x)T_{HPW}(x)HPW' = V_C$?. If above equations hold, which means Bob and Carl are two legal users, or Alice will abort this process. After authenticating Bob and Carl, Alice uses the new password PW' instead of PW.

Remark: there is another scene about password changing phase: if the three nodes just finished the three-party PAKA phase, they can use the session key to change the new password directly. This method is simple, so we omit it the concrete the processes.

4. Security Analysis.

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

Theorem 4.1. *Let 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,

$$Adv_{\Pi,D}(A) \leq \frac{2q_h^2}{2^{l+1}} + \frac{3(q_s+q_e)^2}{2p^2} + q_e \cdot Adv_{x,p}^{cdh}(A) + \frac{1}{D} + \frac{q_s^2}{D}$$

where $Adv_{x,p}^{cdh}(A)$ is the success probability of A of solving the chaotic maps-based computational Diffie–Hellman problem (see definition 2.4). 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

$$Adv_{\Pi,D}(A) = 2|\Pr[Succ_0] - \frac{1}{2}| \tag{13}$$

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

$$\Pr[Succ_1] = \Pr[Succ_0] \tag{14}$$

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 $\{T_{a'}(x), V_A\}$ or $\{T_{b'}(x), V_B\}$ or $\{T_{c'}(x), V_C\}$ 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, a', b, b', c, c' were selected uniformly at random. Thus, we have

$$\Pr[Succ_2] - \Pr[Succ_1] = \frac{q_h^2}{2^{l+1}} + \frac{(q_h+q_e)^2}{2p^2} \tag{15}$$

Game G_3 In this game, the simulation of all the oracles is identical to game G_2 except that the game is terminated if the collision occurs in the simulation of the partial transcripts $\{E_A\}$ or $\{E_B\}$ or $\{E_C\}$. For any instance, we change the simulation of queries to the Send oracle for the selected session in game G_2. There are two possible cases where the adversary distinguishes the real partial transcripts (such as $\{E_A\}$) and the random messages as follows:

Case 1. We only consider the computation way of values E_A so that they become independent of passwords and ephemeral nonces. When Send query $\{E_A\}$ is asked, we set $E_A = T_a(x)T_{HPW}T_{a'}(x) = T_a(x)T_{a'}T_{HPW}(x) = T_a(x)T_A(x)$, and simplify it to $T_{HPW}T_{a'}(x) = T_{a'}T_{HPW}(x) = T_A(x)$, where A is selected from $[1, p+1]$ at random. The event occurs is lower than $q_e \cdot Adv_{x,p}^{cdh}(A)$.

Case 2. The adversary asks Send query except Send $\{ID_A, T_{a'}(x), E_A, V_A\}$ with $HPW = H(PW)$, and sucessfully impersonates A to B or A to C. The adversary is

allowed to reveal the static messages $\{ID_A, T_{a'}(x), E_A, V_A\}$ (or called it replay attack method), but it is not allowed to reveal the shared password PW. Thus, in order to impersonate A, the adversary has to obtain some information of the shared password of Alice. The probability is $1/D$.

In conclusion, the difference between the game G_3 and the game G_2 is as follows:

$$|\Pr[Succ_3] - \Pr[Succ_2]| \leq q_e \cdot Adv_{x,p}^{cdh}(A) + \frac{1}{D} \tag{16}$$

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 session key $SK = T_{H(T_a(x)T_b(x)T_c(x))}(x)$. A gets the session key SK by hash function query with probability at most $\frac{q_h^2}{2^{l+1}}$, and the probability of collisions in the transcripts simulations is at most $\frac{(q_h+q_e)^2}{p^2}$. Since a, b, c were selected uniformly at random. Hence, we have

$$|\Pr[Succ_4] - \Pr[Succ_3]| \leq \frac{q_h^2}{2^{l+1}} + \frac{(q_h + q_e)^2}{p^2} \tag{17}$$

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:

$$Adv_{\Pi,D}(A) \leq \frac{2q_h^2}{2^{l+1}} + \frac{3(q_s + q_e)^2}{2p^2} + q_e \cdot Adv_{x,p}^{cdh}(A) + \frac{1}{D} + \frac{q_s^2}{D}$$

4.2. **Further Security Discussion of the Proposed Scheme.**

Proposition 4.1. *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 $\{E_B, V_B\}$ or $\{E_C, V_C\}$.

- 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, the adversary 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 or E_C.
- 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. It has the same proof process for V_B or V_C.

- Combine $\{m_A, m_B, m_C\}$ to launch password guessing attack. Any combination of these messages $\{m_A, m_B, m_C\}$ 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 4.2. *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 4.3. *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 or CDL 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 4.4. *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 any other 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 may try to generate a valid authenticated message $m_A = \{ID_A, T_{a'}(x), E_A, V_A\}$ for two random values a, a'. 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 4.5. *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 4.6. *The proposed scheme could support mutual authentication.*

Proof: In our scheme, the user B verifies the authenticity of user A and C's requests by verifying the condition $4T_a(x)T_c(x)(T_{HPW}(x))^2 = V_A V_C$? during the proposed phase. To compute E_A, V_A, E_C, V_C, the shared password is needed. Therefore, an adversary cannot forge these messages. Additionally, E_A, V_A, E_C, V_C includes large random nubmers a, a' and c, c', 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 and C, and the user C authenticating the user A and B. Hence, mutual authentication can successfully achieve in our scheme.

Proposition 4.7. *The proposed scheme could have Key freshness property.*

Proof: Note that in our scheme, each established session key $SK = T_{H(T_a(x)T_b(x)T_c(x))}(x)$ includes random values a, b, c. The unique key construction for each session shows that proposed scheme supports the key freshness property.

Proposition 4.8. *The proposed scheme could have known key secrecy property.*

Proof: In our scheme, if a previously established session key $SK = T_{H(T_a(x)T_b(x)T_c(x))}(x)$ 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 three nonces, which ensures different key for each session.

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

Proposition 4.9. *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 4.10. *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.

5. **Efficiency Analysis.** Let T, E, D, H and X be the time for performing a Chebyshev polynomial computation, a modular exponentiation, a symmetric encryption/decryption, a one-way hash function and a XOR operation, respectively. The performance comparison of authentication and key agreement phase between our scheme and other two recently proposed related schemes in [5, 6] is given in **Table 2**.

TABLE 2. Comparisons between our proposed scheme and the related literatures

Protocol (Authentication phase)		**[5] (2013)**	**[6] (2015)**	**Ours**
Computation	User_A	3T + 5H + 2D	3T + 3H	8T + 2H
	User_B	3T + 5H + 2D	3T + 3H	8T + 2H
	Server or User_C	2T + 4H + 4D	4T + 6H	8T + 2H
	Total	8T + 14H + 8D	10T + 12H	24T + 6H
Communication	Messages	7	7	3
	rounds	4	4	1
	Number of nonces	2	4	6
	Model	Random Oracle	/	Random Oracle

In our novel scheme based on new theorem of chaotic maps, we abandon some time-consuming algorithm, such as modular exponentiation and scalar multiplication on elliptic curves. There are some works [1, 18, 20] about the computational time of a one-way hashing operation, a symmetric encryption/decryption operation, an elliptic curve point multiplication operation, Chebyshev polynomial operation and XOR operation. There are slightly differences on the basis of these literatures [1,18,20], but the basic conclusions are the same: $E > T > D \gg H \gg X$. Therefore, the computational cost of XOR operation could be ignored when compared with other operations, and we can compared the counts of each algorithm to analyze the efficiency. From the **Table 2**, the proposed scheme enjoys acceptable efficiency.

6. **Conclusion.** The study presented a novel Three-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. The security proof 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. We will further explore the new theorem of chaotic maps in N-party or in different application environments.

REFERENCES

[1] N. K. Pareek, V. Patidar, K. K. Sud, Discrete chaotic cryptography using external key. *Phys. Lett. A* vol. 309, pp. 75–82 (2003).

[2] M. S. Baptista, Cryptography with chaos, *Physics Letters A*, vol. 240, no. 1, pp. 50-54, 1998.

[3] Y. Liu and K. P. Xue, An improved secure and efficient password and chaos-based two-party key agreement protocol. *Nonlinear Dyn* (2016) vol. 84, pp.549–557.

[4] T. F. Lee, Enhancing the security of password authenticated key agreement protocols based on chaotic maps. *Inf. Sci.* vol.290, 63–71 (2015).

[5] Q. Xie, J. M. Zhao, X. Y. Yu, Chaotic maps-based three-party password-authenticated key agreement scheme. *Nonlinear Dyn* vol. 74, pp.1021–1027, 2013.

[6] Q. Xie, B. Hu, T. Wu, Improvement of a chaotic maps-based three-party password-authenticated key exchange protocol without using server's public key and smart card. *Nonlinear Dyn* , vol. 79, pp.2345–2358, 2015.

[7] 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. 1-10, 2016.

[8] F. Özkaynak, Cryptographically secure random number generator with chaotic additional input. *Nonlinear Dyn.* vol. 78, no. 3,pp. 2015–2020 , 2014.

[9] J. Chen, J. Zhou, K. W. Wong, A modified chaos-based joint compression and encryption scheme. *IEEE Trans. Circuits Syst.* II *Express Briefs* vol. 58, no. 2, pp. 110–114, 2011.

[10] P. Bergamo, P. D'Arco, A. De Santis, L. Kocarev, Security of public-key cryptosystems based on Chebyshev polynomials. *IEEE Trans. Circuits Syst.* I, vol. 52, no.7, pp. 1382–1393, 2005.

[11] S. J. Xu, , X. B.Chen, R. Zhang, Y. X. Yang, Y. C. Guo, An improved chaotic cryptosystem based on circular bit shift and XOR operations. *Phys. Lett.* A vol.376, no. 10, pp. 1003–1010, 2012.

[12] K. Chain, W. C. Kuo. A new digital signature scheme based on chaotic maps. *Nonlinear Dyn* , vol. 74, pp. 1003–1012, 2013.

[13] H. Wang, H. Zhang, J. Li and X. U. Chen, "A(3, 3) visual cryptography scheme for authentication," *Journal of Shenyang Normal University* (*Natural Science Edition*), vol. 31, no. 3, pp. 397-400, 2013.

[14] H. F. Zhu, A Provable Privacy-Protection System for Multi-server Environment, *Nonlinear Dyn* , vol.82, pp. 835–849, 2015.

[15] H. F. Zhu, Flexible and Password-Authenticated Key Agreement Scheme Based on Chaotic Maps for Multiple Servers to Server Architecture, *Wireless Pers Commun*, Volume vol.82, no. 3, pp. 1697-1718, 2015.

[16] Dolev, D., & A. C. Yao, On the security of public key protocols. *IEEE Transactions on Information Theory*, vol. 29, no.2, , pp. 198–208, 1983.

[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, 2014.

[18] X. Wang , and J. Zhao , An improved key agreement protocol based on chaos, *Commun. Nonlinear Sci. Numer. Simul*, vol. 15, pp. 4052–4057, 2010.

[19] L. Zhang, Cryptanalysis of the public key encryption based on multiple chaotic systems, *Chaos Solitons Fractals*, vol. 37, no. 3, pp. 669–674, 2008.

[20] L. Kocarev, and S. Lian, *Chaos-Based Cryptography: Theory, Algorithms and Applications*, pp. 53–54, 2011.

[21] H. F. Zhu, Y. F. Zhang, Y. Zhang and H. Y. Li, A Novel and Provable Authenticated Key Agreement Protocol with Privacy Protection Based on Chaotic Maps towards Mobile Network, *International Journal of Network Security*, vol. 18, no. 1, pp. 116-123, 2016.

[22] H. F. Zhu, Y. F. Zhang, X. Yu , and H. Y. Li, "Password-Authenticated Key Exchange Scheme Using Chaotic Maps towards a New Architecture in Standard Model, *International Journal of Network Security*, vol. 18, no. 2, pp. 326-334, 2016.

[23] H. F. Zhu, Y. F. Zhang, and Y. Zhang, A Provably Password Authenticated Key Exchange Scheme Based on Chaotic Maps in Different Realm, *International Journal of Network Security*, vol. 18, no. 4, pp. 688-698, 2016.