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

Multi-party Password-Authenticated Key Exchange Scheme with Privacy Preserving using Chaotic Maps in Random Oracle Model

Hongfeng Zhu and Rui Wang

Software College, Shenyang Normal University
No.253, HuangHe Bei Street, HuangGu District, Shenyang, P.C 110034 - China
zhuhongfeng1978@163.com; 670322496@qq.com

Received April, 2016; revised August, 2016

Abstract. *Since the 1990s, chaotic systems have widely used to cryptography which can be used to design kinds of secure protocols, digital signatures, hash functions and so on, so there is an intuitive connection between group key agreement and chaotic maps. And the features of multi-party communication, such as fast, efficiency and security which would be suitable for using chaotic maps to construct. In this paper, we propose a novel chaotic maps-based multi-party password authentication key agreement protocol. In contrast to the recent literatures, our proposed scheme not only cares about security and efficiency of multi-party PAKE, but also pays more attention to multi-function, such as privacy protection and members dynamic change. Finally, we give the security proof and the efficiency analysis of our proposed scheme.*
Keywords: Key agreement, Multi-party, Privacy protection, Chaotic maps

1. **Introduction.** Mutual authentication key exchange (MAKE) 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 DeffieHellman (DH) 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, MAKE protocols are only related with the involving participants, so naturally we associated with the most general form, multi-party password authentication key exchange (M-PAKE).

Obviously, many computing surroundings need M-PAKE architecture, such as internet conference, multi-user dimension and many more multi-party applications. Ever since the first two-party PAKE protocol was proposed [2], the PAKE protocols were developing into three-party PAKE [3-5] and M-PAKE [6]. Many other PAKE protocols are similar as [2-6], so they can be classified as three types of evolutionary way: efficiency evolutionary, security evolutionary and functionality evolutionary. For example, the literature [6] pointed out that the literature [4] does not provide key authentication, key confirmation

and has four rounds (efficiency), that are satisfied both efficiency evolutionary and security evolutionary. Another example, the literature [6] provides user anonymity which is the functionality evolutionary.

Besides, compared with other cryptosystem systems, chaotic system has numerous advantages, such as extremely sensitive to initial parameters, unpredictability, deterministic random-like process and so on. In the past few years, cryptography systems based on chaos theory have been studied widely [9-12], the representative directions such as N-party($N \geq 2$) AKA protocols [9-12], random number generating [9], hash functions [10], digtal signature [11], anonymity [12], multi-server Environment [12].

However, as a kind of different password with centralized model (a server involved) M-PAKE, the latest literature [6] still has some flaws and lack of some important functionalities. In this paper, we demonstrate that Lus protocol [6] has still security problems: stolen-verifier attacks and leaking the timestamps. Based on chaotic maps, we provide a secure and efficient M-PAKE protocol with the important functionalities. The main contributions are shown as below: (1) Security. By analyzing of Lu et al.s scheme, we demonstrate that Lus protocol [6] has still security problems: stolen-verifier attacks and leaking the timestamps. The former directly leads to two kinds of serious consequences: one side is leaking verifier table in the server will cause an attack launch the others attack, such as modify verifier table even delete it, or s/he disguise the server to cheat the legal users and so on. The other side is about efficiency: the server maintains verifier table will waste amount of computing cycles and storage space. The latter may result in revealing some commercial transactions time. Besides, we also prove the security of our scheme in random oracle model. (2) Functionality. Most M-PAKE [6] lacks of the measures about changing of members. Our proposed protocol not only provides the method of members join or revocation, but also satisfies privacy protection including timestamp with unlinkability. One point should be noted that privacy protection can also be achieved during members join or revocation, and we called that dynamic privacy. Furthermore, our proposed protocol provides password changing phase.

The rest of the paper is organized as follows: fundamental knowledge of chaotic maps is given in Section 2. Next, a proposed privacy-protection M-PAKE protocol 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. **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 [8] $T_n(x) : [-1,1] \rightarrow [-1,1]$ is defined as $T_n(x) = cos(ncos^{-1}(x))$.Chebyshev polynomial map $T_n : R \rightarrow R$ of degree n is defined using the following recurrent relation: $T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)$,where $n \geq 2$, $T_0(x) = 1$, and $T_1(x) = x$.

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

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

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

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

Because it is actually proven insecure in literature [14] that Chebyshev polynomials are running the polynomial on decimal number, we adopts the enhanced Chebyshev polynomials to design our frameworks. In order to enhance the security, Zhang [15] proved that semi-group property holds for Chebyshev polynomials defined on interval $(-\infty, +\infty)$.

Definition 2.1. *(Enhanced Chebyshev polynomials) The enhanced Chebyshev maps of degree* $n(n \in N)$ *are defined as:* $T_n(x) = (2xT_{n-1}(x) - T_{n-2}(x))(mod\ p)$*, 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 2.2. *(DLP, Discrete Logarithm Problem) Given an integer* a*, find the integer* r*, such that* $T_r(x) = a$*.*

Definition 2.3. *(CDH, Computational DiffieHellman 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.

2.2. **Threat Model.** The threat model should be adopted the widely accepted security assumptions about password based authentication schemes [7].

(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 participants 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 users instances Π_U^i and servers 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^icontinues 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 users 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 M-PAKE protocol with dynamic privacy.** A M-PAKE scheme consists of three phases: user registration phase, the M-PAKE with privacy protection phase, password changing phase. And we also give the Members join or revocation methods. Some notations used hereafter are: ID_i, ID_S: The identities of the useri and the server, respectively; PW_i: The password of the $user_i$; R, r_i, s, a: Random numbers; $(x, T_k(x))$: Public key based on Chebyshev chaotic maps for the server;k: Secret key based on Chebyshev chaotic maps for the server;H:A secure one-way hash function;||: Concatenation operation;T: Timestamp.

3.1. **User registration phase. Fig.1** illustrates the user registration phase.

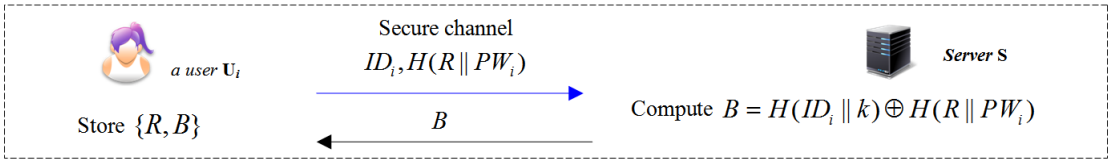

FIGURE 1. User registration phase

(1)$User_i \rightarrow Server$S : $\{ID_i, H(R||PW_i)\}$

When a user wants to be a new legal user, s/he chooses her/his identityID_i, a random number$R$, and computes$H(R||PW_i)$. Then Useri submits $ID_i, H(R||PW_i)$ to the S via a secure channel.

(2)$Server$S $\rightarrow User_i : \{B\}$

Upon receiving $ID_i, H(R||PW_i)$from $User_i$, the S computes $B = H(ID_i||k) \oplus H(R||PW_i)$, where k is the secret key of the server S. Then Useri stores $\{R, B\}$in a secure way.

3.2. **The M-PAKE with privacy protection phase.** This concrete process is presented in the following **Fig.2**

(1)$User_i \rightarrow Server$S : $\{T_{r_i}(x), C_{i_1}, C_{i_2}\}$

If N-party wishes to consult a group session key by the servers help in a anonymous way, each party Useri will input his own password and compute $B^* = B \oplus H(R||PW_i)$, and then choose a random integer number r_ia timestamp Ti and compute$T_{r_i}(x)$,$C_{i_1} = T_{r_i}T_k(x)(ID_i||T_i)$,$C_{i_2} = H(B^*||C_{i_1})$. After that, Useri sends $\{T_{r_i}(x), C_{i_1}, C_{i_2}\}$to S.

(2)$Server$S $\rightarrow User_i : \{C_{i_3}, C_{i_4}\}$

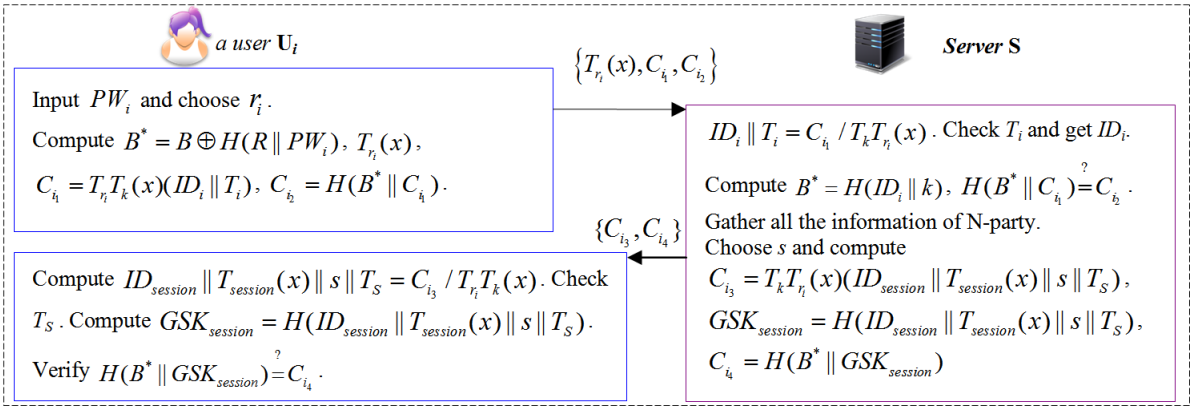

FIGURE 2. The proposed M-PAKE with privacy protection

After receiving the message $\{T_{r_i}(x), C_{i_1}, C_{i_2}\}$ from all the N-party, S firstly must confirm the identities of these messages and check the timestamps. So based on the private key k, S computes $C_{i_1}/T_k T_{r_i}(x) = ID_i||T_i$ to get these source of this messages and timestamps. If all the T_i are passed validation, S will compute $B^* = H(ID_i||k)$ and verifies $H(B^*||C_{i_1}) \stackrel{?}{=} C_{i_2}$. If above equations hold that means all the users are legal, or S will abort this process. After authenticating Useri, S chooses a random s, timestamp T_s and computes $C_{i_3} = T_k T_{r_i}(x)(ID_{session}||T_{session}(x)||s||T_S, GSK_{session} = H(ID_{session}||T_{session}(x)||s||T_S), C_{i_4} = H(B^*||GSK_{session})$ Finally S sends $\{C_{i_3}, C_{i_4}\}$ to Useri, where $ID_{session} = ID_S||ID_1||ID_2||...||ID_n$ and $T_{session}(x) = T_{r_1}(x)||T_{r_2}(x)||...||T_{r_n}(x)$.

$(3) User_i$

Because $T_{r_i}T_k(x)$ has already computed before, $User_i$ can get $ID_{session}||T_{session}(x)||s||T_S = C_{i_3}/T_{r_i}T_k(x)$ directly. Next, $User_i$ checks the timestamp T_s . If the timestamp passed the verification, Useri computes $GSK_{session} = H(ID_{session}||T_{session}(x)||s||T_S)$ and verifies $H(B^*||GSK_{session}) \stackrel{?}{=} C_{i_4}$. If above equation holds, that means S is a legal server, or $User_i$ will abort this process.

3.3. **Password changing phase.** Fig.3 illustrates the password changing phase.

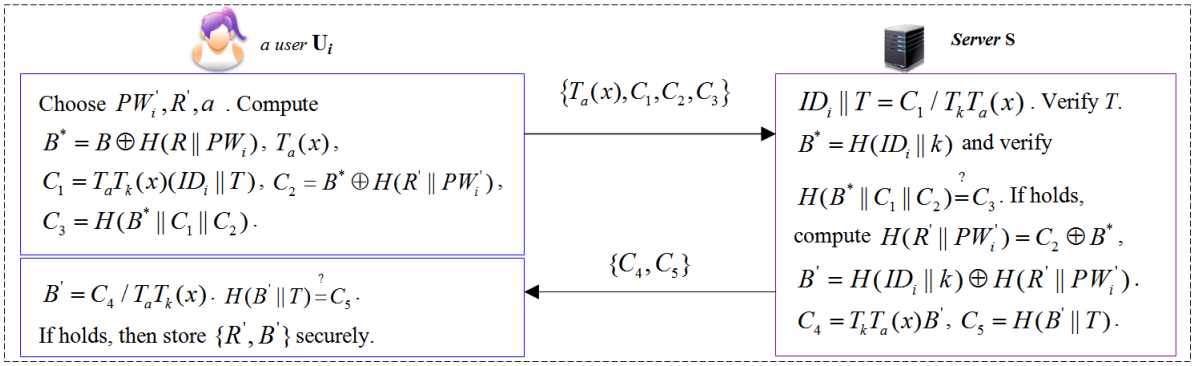

FIGURE 3. Password changing phase

$(1) User_i \rightarrow Server S : \{T_a(x), C_1, C_2, C_3\}$

When $User_i$ wants to change her password, she chooses $PW_A^{'}$, two random numbers $R^{'}, a$, a timestamp T, and computes $B^* = B \oplus H(R||PW_A), T_a(x), C_1 = T_a T_k(x)(ID_A||T), C_2 = B^* \oplus H(R^{'}||PW_A^{'}), C_3 = H(B^*||C_1||C_2)$. Then $User_i$ sends $\{T_a(x), C_1, C_2, C_3\}$ to the S.

(2)$ServerS \rightarrow User_i : \{C_4, C_5\}$

Upon receiving $\{T_a(x), C_1, C_2, C_3\}$ from $User_i$,firstly must confirm the identity of this message and verify timestamp. So based on the private key k, S computes $C_1/T_kT_a(x) = ID_A||T$ to get the source of this message and timestamp. If T is passed validation, S computes $B^* = H(ID_A||k)$ and verifies $H(B^*||C_1||C_2) \stackrel{?}{=} C_3$. If above equation holds, that means $User_i$ is a legal user, or S will abort this process. After authenticating $User_i$, S computes $H(R^{'}||PW_A^{'}) = C_2 \oplus B^*$,$B^{'} = H(ID_A||k) \oplus H(R^{'}||PW_A^{'})$,$C_4 = T_kT_a(x)B^{'}$,$C_5 = H(B^{'}||T)$ and sends $\{C_4, C_5\}$ to $User_i$.

(3) After receiving the message $\{C_4, C_5\}$, $User_i$ computes stores $B^{'} = C_4/T_aT_k(x)$ and verifies $H(B^{'}||T) \stackrel{?}{=} C_54$. If above equation holds,$User_i$ will store $\{R, B\}$ in a secure way.

3.4. **Members join or revocation. (1) Members join/Group merge**

We assume that some users $U_{n+1}...U_{n+m}$ want to join the group disscussion. For generalized, we view a user U_j is one of the new members.

(a) Efficient method without privacy protection. This concrete process is presented in the following **Fig.4**. In the stage1 and stage2, two groups will negotiate two group session key GSK_1 and GSK_2 using the protocol in the section 3.3. Then, the server S will choose a new timestamp T_{S-new} and compute $GK = GSK_1 \oplus GSK_2$, $ID_{session} = ID_S||ID_1||...||ID_{n+m}$,$GSK_{12} = H(GSK_1||GSK_2||T_{S-new})$,
$\delta = H(ID_{session}||GSK_1||GSK_2||T_{S-new})$.
Next, S broadcasts the messages $\{ID_{session}, GK, \delta, T_{S-new}\}$ to all the members including the new group.After reveiving the message for each member, any user can get the new session $GSK_{12} = H(GSK_1||GSK_2||T_{S-new})$ by using the old session key GSK1 or GSK2. Finally, each member verifies $H(ID_{session}||GSK_1||GSK_2||T_{S-new}) \stackrel{?}{=} \delta$ to get the key confirmation.**Remark:** In order to improve efficiency, we sacrifice the privacy protection in the **Fig.4** stage3. Because if the all the two group members identites are transmited in plaintext, we can save at least $2(n+m)T_c$(T_c means the time for executing the $T_n(x)$ mod p in Chebyshev polynomial).

(b) Privacy protection method. The only difference with (a) is the stage3 in **Fig.4**.For capturing privacy protection, we must sacrifice efficiency.The server S must choose a new timestamp T_{S-new} and compute
$C_{i-join} = T_kT_{r_i}(x)(ID_{session}||GSK_2||T_{S-new})(1 \le i \le n)$,
$C_{j-join} = T_kT_{r_j}(x)(ID_{session}||GSK_1||T_{S-new})(n+1 \le j \le n+m)$,
$\delta = H(ID_{session}||GSK_1||GSK_2||T_{S-new})$.Then S sends $\{C_{i-join}, \delta\}$ to each party in Group1 and sends $\{C_{i-join}, \delta\}$ to each party in Group2. Each party in both Group1 and Group2 can all use his secret random nonce to decrypt C for getting information $(ID_{session}||GSK_2||T_{S-new})$ or $(ID_{session}||GSK_1||T_{S-new})$.Finally,each party can compute $GSK_{12} = H(GSK_1||GSK_2||T_{S-new})$ and verify $H(ID_{session}||GSK_1||GSK_2||T_{S-new}) \stackrel{?}{=} \delta$.

(2) Members revocation. Without loss of generality, we assume that the subgroup $U_{n-j+1}, ..., U_n$ wants to leave.

(a) Efficient method without privacy protection. This concrete process is presented in the following **Fig.5** . In the stage1, Group1 has negotiated the group session key GSK_1 using the protocol in section 3.3. Then in the stage2, some parties want to leave Group1, so then will sends the identities of subgroup revocation to server S. Next, in stage3 S Chooses a new s, a new timestamp T_{S-new} and computes
$ID_{s-new} = ID_S||ID_1||...||ID_{n-j}$,$T_{s-new}(x) = T_{r_1}(x)||..||T_{r_{n-j}}(x)$,
$C_{i_1} = T_kT_{r_i}(x)(ID_{s-new}||T_{s-new}(x)||s||T_{S-new})$,
$GSK_{s-new} = H(ID_{s-new}||T_{s-new}(x)||s||T_{S-new})$,$C_{i_2} = H(B^*||GSK_{s-new})$. For each remain member i can compute $ID_{s-new}||T_{s-new}(x)||s||T_{S-new} = C_{i_1}/T_kT_{r_i}(x)$ and

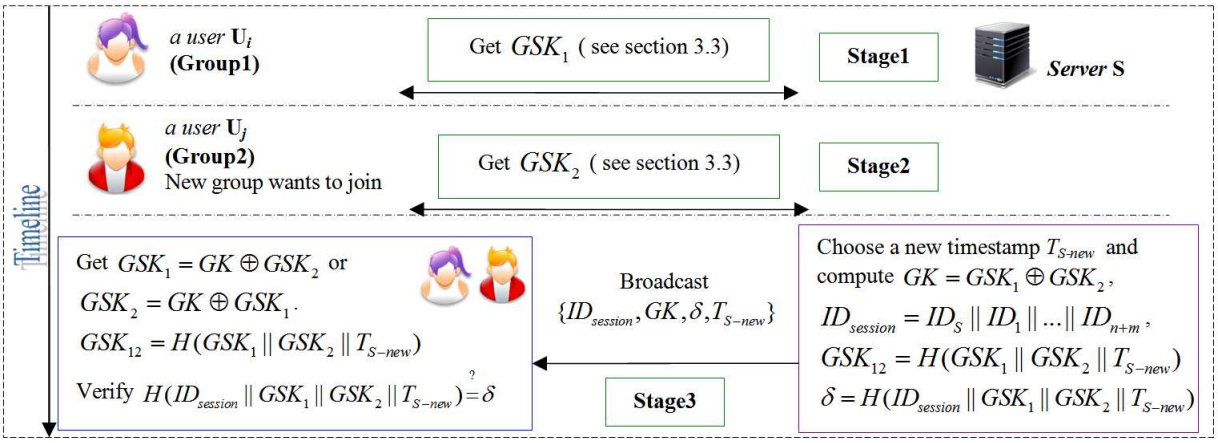

FIGURE 4. Members join

$GSK_{s-new} = H(ID_{s-new}||T_{s-new}(x)||s||T_{S-new})$.

Finally, each remain party verifies $H(B^*||GSK_{s-new}) \stackrel{?}{=} C_{i_2}$ and judges if the new GSK_{s-new} is valid or not. Remark: In order to improve efficiency, we sacrifice the privacy protection in the Fig.5 stage2. Because if the all the subgroup members' identites are transmited in plaintext, we can save at least $2jT_c$.

(b) Privacy protection method. The only difference with (a) is the stage2 in Fig.5. For capturing privacy protection, we must sacrifice efficiency. Some parties want to leave Group1, so each of them will sends the $\{C_{j_1}, C_{j_2}\}$ to server S, where $C_{j_1} = T_{r_j}T_k(x)(ID_j||T_j)$ and $C_{j_2} = H(B^*||C_{j_1})$. The server S will decrypt each message to get the $ID_j||T_j$ by his own secret key. Then, S check these messages are legal or not by C_{j_2}. Finally, S continues to execute the stage3.

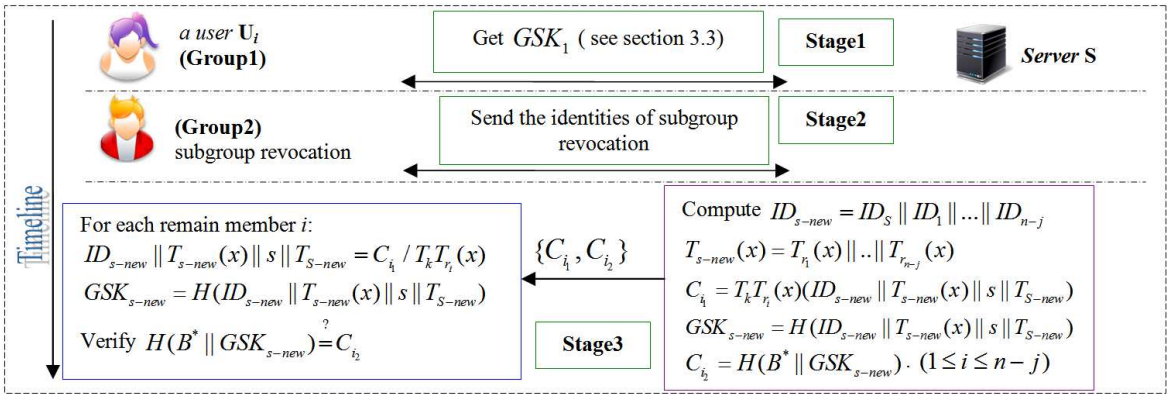

FIGURE 5. Members revocation

4. Security Analysis.

4.1. Formal Security Analysis of the Proposed Scheme [7].

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,

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

where $Adv_G^{cdh}(A)$ is the success probability of A of solving the chaotic maps-based computational DiffieHellman 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 $G_i(0 \le i \le 5)$, 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_i and the server S_j 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

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

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

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

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 transcripts $< T_{r_i}(x), C_{i_1}, C_{i_2} >$ and $< C_{i_3}, C_{i_4} >$. According to the birthday paradox, the probability of collisions of the simulation of hash oracles is at most $q_h^2/2p$. Similarly, the probability of collisions in the transcripts simulations is at most $(q_h+q_e)^2/2p$. Since r_i and s was selected uniformly at random. Thus, we have

$$\Pr[Succ_2] - \Pr[Succ_1] = q_h^2/2p + (q_h+q_e)^2/2p \,(3)$$

Game G_3 The simulation of this game is similar to the previous game except the game will be aborted if A can correctly guessed the authentication values C_{i_2} and C_{i_4} without asking oracle h. This game and earlier game are indistinguishable unless the instances $\Pi_{U_i}^i$ and $\Pi_{S_j}^i$ rejects a valid authentication value. Hence, we have

$$\Pr[Succ_3] - \Pr[Succ_2] = q_h/p \,(4)$$

Game G_4 In this game, the group session key is guessed without asking the corresponding oracle h so that it become independent of password and ephemeral keys s which is protected by the chaotic maps-based computational DiffieHellman problem. We change the way with earlier game unless A queries h on the common value $H(ID_{session}||T_{session}(x)||s||T_S)$. Thus,$Adv_G^{cdh}(A) \ge \frac{1}{q_h}|\Pr[Succ_4] - \Pr[Succ_3]| - \frac{1}{p}$, that is, the difference between the game G_4 and the game G_3 is as follows:

$$|\Pr[Succ_4] - \Pr[Succ_3]| \le q_h Adv_G^{cdh}(A) + q_h/p \,(5)$$

Game G_5 This game is similar to the game G_4 except that in Test query, the game is aborted if A asks a hash function query with $H(ID_{session}||T_{session}(x)||s||T_S)$. A gets the session key $GSK_{session}$ by hash function query with probability at most . Hence, we have

$$|\Pr[Succ_5] - \Pr[Succ_4]| \leq q_h^2/2p \,(6)$$

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 q_s^2/D . Combining the Eqs. 1-6 one gets the announced result as:

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

4.2. **Further Security Discussion of the Proposed Scheme. Proposition 1** The proposed scheme could provide users anonymity with unlinkability.

Proof There are no plaintext in the two messages of the proposed M-PAKE phase. The message $\{T_{r_i}(x), C_{i_1}, C_{i_2}\}$ includes covered ciphertext $\{T_{r_i}(x), C_{i_1}\}$ which can transmit any important information to appointed node with the peers public key, such as identity and timestamp in the proposed scheme, and message $< C_{i_2} >$ is the verification ciphertext using one-way secure hash function. The other message $\{C_{i_3}, C_{i_4}\}$ includes covered ciphertext $\{C_{i_3}\}$ and verification ciphertext $\{C_{i_4}\}$. All the covered ciphertexts are protected by chaotic maps-based Computational DiffieHellman Problem and all the verification ciphertexts use a one-way secure hash function. Additionally, no message part is repeated in consecutive communications. This shows that our scheme achieve unlinkability property along with anonymity.

Proposition 2 The proposed scheme could withstand privileged-insider attack.

Proof During the registration phase, a legal user Ui submits masked password $H(R||PW_i)$ to the server instead of password PW_i , where R is a randomly selected value. Thus, an insider cannot achieve the password PW_i due to the non-retrieval property of the one-way hash function $H(\cdot)$. Moreover, the insider cannot guess the password as user does not submit R to the server. This shows that the proposed scheme resists insider attack.

Proposition 3 The proposed scheme could resist stolen verifier attack.

Proof In the proposed scheme, the server stores nothing about the legal users information. All the en/decrypted messages can be deal with the servers secret key which is equivalent to CDH problem in chaotic maps, so the proposed scheme withstands the stolen verifier attack.

Proposition 4 The proposed scheme could resist off-line password guessing attack.

Proof In this attack, an adversary may try to guess a legal user Uis password $PW - i$ using the transmitted messages. In our proposed scheme, there is no password involved in any transmitted messages, so the off-line password guessing attack cannot be launched.

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

Proof The login and verification messages include the timestamp. Therefore, an adversary cannot repeat the messages, since the maximum transmission delay ΔT is very short in communication. More important thing is that all the timestamps are protected by CDH problem in chaotic maps which only can be uncovered by the legal users (using B^* and ri) or the legal server (using k). So our proposed scheme resists the replay and man-in-the-middle attacks.

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 Ui to successfully login to the server. For any adversary, there are two ways to carry this attack: (1) The adversary may try to login to the server using the replay attack. However, the proposed scheme resists the replay attack. (2) The adversary A may try to generate a valid login message $\{T_{r_A}(x), C_{A_1}, C_{A_2}\}$ for a random value a and current timestamp TA, where $C_{A_2} = H(B^*||C_{A_1})$. Howerer, the adversary cannot compute C_{A_2} as computation of C_{A_2} requires B^* which is only known to legal user. It is clear that the adversary cannot generate valid login message. This shows that the proposed scheme resist user impersonation attack.

Proposition 7 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 Ui. For any adversary, there are two ways to carry this attack: (1) The adversary may try to login to the server using the replay attack. This attempt cannot succeed as the login and response message includes timestamp, and the proposed scheme resists the replay attack. (2) The adversary may try to generate a valid response message $\{C_{i_3}, C_{i_4}\}$ for current timestamp TS, where $C_{i_3} = T_k T_{r_i}(x)(ID_{session}||T_{session}(x)||s||T_S)$ and $C_{i_4} = H(B^*||GSK_{session})$. But this requires the secret key k of the server. This shows that our proposed scheme has the ability to resist the server impersonation attack.

Proposition 8 The proposed scheme could support mutual authentication.

Proof In our scheme, the server verifies the authenticity of users request by verifying the condition $H(B^*||C_{i_1}) \stackrel{?}{=} C_{i_2}$ during the proposed M-PAKE phase. To compute $\{C_{i_1}, C_{i_2}\}$, Uis password and B^* are needed. Therefore, an adversary cannot forge the message. Additionally, C_{i_1} includes timestamp, the adversary cannot replay the old message. This shows that the server can correctly verify the message source. Ui also verifies the authenticity of the server with the condition $H(B^*||GSK_{session}) \stackrel{?}{=} C_{i_4}$, which also requires the servers sercet key k for computing B^*. This shows that the user Ui can also correctly verify the server challenge. Hence, mutual authentication between Ui and the server can successfully achieve in our scheme.

TABLE 1. Security of our proposed protocol

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	S16	S17
[4](2008)	No	Mutual	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes	/	No
[5](2014)	No	Mutual	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	/	No
[6](2015)	No	Mutual	Yes	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	No	Yes
Ours	Yes	Mutual	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

S1: Single registration; **S2**: Authentication; **S3**: Privacy protection; **S4**: Resistance to privileged-insider attack
S5: Resistance to stolen-verifier attack; **S6**: Resistance to guessing attacks (On-line or off-line) (Including Prevent Password Guessing Attacks for privileged-insider or for any adversary); **S7**: Resistance to man-in-the-middle attack and replay attack
S8: Resistance to impersonation attack; **S9**: key freshness property; **S10**: known key secrecy property; **S11**: Perfect forward secrecy;
S12: forward secrecy property; **S13**: Dynamic privacy for members join or revocation; **S14**: Update password phase
S15: Formal security proof **S16**: Hiding timestamp **S17**: M-PAKE **Yes/No**: Support/Not support /: Not mentioned

Proposition 9 The proposed scheme could have Key freshness property.

Proof Note that in our scheme, each established group session key $GSK_{session} = H(ID_{session}||T_{session}(x)||s||T_S)$ includes timestamp T_S, and random values r_i and s. The timestamp are used to achieve the freshness for each session. Uniqueness property of timestamp, guaranties the unique key for each session. The unique key construction for each session shows that proposed scheme supports the key freshness property.

Proposition 10 The proposed scheme could have known key secrecy property.

Proof In our scheme, if a previously established group session key $GSK_{session} = H(ID_{session}||T_{session}(x)||s||T_S)$ is compromised, the compromised session key reveals no information about other session keys due to following reasons: (1) Each session key is hashed with one-way hash function. Therefore, no information can be retrieved from the session key. (2) Each session key includes timestamp, 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 11 The proposed scheme could have forward secrecy property.

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

Proposition 12 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 group session key has not included the servers long-term secret key k. This shows that our scheme provides the perfect forward secrecy property.

From the **Table 1**, we can see that the proposed scheme can provide privacy protection, 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. Comparisons between our proposed scheme and the related literatures

Protocols (Authentication phase)		**[5] (2014)**	**[6] (2015) (M-PAKE, $n = 2$)**	**Ours (M-PAKE, $n = 2$)**
Computation	For a user	$3T_{EXP} + 7T_H$	$4T_{EM} + T_{INV} + 4T_H + T_{SE}$	$3T_H + 2T_{MUL} + T_{EC}$
	Server	$3T_{EXP} + 8T_H$	$2nT_{EM} + nT_{SD} + (2n+1)T_H$	$4nT_H + 2nT_{MUL} + nT_{EC}$
	Total	$6T_{EXP} + 15T_H$	$(4+2n)T_{EM} + T_{INV} + nT_{SD} + (2n+5)T_H + T_{SE}$	$(4n+3)T_H + (2n+2)T_{MUL} + (n+1)T_{EC}$
Communication	Messages	8	2	2
	rounds	4	3	2
Design	Concise design	No	Yes	Yes
	Number of nonces	3	2	2
	Model	/	/	Random Oracle

5. **Efficiency Analysis.** On an Intel Pentium4 2600 MHz processor with 1024 MB RAM, where n and p are 1024 bits long, the computational time of some common algorithms can be summarized as follows [8, 16];

T_{Mac}:the time for executing a strongly unforgeable MAC algorithm computation;

T_F:the time for executing a secure pseudorandom function computation;

$T_{MUL/EXP/INV}$:the time for computing a modular multiplication/exponentiation/inversion $T_{EXP} \approx 240T_{MUL}$, $T_{INV} \approx 10T_{MUL}$);

T_H:the time for computing a one-way hash function computation ($T_H \approx 4T_{MUL}$);

$T_{EM/EA}$:the time for computing a point multiplication/addition operation over an elliptic curve($T_{EM} \approx 29T_{MUL}$, $T_{EA} \approx 0.12T_{MUL}$);

$T_{SE/SD}$:the time for performing a symmetric encryption/decryption algorithm computation ($T_{SE} \approx T_H \approx 4T_{MUL}$, $T_{SD} \approx T_H \approx 4T_{MUL}$);

T_{EC}:the time for executing enhanced Chebyshev polynomial ($T_{EC} \approx 60T_H$);

T_{XOR}: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 literatures of [5, 6].

As in **Table 1** and **Table 2**, we can draw a conclusion that the proposed scheme has achieved the improvement of both efficiency and security.

6. **Conclusion.** In the paper, we discusse the merits and demerits of the existing M-PAKE schemes and show that the existing schemes are failing to satisfy desirable attributes, and propose a new M-PAKE with dynamic privacy based on chaotic maps. 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.

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

REFERENCES

[1] W. Diffie ,M.E. Hellman, New directions in cryptography, *IEEE Trans. Inf. Theory* vol. 22, no. 6, pp. 644654, 1976.

[2] S. M. Bellovin and M. Merritt, Encrypted key exchange: Password-based protocols secure against dictionary attacks, *in Proc. of 1992 IEEE Computer Society Conference on Research in Security and Privacy*, pp. 72-84, May 4-6, 1992.

[3] H. Wang, H. Zhang, J.Li and C. Xu, A (3,3) visual cryptography scheme for authentication. *Journal of Shenyang Normal University*(Natural Science Edition), vpl.31, no.101(03), pp. 397-400, 2013.

[4] J. O. Kwon, I. R. Jeong and D. H. Lee, Practical Password-Authenticated Three-Party Key Exchange, *KSII Transactions on Internet and Information Systems*, vol. 2, no. 6, pp. 312-332, December, 2008.

[5] M. S. Farash and M. A. Attari, An enhanced and secure three-party password-based authenticated key exchange protocol without using server's public-keys and symmetric cryptosystems, *Information Technology And Control*, vol. 43, no. 2, pp. 143-150, 2014.

[6] C. F. Lu, Multi-party Password-Authenticated Key Exchange Scheme with Privacy Preservation for Mobile Environment, *KSII Transactions on Internet and Information Systems*, vol. 9, no. 12, pp. 5135 5149, 2015.

[7] 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.

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

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

[10] X. Guo, , J.Zhang, Secure group key agreement protocol based on chaotic hash. *Journal of Inf. Sci.* vol.180, no. 20, pp. 4069-4074, 2010.

[11] Y. Liu, K. Xue, An improved secure and efficient password and chaos-based two-party key agreement protocol, Nonlinear Dyn. *Published online:* 23 November 2015. DOI 10.1007/s11071-015-2506-2.

[12] H. F. Zhu, A Provable One-way Authentication Key Agreement Scheme with User Anonymity for Multi-server Environment, KSII trans.on internet and information systems, vol. 9, no. 2, pp. 811-829, Feb. 2015.

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

[14] Bergamo, Pina, et al. Security of public-key cryptosystems based on Chebyshev polynomials. Circuits and Systems I: Regular Papers, *IEEE Transactions on* vol. 52, no.7, pp. 1382-1393, 2005.

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

[16] N. Koblitz, A. Menezes and S. Vanstone, The state of elliptic curve cryptography, *Designs, Codes and Cryptography*, vol. 19, no. 2, pp. 173-193, March, 2000.