Journal of Information Hiding and Multimedia Signal Processing ©2015 ISSN 2073-4212
Ubiquitous International Volume **6**, Number **5**, September **2015**

A Novel Client-to-Client Password-Authenticated Key Exchange Protocol Using Chaotic Maps in The Standard Model

Hong-Feng Zhu,Yu Xia, and Yan Zhang

Software College, Shenyang Normal University
no. 253, HuangHe Bei Street, HuangGu District, Shenyang, P.C 110034 - China
zhuhongfeng1978@163.com; 876526606@qq.com; 1505733680@qq.com

Received December, 2014; revised July, 2015

Abstract. *Until now, the overwhelming majority of password-authenticated key agreement protocols using chaotic maps are based on three architectures (client/server, two clients/server and multi-server) and four security models (heuristic security, random oracle, ideal cipher and standard model). However, with rapid changes in the modern communication environment such as wireless mesh networks and cloud storing, it is necessary to put forward a kind more flexible and general architecture to adapt it. Moreover, most of the key exchange schemes adopting chaotic maps are usually by symmetric cryptography for exchanging some information. This will lead to a high calculated amount. So, the paper will wipe out the symmetric cryptography, and only use chaotic maps, a secure pseudo-random function to construct a provable secure two-party in two-realm key agreement protocol in the standard model. Our proposed protocol is more general and it is easy to expand to many other forms, such as three-party or N-party in different realms. The new protocol resists dictionary attacks mounted by either passive or active network intruders, allowing, in principle, even weak password phrases to be used safely. It also offers perfect forward secrecy, privacy protection and some others security attributes. Finally, we give the security proof in the standard model and the efficiency analysis of our proposed scheme.*

Keywords: Different realms, Key exchange, Mutual authentication, Chaotic maps, Privacy Protection

1. **Introduction.** Nowadays, chaos theory has widely used to cryptography. Chaotic system 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. In a word, chaos theory and chaotic system have exploited a new way for cryptography.

In 1998, Baptista [1] firstly connects cryptography with chaos theory. As a fundamental cryptographic primitive, key agreement protocol allows two or more parties to agree on shared keys which will be used to protect their later communication. Then, conbiming chaos theory and key agreement primitive, many authenticated key exchange (AKE) protocols [2-9] have been proposed. The literature [3] firstly proposed a new one-way authenticated key agreement scheme (OWAKE) based on chaotic maps with multi-server architecture. The OWAKE scheme is widely used to no need for mutual authentication environment on Internet, such as readers-to-journalists model and patient-to-expert model. Using the chaotic maps, the literature [4] firstly proposed a new multiple servers to server architecture (MSTSA) to solve the problems caused by centralized architecture, such as multi-server architecture with the registration center (RC). The core ideas of the proposed scheme are the symmetry (or called peer to peer) in the servers side and the transparency for the clients side. In brief, based on chaotic maps, there were many AKE protocols from functionality aspect, or from efficiency aspect, or from security aspect, or from architecture aspect to improve the AKE protocols.

However it is quite unrealistic that two clients trying to communicate with each other are registered on the same server. In the real situation with distributed applications, an authentication setting usually occurs such that two clients are registered in different servers. For example, from a users point of view in a mobile computing environment, a secure end-to-end channel between one mobile user in cell A and another user in cell B may be a primary concern. Additionally, the end-to-end security service minimizes the interferences from the operator controlled network components. Over the past years, many protocols based on the different password authentication (DPWA) model have been presented in the cross-realm setting and some of them have been easily broken and subsequently modified [12-17]. Byun et al. first proposed a Client-to-Client Password-Authenticated Key Exchange (C2C-PAKE) in the cross-realm setting where two clients are in two different realms and hence two servers involved [12]. Unfortunately, the scheme was found to be flawed. Chen first pointed out that one malicious server in the cross-realm setting could mount a dictionary attack to obtain the password of a client who belongs to the other realm [14]. In [17], Wang et al. showed dictionary attacks by a malicious server on the same protocol. Kim et al. [15] pointed out that the protocol was susceptible to Dening-Sacco attacks [18], and they also proposed an improved C2C-PAKE protocol. However, very recently, Phan and Goi suggested two unknown key share attacks on the improved C2C-PAKE protocol. They presented countermeasures in [16]. Up until now, several countermeasures to protect the attacks on the C2C-PAKE protocol have been presented in [13-17]. Recently Byun[19] presented an efficient C2C-PAKE protocol and proved it is secure under decisional Diffie-Hellman assumption in the ideal cipher and random oracle models. But most of the presented protocols were susceptible to Off-line Password Guessing Attacks with Server Compromise. The main reason [11, 20] is that there is a need for the password to encrypt or decrypt some information during the protocol process. This implies that the server has to store the plaintext password. So the password verification information in the server obtained by the attacker may mount an Off-line Password Guessing Attacks.

Based on the chaotic maps, we believe the more general architecture should be involved in AKE protocols. So we propose the first two-party in two-realm key exchange protocol using chaotic maps in standard model.

The rest of the paper is organized as follows: Some preliminaries are given in Section 2. Next, a novel chaotic maps problem is described in Section 3. Then, the non-interactive twin chaotic maps-key exchange protocol is given in Section 4. The Security of our proposed protocol is given in Section 5. The efficiency analysis of our proposed protocol and some feasible applications are given in Section 6. This paper is finally concluded in Section 7.

2. Preliminaries.

2.1. Pseudo-random function ensembles.

If a function ensemble $F = F_{n_{n \in N}}$ is pseudo-random [21], then for every probabilistic polynomial oracle A and all large enough n, we have that:

$$Adv^F(A) = |pr[A^{F_n}(1^n) = 1] - pr[A^{G_n}(1^n) = 1]| < \varepsilon(n)$$

where $G = \{G_n\}_{n \in N}$ is a uniformly distributed function ensemble, $\varepsilon(n)$ is a negligible function, $Adv^F = max_A\{Adv^F(A)\}$ denotes all oracle A, and $Adv^F(A)$ represents the accessible maximum.

2.2. **Definition and hard problems of Chebyshev chaotic maps.** Let n be an integer and let x be a variable with the interval $[-1, 1]$. The Chebyshev polynomial [6-9]$T_n(x) : [-1, 1] \to [-1, 1]$ is defined as $T_n(x) = cos(ncos_{-1}(x))$. Chebyshev polynomial map $T_n : R \to R$ of degree n is defined using the following recurrent relation:

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

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

The first few Chebyshev polynomials are:

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

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

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

In order to enhance the security, Zhang [10] proved that semi-group property holds for Chebyshev polynomials defined on interval $(-\infty, +\infty)$. The enhanced Chebyshev polynomials are used in the proposed protocol:

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

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 2.1. *(Semi-group property) Semi-group property of Chebyshev polynomials:* $T_{rs}(x) = T_r(Ts(x)) = cos(rcos^{-1}(scos^{-1}(x))) = cos(rscos^{-1}(x)) = T_s(T_r(x)) = T_{sr}(x)$, *where r and s are positive integer and $x \in [-1, 1]$.*

Definition 2.2. *(Chaotic Maps-Based Discrete Logarithm (CDL) problem)*
Given x and y, it is intractable to find the integer s, such that $T_s(x) = y$. The probability that a polynomial time-bounded algorithm A can solve the CDL problem is defined as $Adv_A^{CDL}(p) = Pr[A(x, y) = r : r \in Z_p^*, y = T_r(x)\ mod\ p]$.

Definition 2.3. *(CDL assumption) For any probabilistic polynomial time-bounded algorithm A, $Adv_A^{CDL}(p)$ is negligible, that is, $Adv_A^{CDL}(p) \leq \epsilon$, for some negligible function ϵ.*

Definition 2.4. *(Chaotic Maps-Based Diffie-Hellman (CDH) problem) Given x, $T_r(x)$ and $T_s(x)$, it is intractable to find $T_{rs}(x)$. The probability that a polynomial time-bounded algorithm A can solve the CDH problem is defined as $Adv_A^{CDH}(p) = Pr[A(x, T_r(x)\ mod\ p, T_s(x)\ mod\ p) = T_{rs}(x)\ mod\ p : r, s \in Z_p^*]$.*

Definition 2.5. *(CDH assumption) For any probabilistic polynomial time-bounded algorithm A, $Adv_A^{CDH}(p)$ is negligible, that is, $Adv_A^{CDL}(p) \leq \epsilon$, for some negligible function ϵ.*

2.3. **Practical Environment.** Now we set a prototype example in practical environment. (1) (2) (3) (4) (5) denote the five rounds in Fig.1 respectively. We assume Alice wants to establish a session key with Bob. So the initiator Alice broadcasts (A, B, $Server_A$, $Server_B$) in (1). Because Alice and Bob have already registered on respective $Server$, each $Server$ can use registered verifiers and ephemeral random numbers to authenticate respective client with the same realm in (2) (3). In (4) $Server_A$ and $Server_B$

will deliver the sensitive information to each other with Chaotic maps cryptosystem after authenticating respective client. $Server_A$ and $Server_B$ will use the peer server's public key to authenticate each other. In (5), $Server_A$ sends sensitive information to Alice and finally Alice use sensitive information and the her own secret ephemeral random number to compute the session key (The same to $Server_B$ and Bob).

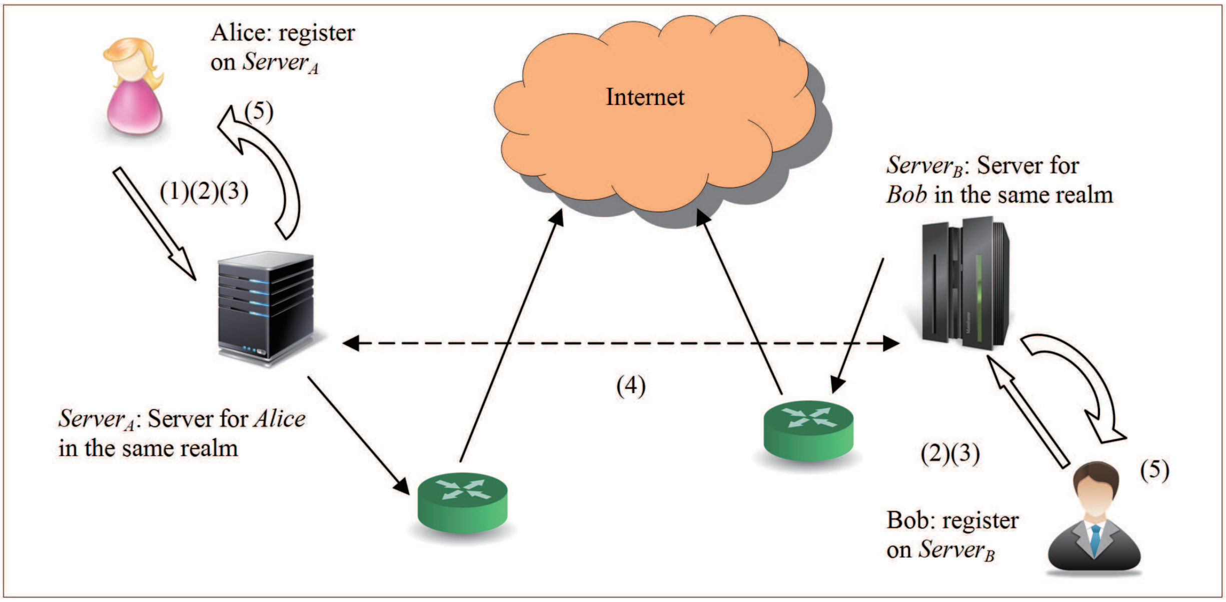

FIGURE 1. An example for practical environment of two-realm AKE

3. The Proposed Protocol.

In this section, under the two-realm environment for two client with two servers, a chaotic maps-based authentication key agreement scheme is proposed which consists of three phases: registration phase, authentication key agreement phase and password update phase.

3.1. Notations.

In this section, any server i has its identity ID_{S_i} and public key $(x, T_{K_i}(x))$ and a secret key K_i based on Chebyshev chaotic maps and a pseudo-random function F. The concrete notations used hereafter are shown in Table1.

TABLE 1. Notations

Symbol	Definition
$ID_A, ID_B, ID_{Session}$	the identity of Alice, Bob and the session respectively
S_i, ID_{S_i}	The ith server, the identity of the ith server, respectively
TID_i	The temporary identity of party i
$t, a, b, S_a, S_b, S_{aa}, S_{bb}$	nonces
$(x, T_K(x))$	public key based on Chebyshev chaotic maps
K	secret key based on Chebyshev chaotic maps
F	pseudo-random function
$\|\|$	concatenation operation

3.2. **Registration phase.** Concerning the fact that the proposed scheme mainly relies on the design of Chebyshev chaotic maps-based in two-realm architecture, it is assumed that Alice can register at the $server_A$ in the same realm by secure channel. The same assumption can be set up for servers. Fig.2 illustrates the server registration phase.

Step 1. When a user Alice wants to be a new legal user, she chooses her identity ID_A and password PW_A. Then Alice computes $Q = F_{PW_A}(ID_A||PW_A)$ and sends $\{ID_A, Q\}$ to the $server_A$ via a secure channel.

Step 2. The $server_A$ computes $FPW_A = F_Q(Q||K_A)$, stores $\{ID_A, FPW_A\}$ securely and sends FPW_A to Alice.

Step 3. Upon receiving FPW_A from the $server_A$, Alice stores FPW_A in a secure way.

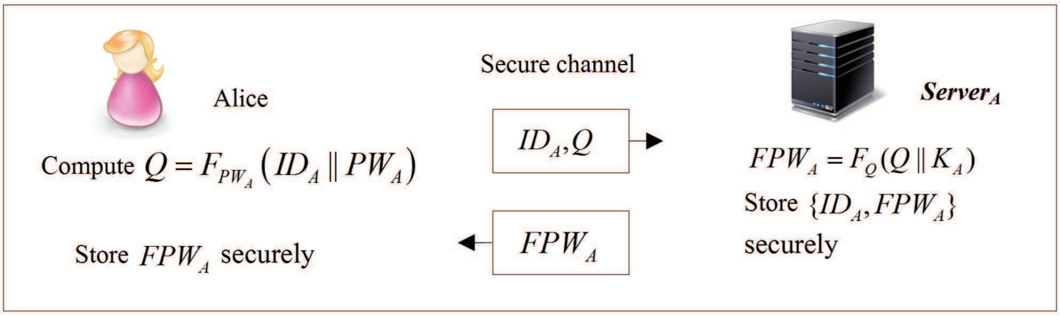

FIGURE 2. Server or a authenticated expert registration phase

3.3. **Preprocessing of TID.** For simplicity, we construct a function to produce TID_i, the temporary identity of party i for clients or servers. Without loss of generality, we assume party i sends a TID_i to party j using $(x, T_{K_i}(x))$ for covering ID_i but only party j can recover the ID_i.

The party i selects a large and random integer t, and computes $T_t(x)$, $C_t = T_tTK_i(x)ID_i$, $FTID_i = F_{ID_i}(C_t||T_t(x))$. Then the party i sends $\{T_t(x), C_t, FTID_i\}$ to the party j. After receiving the message $\{T_t(x), C_t, FTID_i\}$ from the party i, the party j will use $T_t(x)$ and his own secret key K_j to recover $ID_i = C_t/T_{K_j}T_t(x) = C_t/T_tT_{K_i}(x)$. Then the party j check if $F_{ID_i}(C_t||T_t(x)) \stackrel{?}{=} FTID_i$. If above equation holds, the party j deems the ID_i is legal identity. Otherwise, the party j terminates the session.

3.4. **Authenticated key agreement phase.** We omit the production of temporary identity TID_i. The concrete process can be found in the Section 3.3.

The authenticated key agreement phase is presented in the following Fig. 3.

Step 1. If Alice wishes to consult some personal issues establish with Bob in a secure way, but they are in different realm. Alice will choose a large and random a. Then the device of Alice will compute $T_a(x)$, $C_{A_1} = T_a(x)T_{FPW_A}T_{K_A}(x)$ and $Mac_{AS} = F_{T_aT_{K_a}(x)}(ID_{Session}||C_{A_1})$. After that, Alice sends $\{TID_A, TID_B, C_{A_1}, Mac_{AS}\}$ to $Server_A$ where she registers on (The same way for Bob).

Step 2. After receiving the message $\{TID_A, TID_B, C_{A_1}, Mac_{AS}\}$ from Alice, $Server_A$ will do the following tasks: (1) $Server_A$ uses FPW_A to compute $T_a(x) = C_{A_1}/T_{FPW_A}T_{K_A}(x)$. (2) $Server_A$ examines whether $Mac_{AS} = F_{T_aT_{K_a}(x)}(ID_{Session}||C_{A_1})$ is valid in terms of the $(ID_{Session}||C_{A_1})$. (3) $Server_A$ selects a large and random integer S_a to compute $T_{S_a}(x)$, $C_{A_2} = T_a(x)T_{S_a}T_{K_B}(x)$, $Mac_{SAB} = F_{T_aT_{K_B}(x)}(ID_{Session}||C_{A_2})$ and sends $\{TID_A, TID_B, C_{A_2}, T_{S_a}(x), Ma$ to $Server_B$ (The same way for $Server_B$).

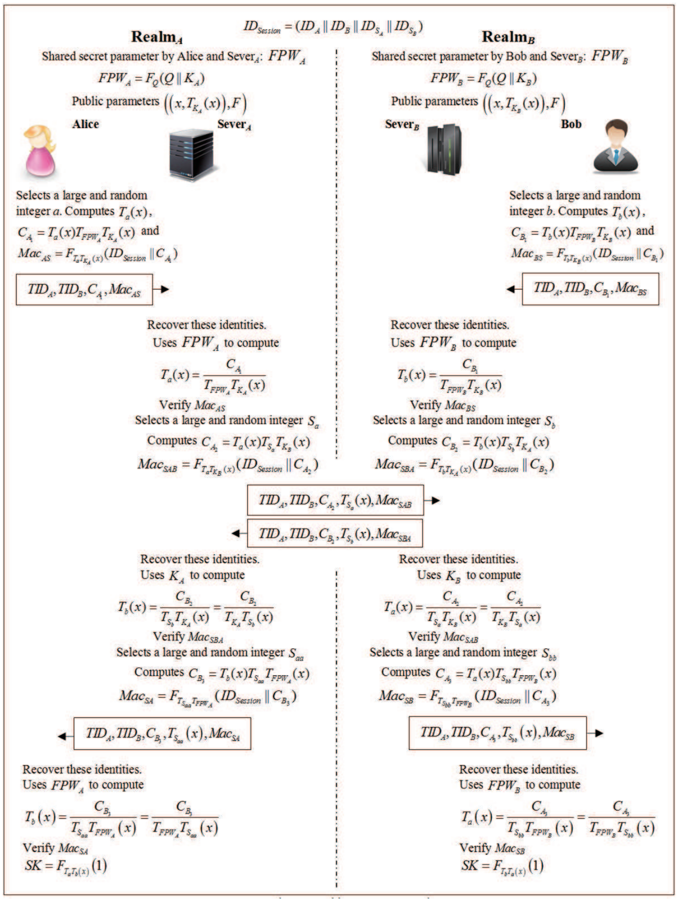

FIGURE 3. Authenticated key agreement phase

Step 3. After receiving the message $\{TID_A, TID_B, C_{A_2}, T_{S_a}(x), Mac_{SAB}\}$ from $Server_A$, $Server_B$ will uses K_B to compute $T_a(x) = C_{A_2}/T_{S_a}T_{K_B}(x) = C_{A_2}/T_{K_B}T_{S_a}(x)$. Then $Server_B$ examines whether $Mac_{SAB} = F_{T_aT_{K_B}(x)}(ID_{Session||C_{A_2}})$ is valid in terms of the $(ID_{Session||C_{A_2}})$. $Server_B$ selects a large and random integer S_{bb} and computes $T_{S_{bb}}(x)$, $C_{A_3} = T_a(x)T_{S_{bb}}T_{FPW_B}(x)$, $Mac_{SB} = F_{T_aT_b(x)}(ID_{Session}||C_{A_3})$ and sends $\{TID_A, TID_B, C_{A_3}, T_{S_{bb}}(x), Ma$ to Bob (The same way for $Server_A$).

Step 4. After receiving the message $\{TID_A, TID_B, C_{A_3}, T_{S_{bb}}(x), Mac_{SB}\}$, Bob uses FPW_B to compute $T_a(x) = C_{A_3}/T_{S_{bb}}T_{FPW_B}(x) = C_{A_3}/T_{FPW_B}T_{S_{bb}}(x)$. Then Bob examines whether $Mac_{SB} = F_{T_{FPW_B}T_{S_{bb}}(x)}(ID_{Session||C_{A_3}})$ is valid in terms of the $(ID_{Session||C_{A_3}})$. If holds, and the session key is $SK = F_{T_bT_a(x)}$. (The same way for Alice).

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

3.5. **Password update phase.** This concrete process is presented in the following Fig. 4.

Step 1. If Alice wishes to update her password with $Server_A$, Alice will choose a new memorable password PW'_A. Then the device of Alice will compute $Q' = F_{PW'_A}(ID_A||F_{PW'_A})$, $C_{A_1} = Q'T_{FPW_A}T_{K_A}(x)$ and $Mac_{AS} = F_{Q'}(ID_A||ID_{S_A}||C_{A_1})$. After that, Alice sends $\{TID_A, C_{A_1}, Mac_{AS}\}$ to $Server_A$ where she registers on.

Step 2. After receiving the message $\{TID_A, C_{A_1}, Mac_{AS}\}$ from Alice, $Server_A$ will do the following tasks: (1) $Server_A$ uses FPW_A to compute $Q' = C_{A_1}/T_{FPW_A}T_{K_A}(x)$. (2) $Server_A$ examines whether $Mac_{AS} = F'_Q(ID_A||ID_{S_A}||C_{A_1})$ is valid in terms of the

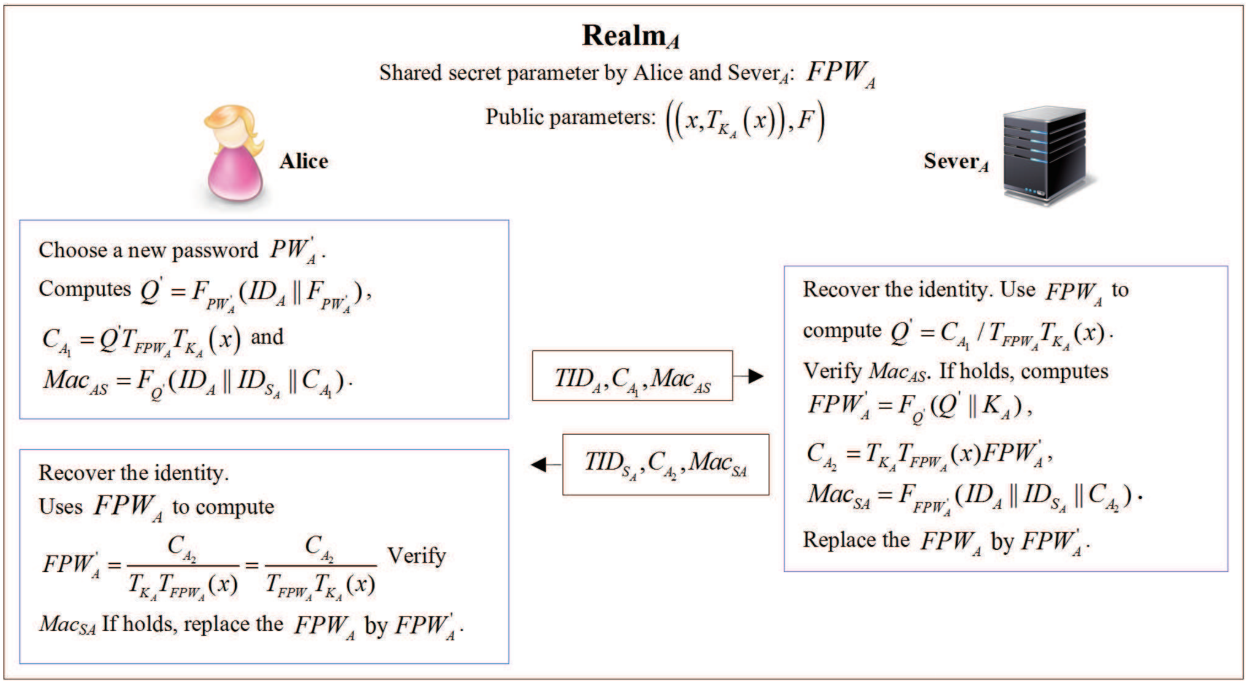

FIGURE 4. Password update phase

$(ID_A||ID_{S_A}||C_{A_1})$. (3) If holds, $Server_A$ computes $FPW'_A = F'_Q(Q'||K_A)$, $C_{A_2} = T_{K_A}T_{FPW_A}(x)FPW'_A$, $Mac_{SA} = F_{FPW'_A}(ID_A||ID_{S_A}||C_{A_2})$ and sends $\{TID_{S_A}, C_{A_2}, Mac_{SA}$ to Alice. Replaces the FPW_A by FPW'_A.

Step 3. After receiving the message $\{TID_{S_A}, C_{A_2}, Mac_{SA}$ from $Server_A$, Alice will uses FPW_A to compute $FPW'_A = C_{A_2}/T_{K_A}T_{FPW_A}(x) = C_{A_2}/T_{FPW_A}T_{K_A}(x)$. Then Alice computes $Mac'_{SA} = F_{FPW'_A}(ID_A||ID_{S_A}||C_{A_2})$ to verify Mac_{SA}. If holds, Alice replaces the FPW_A by FPW'_A.

4. **Security Consideration.** The section a theorem concerning the semantic security of our proposed protocol is given.

4.1. **Security Model.** We recall the protocol syntax and communication model [22-24]. The basic descriptions and some queries are shown in Table 2.

4.2. **Security Proof. Theorem 2.** Let Γ be a two-party in two-realm PAKE protocol described in Fig.3. Let $F : \{0,1\}^n \rightarrow \{0,1\}^{l(n)}$ be a pseudo-random function ensembles. Because the DDH assumption holds in enhanced Chebyshev chaotic maps, then

$$Adv^{2P2RPAKE}_{x,T_u,F}(t,R) \leq \frac{2q_e^2+3q_s^2+2(q_e+q_s)^2}{N_1} + 2(q_e+q_s)Adv^F + 2(min\{q_e,q_r\}+min\{q_s,q_r\})Adv^F$$
$$+2(q_e+q_s)Adv^{DDH}_{x,T_u} + \frac{q_s}{2^{n-1}} + \frac{(q_e+q_s)^2}{N_1}\frac{q_s}{N}$$

where n is a safe parameter, $l(\cdot)$ is a function that can be computed in polynomial time. N_1 is a large prime number, $u, T_u(x)$ are the private and public keys of the server, q_e, q_r, q_s represent the maximum number of Execute and Test that the adversary can inquire, and queries from Send-Client and Send-Server, N is the password dictionary D's size, $Adv^{DDH}_{x,T_u(x)}$ represents the probability of breaking the DDH hypothesis, and Adv^F denotes the probability of breaking the pseudo-random function ensembles.

In order to make the security proof simple, we firstly point out the differences between the literature [23] and our proposed protocol. Then we give the differences between the literature [24] and our proposed protocol. Finally, we will get the **theorem 2.**

TABLE 2. Descriptions the model and the queries

Symbol	Definition
parties $P_1, \ldots P_n$ or $(C_1, \ldots C_n, S_1, \ldots S_n)$	Modeled by probabilistic Turing machines. Two non-empty sets: User, the set of all clients, and Server, the set of trusted servers constitute the participants in our 2P2RPAKE protocol.
Adversary Λ	A probabilistic Turing machine which controls all communication, with the exception that the adversary cannot inject or modify messages (except for messages from corrupted parties or sessions), and any message may be delivered at most once.
Sessiosns matching	If the outgoing messages of one are the incoming messages of the other
$\Pi^i_{U_1}, pid^i_{U_1}, sid^i_{U_1}$ $\Pi^j_{U_2}, pid^j_{U_2}, sid^j_{U_2}$	Denote participant U_1's instance i, who is involved with a partner participant U_2 in a session. $\Pi^i_{U_1}$ has the partner identification $pid^i_{U_1}$ and the session identification $sid^i_{U_1}$. The same means for $\Pi^j_{U_2}, pid^j_{U_2}, sid^j_{U_2}$.
Execute $(\Pi^i_{U_1}, S^i, S^j, \Pi^j_{U_2})$	This query returns the messages that were communicated in the course of an honest execution of the protocol among $\Pi^i_{U_1}, S^i, S^j, \Pi^j_{U_2}$.
Send-Client $(\Pi^i_{U_k}(k=1,2), m)$	This query returns the message that client instance $\Pi^i_{U_k}$, which would generate upon receipt of message m.
Send-Server $(S^k(k=1,2), m)$	This query returns the message that server instance S^k would generate upon receipt of message m. When receiving a fabricated message by an adversary, the server instance S^k responds in the manner prescribed by the protocol.
Corrupt $(U_k(k=1,2))$	This query returns the session key of the client instance $U_k(k=1,2)$.
Reveal $(\Pi^i_{U_k}(k=1,2))$	This query returns the password and the states of all instances of $U_k(k=1,2)$ only when it is defined.
Test $(\Pi^i_{U_k}(k=1,2))$	This query allows the adversary to be issued at any stage to a completed, fresh, unexpired session. A bit b is then picked randomly. If $b=0$, the test oracle reveals the session key, and if $b=1$, it generates a random value in the key space. The adversary Λ can then continue to issue queries as desired, with the exception that it cannot expose the test session.
Partnering	We say two instances $\Pi^i_{U_1}$ and $\Pi^j_{U_2}$ are partners iff: (a) They are successfully accepted. (b) $sid^i_{U_1} = sid^j_{U_2}$. (c) pid for $\Pi^i_{U_1}$ is $\Pi^j_{U_2}$ and vice versa. (d) No instance other than $\Pi^i_{U_1}$ and $\Pi^j_{U_2}$ accepts with a pid equal to $\Pi^i_{U_1}$ or $\Pi^j_{U_2}$.
Freshness	Let $\Pi^i_{U_1,U_2,S_1,S_2}$ be a completed session by a party U_1 with some other party U_2, and $\Pi^j_{U_2,U_1,S_2,S_1}$ be the matching session to $\Pi^i_{U_1,U_2,S_1,S_2}$. We say that the session $\Pi^i_{U_1,U_2,S_1,S_2}$ is fresh if U_1 and U_2 in session $\Pi^i_{U_1,U_2,S_1,S_2}$ and the matching session $\Pi^j_{U_2,U_1,S_2,S_1}$ are honest and the following conditions hold: (a) $\Pi^i_{U_1,U_2,S_1,S_2}$ has accepted the request to establish a session key. (b) $\Pi^i_{U_1,U_2,S_1,S_2}$ has not been revealed. (c) No matching conversation $\Pi^j_{U_2,U_1,S_2,S_1}$ of $\Pi^i_{U_1,U_2,S_1,S_2}$ has been revealed. (d) U_2, S have not been corrupted. (e) The adversary asks neither Send-Client $(\Pi^i_{U_1}, m)$ nor Send-Client $(\Pi^j_{U_2}, m)$ query.

(1) The differences between the literature [23] and our proposed protocol.

Using enhanced Chebyshev chaotic maps to replace ElGamal encryption. To be specific, g^{x_2}, rg^{x_1}, Zg^{x_1} and $g^{x_1}h^{x_2}$ in the literature [23] should be replaced by $T_{x_2}(x)$, $rT_{x_1}(x)$, $ZT_{x_1}(x)$ and $T_{x_1}(x)T_{x_2}$, respectively.

The birthday paradox should be used to replace the probability of random events when the event collision occurs. According to the birthday paradox, the probability of collisions in output $T_n(x)$ is at most $q_s^2/2N_1$, where q_s denotes the maximum number of Send-Client and Send-Server queries.

According to the birthday paradox, the probability of collisions in output $T_n(x)$ is at most $(q_s + q_e)^2/2N_1$, where q_s denotes the maximum number of Send-Client and Send-Server queries, q_e denotes the maximum number of Execute queries. Hence, the probability of distinguishing Mac_{**} with random integers is $(q_s + q_e)^2/2N_1$.

(2) The differences between the literature [24] and our proposed protocol.

We convert the low entropy secret password PW to high entropy cryptography key by a pseudo-random function $FPW_A = F_Q(Q||K_A)$ which is more secure way than the literature [24] only stored password in the server database.

Different architecture. Our proposed protocol sets up in different realm and the two-party has different password with his/her service server. That means one Send-Client query will test two passwords in the same set. So in our protocol, when relating with N (N is the password dictionary D's size), and it is the same with the literature [24].

Round 1. Our proposed protocol has one more Mac_{**} for each party, so there is must have one more $(q_s + q_e)^2/2N_1$.

Round 2. The only difference between the literature [24] and our proposed protocol is that one server changes into two servers. So that brings about two points changed: (1) There are two more Mac_{**}, so the probability of distinguishing Mac_{**} with random integers is $(q_s + q_e)^2/2N_1$. (2) According to the birthday paradox, there are two more $T_n(x)$, so the probability of collisions in output $T_n(x)$ is at most q_s^2/N_1.

Round 3. It is the same with the literature [24].

The detailed descriptions of these games and lemmas are analogous to those in literature [24], with the differences discussed above, and therefore, they are omitted.

Theorem 3 Our proposed two-realm PAKE protocol ensures key privacy against the server based on the fact that DDH assumption holds in the enhanced Chebyshev chaotic maps and F is a secure pseudo-random function ensemble, and

$$Adv_D^{K_P}(\Lambda_{K_P}) \leq 4q_s Adv_{x,T_u(x)}^{DDH} + 2q_e Adv^F,$$

where q_e and q_s denote the maximum number of queries to the oracle Execute and Send-Client.

The proof of **Theorem 3** is similar to that of Theorem 5.2 in [23] and Theorem 3 in [24]. The difference between our proposed protocol and the literature [23] is that we just replace the enhanced Chebyshev chaotic map values with the ElGamal discrete logarithm values. The difference between our proposed protocol and the literature [24] is that our proposed protocol is designed in different realm with different password, so some changed details can be described in the section 4.2(2).

Next, from the Table 3, we can see that the proposed scheme can provide secure session key agreement, perfect forward secrecy, and privacy protection and so on. As a result, the proposed scheme is more secure and has much functionality compared with the recent related scheme.

Table 3. Security comparison existing protocols for 3PAKE based on Chebyshev chaotic maps and our protocol

	Model	PP	KP	MA	AR	FS	UDOD	UKS	PCI	OFD
Our protocol	S	Yes	Yes	Yes	C2S2	Yes	Yes	Yes	Yes	Yes
Yang and Cao's protocol [23]	S	No	Yes	Yes	C2S	Yes	Yes	Yes	Yes	Yes
Lai et al.'s protocol [24]	S	No	Yes	Yes	C2S	Yes	Yes	Yes	Yes	Yes
Yoon-Jeon's protocol [25]	N	No	No	Yes	C2S	No	No	Yes	No	No
Xie et al.'s protocol [26]	N	No	Yes	Yes	C2S	Yes	Yes	Yes	No	No
Lee et al.'s protocol [17]	N	Yes	Yes	Yes	C2S	No	No	Yes	Yes	No

S standard model, *N* nonstandard model, *PP* privacy protection, *KP* key privacy, *MA* mutual authentication, *AR* architecture, *C2S* client-to-server, *C2S2* Two-client-two-server, *FS* forward security, *UDOD* security against undetectable on-line dictionary attack, *UKS* security against unknown key-share attack, *PCI* security against password compromised impersonation attack, *OFD* security against off-line dictionary attack.

5. **Efficiency Analysis.**

5.1. **The comparisons between our scheme and the literature in different realms with different algorithms.** 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 [9] proposed several methods to solve the Chebyshev polynomial computation problem.

To be more precise, on an Intel Pentium4 2600 MHz processor with 1024 MB RAM, where n and p are 1024 bits long, the computational time of a symmetric encryption/decryption operation, an elliptic curve point multiplication operation and Chebyshev polynomial operation is 0.0087s, 0.063075s and 0.02102s separately [18]. Moreover, the computational cost of XOR operation could be ignored when compared with other operations.

Our proposed protocol is mainly based on chaotic maps algorithms which is more efficient than the other algorighms, such as RSA and ECC, in the literatures [19]. The the literature [16] is about cryptanalysis of an improved client-to-client password-authenticated key exchange scheme. Table 4 given the comparison for RSA, ECC and Chaotic maps.

Table 4. Comparison for RSA, ECC and Chaotic maps

	RSA encryption algorithm	**ECC encryption algorithm**	**Chaotic maps encryption algorithm**
Items	***Differences***		
Mathematical basis	Large prime number	Elliptic curve	Chebyshev polynomial
Difficult problem assumptions	large prime factorization problem	Discrete logarithm calculation problem on the elliptic curve	Chaotic maps discrete logarithm problem, Chaotic maps Diffie-Hellman problem
Operation Cost	√ √	√ √ √	√ √ √
Operation Speed	√	√ √	√ √ √
Security Level	√	√ √ √	√ √ √
Normal √	Good √ √	Excellent √ √ √	

For simplicity, the literatures [19] in the different realms architecture, we give comparisons table detailedly in Table 5.

Table 5. Comparisons of computational and communicating costs in authenticated key agreement phase

	Byun and lee protocol [19]		Our protocol	
	A / *B*	$server_A$ / $server_B$	*A* / *B*	$server_A$ / $server_B$
Exponent operation	3/3	2/2	0/0	0/0
Chaotic maps	0/0	0/0	2/2	4/4
Hash operation	1/1	1/1	0/0	0/0
Pseudo-random function	0/0	0/0	3/3	4/4
XOR operation	0/0	0/0	0/0	0/0
Nonces	2/2	2/1	1/1	2/2
Communicating round	8		3	
Symmetric encryption/decryption	3/3	4/4	0/0	0/0

5.2. **The comparisons between our scheme and the literature with the same algorithms.** Table 6 shows performance comparisons between our proposed scheme and the literature of [17, 23-26] in three-party architecture with chaotic maps.

Table 6. Cost comparison existing protocols for 3PAKE

The others Our protocol	R	RN (A/B/S) ($A/B/S_A/S_B$)	PKE (A/B/S) ($A/B/S_A/S_B$)	SKE (A/B/S) ($A/B/S_A/S_B$)	*T* (A/B/S) ($A/B/S_A/S_B$)	*H* (A/B/S) ($A/B/S_A/S_B$)	*D* (A/B/S) ($A/B/S_A/S_B$)	*F* (A/B/S) ($A/B/S_A/S_B$)
Our protocol	3	1/1/1/1	0/0/2/2	0/0/0/0	2/2/4/4	0/0/0/0	0/0/0/0	2/2/4/4
Yang and Cao's protocol [23]	4	2/2/3	0/0/1	0/0/1	0/0/0	0/0/0	0/0/0	4/4/2
Lai et al.'s protocol [24]	4	2/2/3	0/0/1	0/0/1	6/6/10	0/0/0	0/0/0	4/4/2
Yoon-Jeon's protocol [25]	5	2/1/0	2/2/0	1/1/1	2/2/0	2/0/2	1/1/2	0/0/0
Xie et al.'s protocol [26]	6	1/1/1	2/2/0	3/3/0	3/3/2	5/5/4	2/2/4	0/0/0
Lee et al.'s protocol [17]	5	1/1/1	2/2/0	4/4/0	3/3/2	4/4/7	0/0/0	0/0/0

R Round, *RN* Random number, *PKE* Public key encryption, *SKE* Secret key encryption. A: participant A, B: participant B, S: Single Server, S_A: $Server_A$, S_B: $Server_B$, *T*, *D*, *H* and *F* represent the time for performing a Chebyshev polynomial computation, a symmetric encryption/decryption, a one-way hash function and pseudo-random function, respectively.

6. **Conclusions.** In this paper, we conduct a comprehensive and general study of two-party in different realms PAKE protocol over standard model using chaotic maps. Most existing researches are concerning about concrete environment, such as two-party AKE or three-party AKE based on chaotic maps, but as far as we know, there is no general and extensible architecture about different realms based on chaotic maps has been proposed. However, through our exploration, we firstly clarify that the PAKE scheme using chaotic maps in different realms is more suitable for the real environment. Then, 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.

REFERENCES

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

[2] C. Lee, C. Li, and C. Hsu, A three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps, *Nonlinear Dyn*, vol. 73, pp. 125-132, 2013.

[3] H.F. Zhu, Y. F. Zhang and Y. Zhang, A One-Way Authentication Key Agreement Scheme with User Anonymity Based on Chaotic maps towards Multi-Server Architecture, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 6, no. 2, pp. 274-287, March 2015.

[4] H. F. Zhu, M. Jiang, X. Hao and Y. Zhang, Robust Biometrics-based Key Agreement Scheme with Smart Cards towards a New Architecture, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 6, no. 1, pp. 81-98, January 2015.

[5] H. F. Zhu, X. Hao, Y. F. Zhang and M. Jiang, A biometrics-based multi-server key agreement scheme on chaotic maps cryptosystem, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 6, no. 2, pp. 211-224, March 2015.

[6] L. R. Devaney, An Introduction to Chaotic Dynamical System, *Cummings Publishing Company Inc.*, The Benjammin, Menlo Park, 1986.

[7] J. C. Jiang, and Y. H. Peng, Chaos of the Chebyshev polynomials, *Nat. Sci. J. Xiangtan Univ*, vol. 19, no. 3, pp. 3739, 1996.

[8] L. Zhang, Cryptanalysis of the public key encryption based on multiple chaotic systems, *Chaos Solitons Fractals*, vol. 37, no. 3, pp. 669674, 2008.

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

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

[11] M. Zhang, New Approaches to Password Authenticated Key Exchange Based on RSA, *In ASIACRYPT*, pp. 230244, 2004.

[12] J. Byun, I. Jeong, D. Lee, C. Park, Password-authenticated key exchange between clients with different passwords, *In: Proceedings of ICICS02, LNCS*, vol. 2513, Springer-Verlag, pp. 134146, 2002.

[13] J. Byun, I. Jeong, D. Lee, C. Park, Password-authenticated key agreement between clients with different passwords, working draft of ISO/IEC 11770-4 in document 27 N 3576, 2003.

[14] L. Chen, A weakness of the password-authenticated key agreement between clients with different passwords scheme, ISO/IEC JTC 1/SC27 N3716.

[15] J. Kim, S. Kim, J. Kwak, D. Won, Cryptoanalysis and improvements of password authenticated key exchange scheme between clients with different passwords, *In: Proceedings of ICCSA 2004*, LNCS, vol. 3044, Springer-Verlag, 2004, pp. 895902.

[16] R. C.-W. Phan, B. Goi, Cryptanalysis of an improved client-to-client password-authenticated key exchange (C2C-PAKE) scheme, *In: Proceedings of ACNS 2005*, LNCS, vol. 3531, Springer-Verlag, 2005, pp. 3339.

[17] C. C. Lee, C. T. Li,and C. W. Hsu, A three-party passwordbased authenticated key exchange protocol with user anonymity using extended chaotic maps, *Nonlinear Dyn*, 73, 125132 (2013).

[18] L. Kocarev, and S. Lian, Chaos-Based Cryptography: Theory, Algorithms and Applications, pp. 5354, 2011.

[19] J. W. Byun, D. H. Lee, J. I. Lim, EC2C-PAKA: An efficient client-to-client password-authenticated key agreement, *INFORMATION SCIENCES*, pp, 1-19, 2007.

[20] T. Wu, The secure remote password protocol, *In Internet Society Network and Distributed System Security Symposium (NDSS)*, pp. 97111, 1998.

[21] V. Shoup, Sequences of games: a tool for taming complexity in security proofs. report 2004/332, *International Association for Cryptographic Research (IACR)*, 2004.

[22] R. Canetti, and H. Krawczyk, Analysis of key-exchange protocols and their use for building secure channels, *EUROCRYPT*, vol. 2045 of Lecture Notes in Computer Science, pp. 453-474, 2001.

[23] J. H. Yang, T. J. Cao, Provably secure three-party password authenticated key exchange protocol in the standard model, *J. Syst. Softw,*, vol. 85, pp. 340350, 2012.

[24] H. Lai, A. M. Orgun, J. H. Xiao, et al, Provably secure three-party key agreement protocol using Chebyshev chaotic maps in the standard model, *Nonlinear Dyn*, vol. 77, pp. 14271439, 2014.

[25] E. J. Yoon, and I. S. Jeon, An efficient and secure Diffie-Hellman key agreement protocol based on Chebyshev chaotic map, *Commun. Nonlinear Sci. Numer. Simul.16*, 23832389, 2011.

[26] Q. Xie, J. M. Zhao, and X. Y. Yu, Chaotic maps-based threeparty password-authenticated key agreement scheme, *Nonlinear Dyn.74*, 10211027, 2013.