Provably Secure and Repeatable Authenticated Privacy-Protection Scheme Using Chaotic Maps with Distributed Architecture

Hongfeng Zhu, Junlin Liu

(Corresponding author: Hongfeng Zhu)

Software College, Shenyang Normal University
No.253, HuangHe Bei Street, HuangGu District, Shenyang, P.C 110034 China
(Email:zhuhongfeng1978@163.com; 272297257@qq.com)

(Received Nov. 5, 2016; revised May 17, 2017 and accepted July 13, 2017)

Abstract

Nowadays, the distributed password-authenticated key agreement schemes become more and more popular. Compare with the three traditional architectures (client/server, two clients/server and multi-server), the distributed architecture can solve problems of single-point of security, single-point of efficiency and single-point of failure. Moreover, it has the characteristics of scalability, flexibility and fairness. In the paper, we proposed a new Provably Secure and Distributed Privacy-Protection scheme using chaotic maps. The proposed scheme firstly achieves mutual authenticated among three nodes in three rounds with privacy protection, and at the same time, the unregistered server can store a temporary authenticator for a while for improving the efficiency. Security of the scheme is based on chaotic maps hard problems and a secure one way hash function. 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: Chaotic Maps Keywords; Distributed Architecture; Key Agreement; Privacy-protection

1 Introduction

Nowadays, more and more people want to enjoy surfing on Internet and meanwhile care about their privacy. The most popular technology is authenticated key exchange (AKE) [5, 12, 13] which can establish an authenticated and confidential communication channel. Many papers adopt multi-server architecture (MSA) [7, 14] to reduce the numbers of users' registration, and the literature [14] can achieve Privacy-Protection and without using symmetric cryptography which can lower the calculated amount. For seeking universal computing environment, Zhu [15] proposed an AKE protocol in different realm, which can make two-party in two-realm negotiate a session key in the standard model. Naturally, the group key agreement scheme with privacy preserving can be proposed in [10, 11, 16]. But the multi-server architecture makes the registration center become the focus of hacker. Furthermore, single-point of efficiency and single-point of failure trouble the registration center all the time.

An excellent architecture can make some hard problems become better easily. For example, distributed architecture can solve centralized architecture problems. Zhu [9] firstly proposed a new distributed architecture which called Multiple Servers to Server Architecture (MSTSA). The paper [9] proposed the first provably secure and flexible password-authenticated key agreement scheme based on chaotic maps [1, 3] with MSTSA in random oracle model [2]. Then, Zhu gives another password-authenticated key agreement scheme [13] with MSTSA which security is proved in standard model. But abovementioned two distributed schemes using chaotic maps have two main problems: without privacy protection and have many communicated rounds [4]. Therefore, the paper proposes a new distributed scheme to solve the two main problems. We adopt chaotic maps because it has numerous advantages, such as extremely sensitive to initial parameters, unpredictability, boundeness, etc. Meanwhile, chaotic sequence generated by chaotic system has the properties of non-periodicity and pseudo-randomness.

The main contributions are shown as below: (1) The paper presents a new password authenticated key exchange scheme with privacy protection towards Multiple Servers to Server Architecture. (2) The proposed scheme achieves mutual authenticated among three nodes in three rounds with privacy protection. (3) The scheme can make the unregistered server store a temporary authenticator for a while for avoiding the registered server involved over and over again. (4) The proposed scheme is mainly based

on chaotic maps without using modular exponentiation and scalar multiplication on an elliptic curve. In Security aspect, the protocol can resist all common attacks, such as impersonation attacks, man-in-the-middle attacks, etc. (5) About functionality, the protocol also has achieved some well-known properties, such as perfect forward secrecy and execution efficiency.

The rest of the paper is organized as follows: Some preliminaries are given in Section 2. Next, a distributed privacy-protection scheme is described in Section 3. Then, the security analysis and efficiency analysis are given in Section 4 and Section 5. This paper is finally concluded in Section 6.

2 Preliminaries

2.1 Multi-server Architecture

In the multi-server environment [7], each user must perform authentication procedure to login the server for a transaction. If the user is in a single authentication architecture, then the user must register at various servers and memorize the corresponding identifications and passwords, which could not be convenient for a user. In order to make the registration to various servers easier for users, each user must register with the registration center to obtain a secure account. Then the user uses the secure account to perform the login and authentication procedures with various servers.

2.2 Multiple Servers to Server Architecture

In the proposed multiple servers to server communication architecture, the registration center is not fixed. In other words, any server can work as a registration center. However in multi-server authentication architecture, the single registration center will face to single-point of security, single-point of efficiency and single-point of failure problems. The proposed architecture can solve the problems under multi-server environment with only one registration center architecture, that means "once security register for all registration" [9].

2.3 Chebyshev Chaotic Maps

Zhang [8] proved that semi-group property holds for Chebyshev polynomials defined on interval $(-\infty, +\infty)$. The enhanced Chebyshev polynomials are used in the proposed protocol:

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

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

$$T_{rs}(x) = T_r(T_s(x)) = T_s(T_r(x))$$

Definition 1. *(Enhanced Chebyshev polynomials) The enhanced Chebyshev maps of degree $n(n \in N)$ are defined as: $T_n(x) = (2xT_{n-1}(x) - T_{n-2}(x))(\mathrm{mod}p)$, where $n \geq 2$,$x \in (-\infty, +\infty)$, and p is a large prime number. Obviously, $T_{rs}(x) = T_r(T_s(x)) = T_s(T_r(x))$.*

Definition 2. *(DLP, Discrete Logarithm Problem) Given an integer a,find the integer r, such that $T_r(x) = a$.*

Definition 3. *(CDH, Computational Diffie-Hellman Problem) Given an integer x, and the values of $T_r(x), T_s(x)$, what is the value of $T_{rs}(x)$?*

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

2.4 Threat Model

The threat model should be adopted the widely accepted security assumptions about password based authentication schemes [2].

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. Only polynomial number of queries occurs between adversary and the participant 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: (1) Both Π_U^i and Π_S^i accept; (2) Both Π_U^i and Π_S^i share the same session identifications(sid); (3) 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: (1) It has accepted; (2) Both Π_U^i and its partner Π_S^i are unopened; (3) 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 Proposed Privacy Protection Scheme with Multiple Servers to Server Architecture

3.1 User Registration Phase

The concrete notations used hereafter are: ID_{S_i} means identity of the ith server; ID_A means the identity of Alice; a, a_1, r_a, r_i are all nonces; $(x, T_{k_i}(x))$, the public key based on Chebyshev chaotic maps of the ith server; k_i, the secret key based on Chebyshev chaotic maps of the ith server; H, A secure one-way hash function. H: $\{0,1\}^* \rightarrow \{0,1\}^l$ for a constant l; $\|$ means concatenation operation.

Figure 1 illustrates the user registration phase.

Step 1. When a user wants to be a new legal user, she chooses her identity ID_A, a random number r_a, and computes $H(r_a||PW)$. Then Alice submits $ID_A, H(r_a||PW)$ to the **RC** via a secure channel.

Step 2. Upon receiving $ID_A, H(r_a||PW)$ from Alice, the **RC** computes $B = H(ID_A||k_i) \oplus H(r_a||PW)$, where k_i is the secret key of S_i. Then Alice stores $\{ID_A, r_a, B\}$ in a secure way.

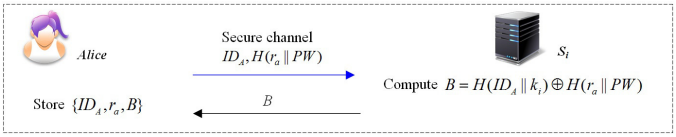

Figure 1: a premium user registration phase

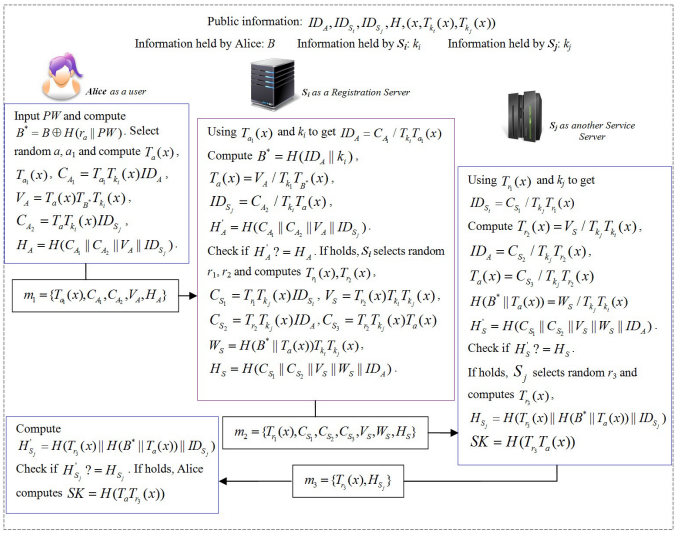

Figure 2: Authenticated key agreement phase for MSTSA with privacy protection

3.2 Authenticated Key Agreement Phase for MSTSA with Privacy Protection

Figure 2 illustrates the process of authenticated key agreement phase.

Step 1. If Alice wishes to consult some personal issues establish with S_j in a secure way, she will input password and compute $B_A^* = B_A \oplus H(r_a||PW)$, and then choose two random integer numbers a, a_1 and compute $T_a(x)$, $T_{a_1}(x)$, $C_{A_1} = T_{a_1} T_{k_i}(x) ID_A$, $V_A = T_a(x) T_{B^*} T_{k_i}(x)$, $C_{A_2} = T_a T_{k_i}(x) ID_{S_j}$, $H_A =$

$H(C_{A_1}||C_{A_2}||V_A||ID_{S_j})$. After that, Alice sends $m_1 = \{T_{a_1}(x), C_{A_1}, C_{A_2}, V_A, H_A\}$ to S_i which she has registered.

Step 2. After receiving the message $m_1 = \{T_{a_1}(x)$, C_{A_1}, C_{A_2}, V_A, $H_A\}$ from Alice, S_i will use $T_{a_1}(x)$ and the secret key k_i to get $ID_A = C_{A_1}/T_{k_i}T_{a_1}(x)$. Then S_i computes $B^* = H(ID_A \,||\, k_i)$, $T_a(x) = V_A/T_{k_1}T_{B^*}(x)$, $ID_{S_j} = C_{A_2}/T_{k_i}T_a(x)$, $H'_A = H(C_{A_1} \,||\, C_{A_2} \,||\, V_A \,||\, ID_{S_j})$. Check if H'_A is equal to H_A. If holds, that means Alice is the real and legal user. Next, S_i selects random r_1, r_2 and computes $T_{r_1}(x), T_{r_2}(x)$, $C_{S_1} = T_{r_1}T_{k_j}(x)ID_{S_i}$, $V_S = T_{r_2}(x)T_{k_i}T_{k_j}(x)$, $C_{S_2} = T_{r_2}T_{k_j}(x)ID_A$, $C_{S_3} = T_{r_2}T_{k_j}(x)T_a(x)$, $W_S = H(B^* \,||\, T_a(x))T_{k_i}T_{k_j}(x)$, $H_S = H(C_{S_1} \,||\, C_{S_2} \,||\, V_S \,||\, W_S \,||\, ID_A)$. After that, S_i sends $m_2 = \{T_{r_1}(x)$, C_{S_1}, C_{S_2}, C_{S_3}, V_S, W_S, $H_S\}$ to server S_j which Alice wants to get service.

Step 3. After receiving the message $m_2 = \{T_{r_1}(x)$, C_{S_1}, C_{S_2}, C_{S_3}, V_S, W_S, $H_S\}$ from S_i, S_j uses $T_{r_1}(x)$ and the secret key k_j to get $ID_{S_i} = C_{S_1}/T_{k_j}T_{r_1}(x)$. Then S_j computes $T_{r_2}(x) = V_S/T_{k_j}T_{k_i}(x)$, $ID_A = C_{S_2}/T_{k_j}T_{r_2}(x)$, $T_a(x) = C_{S_3}/T_{k_j}T_{r_2}(x)$, $H(B^* \,||\, T_a(x)) = W_S/T_{k_j}T_{k_i}(x)$, $H'_S = H(C_{S_1} \,||\, C_{S_2} \,||\, V_S \,||\, W_S \,||\, ID_A)$. Check if H'_S is equal to H_S. If holds, that means S_i is the real and legal server. Next, S_j selects random r_3 and computes $T_{r_3}(x)$, $H_{S_j} = H(T_{r_3}(x)||H(B^*||T_a(x))||ID_{S_j})$, $SK = H(T_{r_3}T_a(x))$. Finally, S_j sends $m_3 = \{T_{r_3}(x), H_{S_j}\}$ to Alice.

Step 4. After receiving the message $m_3 = \{T_{r_3}(x), H_{S_j}\}$, Alice computes $H'_{S_j} = H(T_{r_3}(x) \,||\, H(B^* \,||\, T_a(x)) \,||\, ID_{S_j})$ using local information. Then, Alice checks if H'_{S_j} is equal to H_{S_j}. If holds, that means S_i has helped Alice to authenticate S_j, because S_j owns the authenticator $H(B^*||T_a(x))$ which only Alice and S_i can compute B^*. Finally, Alice computes the session key $SK = H(T_aT_{r_3}(x))$.

If any authenticated process does not pass, the protocol will be terminated immediately.

Remark 1. *$H(B^*||T_a(x))$ is the temporary authenticator which can be used for a certain time. So, Alice and S_j can use $H(B^*||T_a(x))$ to construct some other session keys, such as $H(H(B^*||T_a(x)))$, $H(H(B^*||T_a(x))||T_{r_3}(x))$ and so on, without S_i involved.*

3.3 Password Changing Phase

Figure **??** illustrates the password changing phase.

Step 1. When a user wants to change her password, she chooses a new password, two random numbers r'_a, a, and computes $B^* = B \oplus H(r_a||PW)$, $T_a(x)$, $K_{A-S_i} = T_aT_k(x)$, $H_A = H(B^*||ID_{S_i}||T_a(x)||C_1||C_2)$, $C_1 = ID_A \times K_{A-S_i}$ and $C_2 = H(r'_a||PW') \times K_{A-S_i}$. Then Alice sends $m_1 = \{T_a(x), C_1, C_2, H_A\}$.

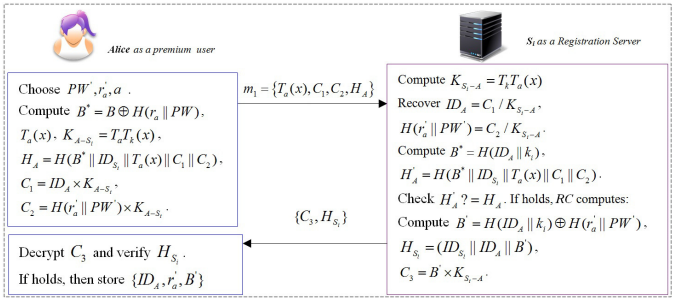

Figure 3: Password changing phase

Step 2. Upon receiving $m_1 = \{T_a(x)$, C_1, C_2, $H_A\}$ from Alice, S_i computes $K_{S_i-A} = T_kT_a(x)$ and recovers $ID_A = C_1/K_{S_i-A}$, $H(r'_a||PW') = C_2/K_{S_i-A}$. Next S_i computes $B^* = H(ID_A||k_i)$ and $H'_A = H(B^*||ID_{S_i}||T_a(x)||C_1||C_2)$. Then S_i checks $H'_A = H_A$ or not. If holds, S_i computes $B' = H(ID_A||k_i) \oplus H(r'_a||PW')$, $H_{S_i} = (ID_{S_i}||ID_A||B')$ and $C_3 = B' \times K_{S_i-A}$, where k_i is the secret key of S_i. Finally S_i sends $\{C_3, H_{S_i}\}$ to Alice.

Step 3. Upon receiving $\{C_3$, $H_{S_i}\}$, Alice uses K_{A-S_i} to decrypt C_3 to get B'. Then Alice computes locally $H'_{S_i} = (ID_{S_i}||ID_A||B')$ to compare with H_{S_i}.If they are equal, Alice stores $\{ID_A, r'_a, B'\}$ in a secure way.

4 Security Analysis

4.1 The Provable Security of the Proposed Scheme [2]

First of all, we transform the process of our proposed scheme with privacy protection in MSTSA to the following two simulation Algorithms.

Theorem 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,*

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

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.

Algorithm 1 Simulation of send query

On a query $send(\Pi_U^i, start)$, assume that U_i is in correct state, then we proceed as follows:
Choose two numbers $a, a_1 \in_R Z_p^*$, compute $\{T_{a_1}(x), C_A, V_A, H_A\}$. This query returns $\{T_{a_1}(x), C_A, V_A, H_A\}$ as answer.
On a query $send(S_i, \{T_{a_1}(x), C_A, V_A, H_A\})$, assume that Si is in correct state, we continue as follows:
Compute $ID_A = C_{A_1}/T_{k_i}T_{a_1}(x)$, $B^* = H(ID_A||k_i)$, $T_a(x) = V_A/T_{k_1}T_{B^*}(x)$, $ID_{S_j} = C_{A_2}/T_{k_i}T_a(x)$ and $H'_A = H(C_{A_1}||C_{A_2}||V_A||ID_{S_j})$.
if $H'_A \neq H_A$ **then**
  Reject the message.
  else S_i choose two numbers $r_1, r_2 \in_R Z_p^*$ and computes $\{T_{r_1}(x), C_{S_1}, C_{S_2}, V_S, W_S, H_A\}$. This query returns $\{T_{r_1}(x), C_{S_1}, C_{S_2}, V_S, W_S, H_A\}$ as answer.
  On a query $send(S_j, \{T_{r_1}(x), C_{S_1}, C_{S_2}, V_S, W_S, H_A\})$, assume that Sj is in correct state, we continue as follows:
  Compute $T_{r_2}(x) = V_S/T_{k_j}T_{k_i}(x)$, $ID_A = C_{S_2}/T_{k_j}T_{r_2}(x)$, $H(B^*||T_a(x)) = W_S/T_{k_j}T_{k_i}(x)$ and $H'_S = H(C_{S_1}||C_{S_2}||V_S||W_S||ID_A)$
  if $H'_S \neq H_S$ **then**
    Reject the message.
    else S_j chooses a number $r_3 \in_R Z_p^*$ and computes $H_{S_j} = H(T_{r_3}(x)||H(B^*||T_a(x))||ID_{S_j})$ and $SK = T_{r_3}T_a(x)$. The query $\{T_{r_3}(x), H_{S_j}\}$ returns as answer.
  end if
end if
On a query $send\{T_{r_3}(x), H_{S_j}\}$, assume that U_i is in correct state, then we proceed as follows:
U_i computes $H'_{S_j} = H(T_{r_3}(x)||H(B^*||T_a(x))||ID_{S_j})$.
if $H'_{S_j} \neq H_{S_j}$ **then**
  Reject the message.
  else compute $SK = T_aT_{r_3}(x)$ and the user U_i instance accepts.
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

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

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

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

$$\Pr[Succ_1] = \Pr[Succ_0] \quad (2)$$

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 $\{T_{a_1}(x), C_A, V_A, H_A\}$, $\{T_{r_1}(x), C_{S_1}, C_{S_2}, V_S, W_S, H_S\}$ or $\{T_{r_3}(x), H_{S_j}\}$ 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_1, r_i were selected uniformly at random. Thus, we have

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

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, r_3 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 $SK = H(T_aT_{r3}(x))$. Thus, $Adv_G^{cdh}(A) \geq \frac{1}{q_h}|\Pr[Succ_3] - \Pr[Succ_2]| - \frac{1}{p}$,that is, the difference between the game G_3 and the game G_2 is as follows:

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

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 $SK = H(T_aT_{r3}(x))$. A gets the session key SK by hash function query with probability at most $\frac{q_h^2}{2^{l+1}}$. Hence, we have

$$|\Pr[Succ_4] - \Pr[Succ_3]| \leq \frac{q_h^2}{2^{l+1}} \quad (5)$$

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 on-line password guessing attack is at most $\frac{q_s}{D}$. Furthermore, the probability of A made off-line password guessing attack is 0, because even if A gets the secret information $\{ID_A, r_a, B\}$, he has no any compared value to authenticate the guessing password is right or not. Combining the Equations (1) - (5) one gets the announced result as:

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

□

4.2 Further Security Discussion

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

Proof. In this attack, an adversary may try to guess a legal user password PW 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. In our protocol, the adversary only can launch the on-line password guessing attack, because there are no any of the transmitted messages including password as the input value. Even if the adversary gets the secret information $\{ID_A, r_a, B\}$, he has no any compared value to authenticate the guessing password is right or not without the server help. In other words, the adversary cannot construct the form $function(*||PW') = y$, where * is any known message, and only the server can compute the value y. On the other side, about on-line password guessing attack, because the maximum number of allowed invalid attempts about guessing password is only a few times, then the account will be locked by the registration server. □

Proposition 2. *The proposed scheme could support mutual authentication.*

Proof. In our scheme, the Registration Server S_i verifies the authenticity of user A's request by verifying the condition $H'_A = H_A$? during the proposed phase. To compute $B^* = H(ID_A||k_i)$, the password is needed. Therefore, an adversary cannot forge the message. Additionally, C_{A_1}, C_{A_2}, V_A includes large random numbers a and a1, the adversary cannot replay the old message. This shows that S_i can correctly verify the message source.

For Alice authenticating the server S_j, it can be divided three steps: Firstly, S_i transfers the authenticator $H(B^*||T_a(x))$ which can only be decrypted by S_j using his own secret key k_j. Secondly, only S_i or S_j can compute $T_{k_i}T_{k_j}(x)$, so S_j authenticates S_i by verifying the condition $H'_S = H_S$? Finally, Alice authenticates the server S_j by verifying the condition $H'_{S_j} = H_{S_j}$? S_j computes H_{S_j} only by the helping of S_i, and while S_i and S_j have achieved mutual authentication.

Hence, mutual authentication can successfully achieve in our scheme. □

Proposition 3. *The proposed scheme could support Privacy-Protection.*

Proof. Alice's identity is anonymity for outsiders because ID_A is covered by $C_{A_1} = T_{a_1}T_{k_i}(x)ID_A$, and then only the Registration Server S_i can use his secret key to recover the ID_A. It is the same way for covered the identity of S_j. Due to PKC-based about our scheme, the ID_A and ID_{S_j} must be emerged to S_i, or it cannot construct the authenticator of the user and send the covered authenticator to S_j.

For the second message $m_2 = \{T_{r_1}(x), C_{S_1}, C_{S_2}, C_{S_3}, V_S, W_S, H_S\}$, we construct $C_{S_2} = T_{r_2}T_{k_j}(x)ID_A$ to covered Alice's identity, and $C_{S_1} = T_{r_1}T_{k_j}(x)ID_{S_i}$, $C_{S_3} = T_{r_2}T_{k_j}(x)T_a(x)$ to covered S_i's identity and $T_a(x)$. The encrypted message $C_{S_1}, C_{S_2}, C_{S_3}$ are generated from r_1, r_2 which are different in each session and are only known by S_i. S_j can decrypt $C_{S_1}, C_{S_2}, C_{S_3}$ using $T_{r_1}(x)$ and his own secret key which is secure under the CMBDLP and CMBDHP assumptions, and furthermore getting all the information $ID_{S_i} = C_{S_1}/T_{k_j}T_{r_1}(x)$, $ID_A = C_{S_2}/T_{k_j}T_{r_2}(x)$ and $T_a(x) = C_{S_3}/T_{k_j}T_{r_2}(x)$. Additionally, since the values r_1, r_2 of the random elements are very large, attackers cannot directly guess the value r_1, r_2 of the random elements to generate $T_{r_1}(x), T_{r_2}(x)$.

For S_j, because it has know all the necessary information including ID_{S_i}, ID_A, $T_a(x)$ and the covered authenticator $H(B^*||T_a(x))$, S_j only need send the authentication of integrity message $m_3 = \{T_{r_3}(x), H_{S_j}\}$ which has no any of information about identities of the involved three nodes.

Therefore, the proposed scheme provides privacy protection. □

Proposition 4. *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, or the secret keys which are covered strictly, so the proposed scheme withstands the stolen verifier attack. □

Proposition 5. *The proposed scheme could withstand replay and man-in-the-middle attacks.*

Proof. The verification messages include the temporary random numbers a, a_1, r_a, r_i. 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 secret keys or password). So our proposed scheme resists the replay and man-in-the-middle attacks. □

Table 1: Security of our proposed protocol

Category		Eun-Jun Yoon's Scheme [14] (2013)	Zhu's Scheme [5] (2015)	Zhu's Scheme [1] (2016)	Our scheme
Architecture		Multi-server (Centralized)	MSTSA (Distributed)	MSTSA (Distributed)	MSTSA (Distributed)
Architecture properties and functionality	Single-point of security	N/A	Provided	Provided	Provided
	Single-point of efficiency	N/A	Provided	Provided	Provided
	Single-point of failure	N/A	Provided	Provided	Provided
	Symmetry	N/A	Provided	Provided	Provided
	Transparency	**	***	***	***
	Simplicity	*	***	***	***
	Expandability	**	***	***	***
	No timestamp	Provided	Provided	Provided	Provided
	Secure password update	Provided	Provided	Provided	Provided
	Repeatable Authenticated	N/A	N/A	N/A	Provided
Security requirements	Privacy-Protection	N/A	N/A	N/A	Provided
	Mutual authentication	Provided	Provided	Provided	Provided
	Guessing attacks	Provided	Provided	Provided	Provided
	Man-in-the-middle attack	Provided	Provided	Provided	Provided
	Replay attack	Provided	Provided	Provided	Provided
	Key freshness property	Provided	Provided	Provided	Provided
	Perfect forward secrecy	Provided	Provided	Provided	Provided
	Data integrity	Provided	Provided	Provided	Provided
	Impersonation attack	Provided	Provided	Provided	Provided
	Known key secrecy property	Provided	Provided	Provided	Provided
	Stolen verifier attack	Provided	Provided	Provided	Provided
Security Model		Heuristic method	Random Oracle	Standard model	Random Oracle
Required components		**Hardware, software, biometric and password**	**Software and password**	**Software and password**	**Software and password**

N/A: not available or not support. *: provided but in low level.
: provided but in middle level. *: provided but in high level.

Proposition 6. *The proposed scheme could resist user impersonation attack.*

Proof. In such an attack, an adversary may try to masquerade as a legitimate user Alice 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 may try to generate a valid authenticated message $\{T_{a_1}(x), C_{A_1}, C_{A_2}, V_A, H_A\}$ for two random values a, a_1. However, the adversary cannot compute $\{C_{A_1}, C_{A_2}, V_A\}$ as computation of $\{C_{A_1}, C_{A_2}, V_A\}$ requires PW which is only known to legal users.

This shows that the proposed scheme resist user impersonation attack. □

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

Proof. Note that in our scheme, each established session key $SK = H(T_a T_{r_3}(x))$ includes random values a and r_3. 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 $SK = H(T_a T_{r_3}(x))$ 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_i, k_j because the session key is $SK = H(T_a T_{r_3}(x))$. This shows that our scheme provides the perfect forward secrecy property. □

From the Table 1, we can see that the proposed scheme is more secure and has much functionality compared with the recent related scheme.

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. In our proposed protocol, no time-consuming modular exponentiation and scalar multiplication on elliptic curves are needed. However, Wang [6] proposed several methods to solve the Chebyshev polynomial computation problem. For convenience, some notations are defined as follows. The computational cost of XOR operation could be ignored when compared with other operations. Table 2 shows performance comparisons between our proposed scheme and the literature of [7] in multi-server architecture and [13, 9] in MSTSA. Therefore, as in Table 2 the concrete comparison data as follows:

Table 2: Efficiency of our proposed scheme

Phase		Eun-Jun Yoon's Scheme [7] (2013)	Zhu's Scheme [9] (2015)	Zhu's Scheme [13] (2016)	Our scheme
A	Total	$1T_{hash}$	Not discussed	$1T_{hash}$	$2T_{hash}$
B	Total	$1T_{hash}$	No need	No need	No need
C	User	$5T_{hash}$ + $2T_{Ecc}$	$4T_{hash}$ +$2T_{sym}$+ $2T_{CH}$	$2T_F$ +$2T_{CH}$	$4T_{hash}$ + $4T_{CH}$
	S_i as a ***RC***/***RC***	$7T_{hash}$	$4T_{hash}$ +$4T_{sym}$+ $2T_{CH}$	$2T_F$ +$4T_{CH}$	$4T_{hash}$ + $5T_{CH}$
	S_j	$5T_{hash}$ +$2T_{Ecc}$	$3T_{hash}$ +$2T_{sym}$+ $2T_{CH}$	$2T_F$ +$2T_{CH}$	$4T_{hash}$ + $5T_{CH}$
	Total	$9T_{hash}$ + $3T_{Exp}$	$11T_{hash}$+ $8T_{sym}$+ $6T_{CH}$	$6T_F$ +$8T_{CH}$	$12T_{hash}$ + $14T_{CH}$
D	Total	$2T_{hash}$	$6T_{hash}$ +$4T_{sym}$+ $2T_{CH}$	$1T_{hash}$ +$2T_F$ +$2T_{CH}$	$7T_{hash}$ + $2T_{CH}$
E	Total	No need	$6T_{hash}$+ $4T_{sym}$+ $2T_{CH}$	No need	No need
F		5	5	4	3

A: User registration **B**: Server registration

C: authentication phase **D**: Password change phase

E: Shared key update among servers phase

F: Rounds of Authentication phase

T_{hash}: The time for executing the hash function;

T_F: The time for executing the pseudo-random function;

- T_{sym}: The time for executing the symmetric key cryptography;

T_{XOR}: The time for executing the XOR operation;

- T_{Exp}: The time for a modular exponentiation computation;
- T_{Ecc}: The time for executing the ECC multiplications

(ECC: Elliptic curve cryptosystem)

- T_{CH}: The time for executing the $T_n(x)$ mod p in Chebyshev polynomial.

6 Conclusion

We only use chaotic maps and a secure one-way hash function to construct a distributed password authenticated key scheme which provides a provable privacy protection towards Multiple Servers to Server Architecture. Our proposed scheme only needs three rounds can catch mutual authenticated with privacy protection among three parties in MSTSA, and the unregistered server can store a temporary authenticator for a certain time without the registered server involved. The above-mentioned innovation points can improve the efficiency of protocol immensely. Based on our discussion we proposed a suitable protocol that covers those goals and offered an efficient protocol that formally meets the proposed security definition. Finally, after comparing with related literatures respectively, we found our proposed scheme has satisfactory security, efficiency and functionality. Therefore, our protocol is more suitable for practical applications.

7 Acknowledgement

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

References

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

[2] 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.

[3] T. F. Lee, "Enhancing the security of password authenticated key agreement protocols based on chaotic maps," *Information Sciences*, vol. 290, pp. 63-71, 2015.

[4] Y. Sun, H. Zhu, and X. Feng, "A novel and concise multi-receiver protocol based on chaotic maps

with privacy protection," *International Journal of Network Security*, vol. 19, No. 3, pp. 371-382, 2017.

[5] H. J. Wang, H. Zhang, J. X. Li and 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, 2013.

[6] X. Wang, J. Zhao, "An improved key agreement protocol based on chaos," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, pp. 4052-4057, 2010.

[7] E. J. Yoon, K. Y. Yoo, "Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem," *Journal of Supercomputing*, vol. 63, pp. 235-255, 2013.

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

[9] H. F. 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, 2015.

[10] H. F. Zhu, "Secure chaotic maps-based group key agreement scheme with privacy preserving," *International Journal of Network Security*, vol. 18, no. 6, pp. 1001-1009, 2016.

[11] H. F. Zhu, R. Wang, "A survey to design privacy preserving protocol using chaos cryptography," *International Journal of Network Security*, vol. 20, no. 2, pp. 313-322, 2018.

[12] H. Zhu, Y. Zhang, "An improved two-party password-authenticated key agreement protocol with privacy protection based on chaotic maps," *International Journal of Network Security*, vol. 19, No. 4, pp. 487-497, 2017.

[13] H. F. Zhu, Y. F. Zhang, Y. Xia 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.

[14] H. F. Zhu, Y. F. Zhang, and Y. Sun, "Provably secure multi-server privacy-protection system based on Chebyshev chaotic maps without using symmetric cryptography," *International Journal of Network Security*, vol. 18, no. 5, pp. 803-815, 2016.

[15] 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.

[16] H. Zhu, Y. Zhang, Y. Zhang and H. 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.

Biography

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 50 international journal papers (SCI or EI journals) on the above research fields.

Junlin Liu graduated with a Bachelor of Engineering from Shenyang Normal University in 2017. In her college, after completing the learning task, she interests in exploring her professional knowledge. During graduate, under the guidance of his master instructor, she researches information security theory and technology.