Password-Authenticated Key Exchange Scheme Using Chaotic Maps towards a New Architecture in Standard Model

Hongfeng Zhu, Yifeng Zhang, Yu Xia, and Haiyang Li
(Corresponding author: Hongfeng Zhu)

Software College, Shenyang Normal University
no. 253, HuangHe Bei Street, HuangGu District, Shenyang 110034, China
(Email:zhuhongfeng1978@163.com@qq.com)
(Received Dec. 12, 2014; revised and accepted Apr. 11 & June 23, 2015)

Abstract

Nowadays, 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. So, in our paper, we firstly propose a provable secure password authenticated key agreement protocol using chaotic maps towards multiple servers to server architecture in the standard model. The multiple servers to server architecture will solve the problems single-point of security, single-point of efficiency and single-point of failure in the centralized registration center towards multi-server architecture. 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, which protects past sessions and passwords against future compromises. Finally, we give the security proof in the standard model and the efficiency analysis of our proposed scheme.

Keywords: Chaotic maps, key exchange, multiple servers to server, mutual authentication

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 [7, 8, 12, 16, 21, 23, 24, 25] have been proposed. The literature [25] 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 [24] 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 server side and the transparency for the client 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.

Recently, Multi-server authenticated key agreement (MSAKA) architecture is more popular among the AKE protocols which aim to register at the registration center for log in other servers without register repeatedly. MSAKA protocols mainly want to solve the problems in a traditional single server with authentication schemes [2, 5, 22] which lead to the fact that user has to register to different servers separately. On a macro level MSAKA protocols can be divided into three phases in chronological order:

1) The creative phase: The pioneer work in the field

was proposed by Li et al. [9] in 2001. However, Lin et al. [13] pointed out that Li et al.s scheme takes long time to train neural networks and an improved scheme based on ElGamal digital signature and geometric properties on the Euclidean plane has also been given.

2) The development phase: the main work in this phase is amended repeatedly. For example, Tsai [15] also proposed an efficient multi-server authentication scheme based on one-way hash function without a verification table. Because Tsai scheme only uses the nonce and one-way hash function, the problems associated with the cost of computation can be avoided in the distributed network environment. However, some researchers [6] pointed out that Tsai scheme is also vulnerable to server spoofing attacks by an insider server and privileged insider attacks, and does not provide forward secrecy.

3) The diversification phase: the research emphasis shifts to functionality. Therefore, identity-based MSAKA protocols, based on bilinear pairings or elliptic curve cryptosystem (ECC) MSAKA protocols, dynamic identity-based MSAKA protocols and other MSAKA protocols came up recently [3, 6, 17].

Based on the chaotic maps, we believe MSAKA protocols is not a general solution because only one centralized registration center cannot handle so complex network environment. So based on our previous studies [24], we believe that we should design an AKE protocol in a more general architecture. So we propose the first towards multiple servers to server architecture 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 One-way Hash Function and Pseudo-random Function Ensembles

A secure cryptographic one-way hash function h: $a \to b$ has four main properties:

1) The function h takes a message of arbitrary length as the input and produces a message digest of fixed-length as the output;

2) The function h is one-way in the sense that given a, it is easy to compute $h(a) = b$. However, given b, it is hard to compute $h^{-1}(b) = a$;

3) Given a, it is computationally infeasible to find a' such that $a' \neq a$, but $h(a') = h(a)$;

4) It is computationally infeasible to find any pair a, a' such that $a' \neq a$, but $h(a') = h(a)$.

Pseudo-random function ensembles:

If a function ensemble $F = \{F_n\}_{n\in N}$ is pseudo-random [14], then for every probabilistic polynomial oracle $\mathcal{A}$ and all large enough n, we have that:

$$\mathcal{A}^{G_n}(1^n) = 1]| < \varepsilon(n)$$

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

2.2 Symmetric Encryption

A symmetric encryption scheme $E_k(Kgen, E, D)$ consists of three algorithms as follows:

1) Randomized Key Generation Algorithm $Kgen$:it returns a key k drawn from the key space $Keys(E_k)$ at random.

2) Encryption Algorithm E: it takes the key $k \in Keys(E_k)$ and a plaintext $M \in \{0,1\}^*$ as the inputs and outputs a ciphertext $C \in \{0,1\}^*$. So it can be written $C = E_k(M)$.

3) Decryption Algorithm D:it takes the key $k \in Keys(E_k)$ and a ciphertext $C \in \{0,1\}^*$ as the inputs and outputs a plaintext $M \in \{0,1\}^*$. So it can be written $M = D_k(C)$.

2.3 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 [16] $T_n(x) : [-1, 1] \to [-1, 1]$ 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 $n \geq 2$, $T_0(x) = 1$, and $T_1(x) = x$. The first few Chebyshev polynomials are:

$$\begin{aligned} T_2(x) &= 2x^2 - 1, \\ T_3(x) &= 4x^3 - 3x, \\ T_4(x) &= 8x^4 - 8x^2 + 1, \\ \cdots &\quad \cdots \end{aligned}$$

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

Definition 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* $\mathcal{A}$ *can solve the CDL problem is defined as* $Adv_{\mathcal{A}}^{CDL}(p) = \Pr[\mathcal{A}(x, y) = r : r \in Z_p^*, y = T_r(x) \bmod p]$.

Definition 3. *(CDL assumption) For any probabilistic polynomial time-bounded algorithm* $\mathcal{A}$*,* $Adv_{\mathcal{A}}^{CDL}$ *(p) is negligible, that is,* $Adv_{\mathcal{A}}^{CDL}(p) \leq \varepsilon$*,for some negligible function* ε.

Definition 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* $\mathcal{A}$ *can solve the CDH problem is defined as* $Adv_{\mathcal{A}}^{CDH}(p) = \Pr[\mathcal{A}(x, T_r(x) \bmod p, T_s(x) \bmod p) = T_{rs}(x) \bmod p : r, s \in Z_p^*]$.

Definition 5. *(CDH assumption) For any probabilistic polynomial time-bounded algorithm* $\mathcal{A}$*,* $Adv_{\mathcal{A}}^{CDH}(p)$ *is negligible, that is,* $Adv_{\mathcal{A}}^{CDH}(p) \leq \varepsilon$*, for some negligible function* ε.

2.4 Definition and Properties of Chebyshev Chaotic Maps [7, 8]

Definition 6. $f : J \to J$ *is said to be topologically transitive if for any pair of open sets* $U, V \subset J$*, there exists* $k > 0$ *such that* $f^k(U) \cap V \neq \phi$.

Definition 7. $f : J \to J$ *has sensitive dependence on initial conditions if there exists* $\delta > 0$ *such that for any* $x \in J$ *and any neighborhood* N *of* x*, there exist* $y \in N$ *and* $n \geq 0$ *such that* $|f^n(x) - f^n(y)| > \delta$.

Definition 8. *Let* V *be a set, then* $f : V \to V$ *is said to be chaotic on* V *if*

1) f *has sensitive dependence on initial conditions.*

2) f *is topologically transitive.*

3) Periodic points are dense in V.

Definition 9. *Let* $f : A \to A$*,* $f : B \to B$ *be two maps, if there exists a continuous surjection* $h : A \to B$ *such that* $h \cdot g = g \cdot h$*,we say that these two maps* f *and* g *are topologically semi-conjugate.*

Theorem 1. *A non-zero polynomial is the* n^{th} *Chebyshev polynomial or its constant times iff the nonzero polynomial is the root of the differential equation*

$$(1 - x^2)y'' - xy' + n^2y = 0(n \in \mathrm{Z}_+).$$

Lemma 1. *If* $f : A \to A$*,* $f : B \to B$ *are topologically semi-conjugate, (1) when* p *is the periodic point of* f*, then* $h(p)$ *is the periodic point of* g*;(2) when the periodic point of* f *is dense in* A*, the periodic point of* g *is dense in* B*,where* h *is the topologically semi-conjugate between* f *and* g.

Lemma 2. *Assume* $f : A \to B$ *is a map,* $A_0, A_1 \subset A$*, then* $f(A_0 \cap A_1) \subset f(A_0) \cap f(A_1)$.

Lemma 3. *When* $f : A \to A$ *is topologically transitive,* $g : B \to B$ *is topologically semi-conjugate* f *via* h*, then* g *is topologically transitive.*

Lemma 4. *Let* $R : S' \to S'$ *be a map of the circle into itself, then* $R(\theta) = n\theta(n \in \mathrm{Z}, n \geq 2)$ *is chaotic,where* θ *is the radian value.*

The concrete proof of chaotic properties can be found in the literature [8] and the enhanced properties of Chebyshev polynomials that defined on interval $(-\infty, +\infty)$ *still have the semi-group property (see [21]).*

2.5 Practical Environment

The literature [24] firstly proposed a new multiple servers to server architecture (MSTSA), and now we set a prototype example in practical environment. (1)(2)(3)(4)(5) denote the five rounds in Figure 1 respectively. We assume Alice wants to establish a session key with $\mathbf{Server}_B$ for getting the service of $\mathbf{Server}_B$. So the initiator Alice broadcasts (A, $\mathbf{Server}_A$, $\mathbf{Server}_B$) in (1). Because Alice have already registered on $\mathbf{Server}_A$, $\mathbf{Server}_A$, can use registered verifiers and ephemeral random numbers to authenticate Alice for helping $\mathbf{Server}_B$ in (2) (3). In (4) $\mathbf{Server}_A$ and $\mathbf{Server}_B$ will deliver the sensitive information to each other with Chaotic maps cryptosystem after authenticating each other. At the same time, $\mathbf{Server}_B$ will compute the session key with Alice after authenticating Alice and $\mathbf{Server}_A$. In (5), $\mathbf{Server}_A$ sends sensitive information to Alice and finally Alice use sensitive information and the her own secret ephemeral random number to compute the session key with $\mathbf{Server}_B$. (The same way for other servers and users)

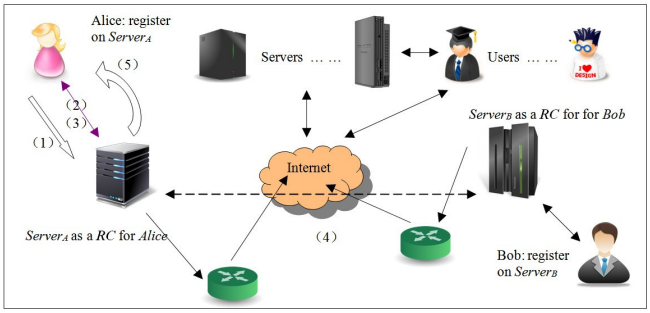

Figure 1: An example for practical environment of multiple servers to server architecture

3 The Proposed Protocol

In this section, under the multiple servers to server architecture, a chaotic maps-based password 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, a secure one-way hash function $H(\cdot)$, a pseudo-random function F, and a pair of secure symmetric encryption/decryption functions $E_K()/D_K()$ with key K. The concrete notations used hereafter are shown in Table 1.

Table 1: Notations

Symbol	Definition
$ID_A, ID_{Session}$	the identity of Alice and the session respectively
S_i, ID_{S_i}	The ith server, the identity of the ith server, respectively
a, S_a, S_b, S_{aa}	nonces
$(x, T_K(x))$	public key based on Chebyshev chaotic maps
K	secret key based on Chebyshev chaotic maps
$E_K() / D_K()$	a pair of secure symmetric encryption/decryption functions with the key K
H	A secure one-way hash function
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 multiple servers to server architecture, it is assumed that Alice can register at the **Server**$_A$ by secure channel and view the **Server**$_A$ as her own registration center to login on other servers for some services. The same assumption can be set up for users. Figure 2 illustrates the user 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 $HPW_A = H(ID_A||PW_A||T_{K_A}(x))$ and sends $\{ID_A, HPW_A\}$ to the server via a secure channel.

Step 2. Upon receiving $\{ID_A, HPW_A\}$ from the Alice, the **Server**$_A$ stores $\{ID_A, HPW_A\}$ in a secure way.

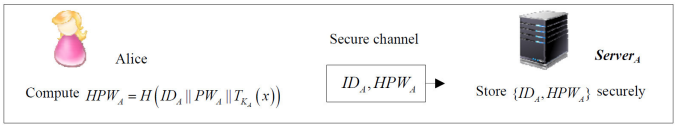

Figure 2: Server or a authenticated expert registration phase

3.3 Authenticated Key Agreement Phase

This concrete process is presented in the following Figure 3.

Step 1. If Alice wishes to consult some personal issues establish with **Server**$_B$ in a secure way, but Alice has not register at **Server**$_B$. So in our multiple servers to server architecture, Alice need not register at **Server**$_B$ and she just uses her account at the **Server**$_A$ to login in **Server**$_B$. Alice will choose a large and random a. Then the device of Alice will compute $T_a(x)$,$C_{A_1} = T_a(x)T_{HPW_A}T_{K_A}(x)$ and $Mac_{AS} = F_{T_aT_{K_A}(x)}(ID_{Session}||C_{A_1})$. After that, Alice sends $ID_A, ID_{S_B}, C_{A_1}, Mac_{AS}$ to **Server**$_A$ where she registers on.

Step 2. After receiving the message: ID_A, ID_{S_B}, C_{A_1}, Mac_{AS} from Alice, **Server**$_A$ will do the following tasks:

1) **Server**$_A$ uses HPW_A to compute $T_a(x) = C_{A_1}/T_{HPW_A}T_{K_A}(x)$.
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 ID_A, ID_{S_A}, C_{A_2}, $T_{S_a}(x)$, Mac_{SAB} to **Server**$_B$.

Step 3. After receiving the message: ID_A, ID_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_b and computes $T_{S_{bb}}(x)$, $C_{A_3} = T_{S_{bb}}(x)T_{HPW_B}T_a(x)$,

$Mac_{SB} = F_{T_aT_b(x)}(ID_{Session}||C_{A_3})$ and sends $ID_A, ID_{S_B}, C_{A_3}, T_{S_b}(x), Mac_{SBA}$ to **Server**$_A$. And then **Server**$_B$ computes the session key is $SK = F_{T_{S_b}T_a(x)}(1)$.

Step 4. After receiving the message: ID_A, ID_{S_B}, C_{A_3}, $T_{S_b}(x)$, Mac_{SBA}, **Server**$_A$ uses K_A to compute $C'_{A_3} = T_{K_A}T_{S_b}(x) = T_{S_b}T_{K_A}(x) = C_{A_3}$. Then **Server**$_A$ examines whether $Mac_{SBA} = F_{T_{K_A}T_{S_b}}(ID_{Session}||C_{A_3})$ is valid in terms of the $(ID_{Session}||C_{A_3})$. If holds, **Server**$_A$ selects a large and random integer S_{aa} and computes $T_{S_{aa}}(x)$, $C_{A_4} = T_{S_{aa}}(x)T_{HPW_A}T_{S_b}(x)$, $Mac_{SA} = F_{T_{S_{aa}}T_{HPW_A}}(ID_{Session}||C_{A_4})$ and sends $ID_A, ID_{S_B}, C_{A_4}, T_{S_{aa}}(x), Mac_{SA}$ to Alice.

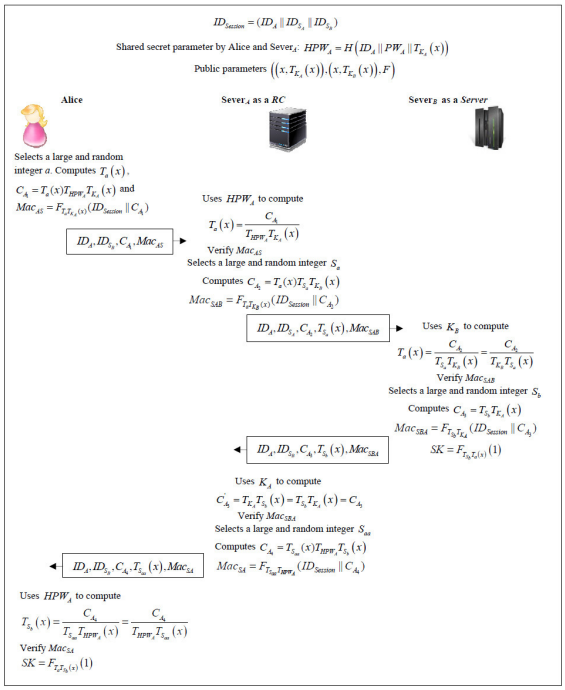

Figure 3: Authenticated key agreement phase

Step 5. After receiving the message: ID_A, ID_{S_B}, C_{A_4}, $T_{S_{aa}}(x)$, Mac_{SA} from **Server**$_A$, Alice will uses HPW_A to compute $T_S(x) = C_{A_4}/T_{S_{aa}}T_{HPW_A}(x) = C_{A_4}/T_{HPW_A}T_{S_{aa}}(x)$. Then Alice examines whether $Mac_{SA} = F_{T_{HPW_A}T_{S_{aa}}(x)}(ID_{Session}||C_{A_2})$ is valid in terms of the $(ID_{Session}||C_{A_4})$. If holds, Alice computes the session key is $SK = F_{T_aT_{S_b}(x)}(1)$. If any authenticated process does not pass, the protocol will be terminated immediately.

3.4 Password Update Phase

This concrete process is presented in the following Figure 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 $HPW'_A = H(ID_A||PW'_A||T_{K_A}(x))$, $T_{HPW'_A}(x)$, $C_{A_1} = T_{HPW'_A}(x)T_{HPW_A}T_{K_A}(x)$ and $Mac_{AS} = F_{T_{HPW'_A}T_{K_A}(x)}(ID_A||ID_{S_A}||C_{A_1})$. After that, Alice sends $ID_A, ID_{S_A}, C_{A_1}, Mac_{AS}$ to **Server**$_A$ where she registers on.

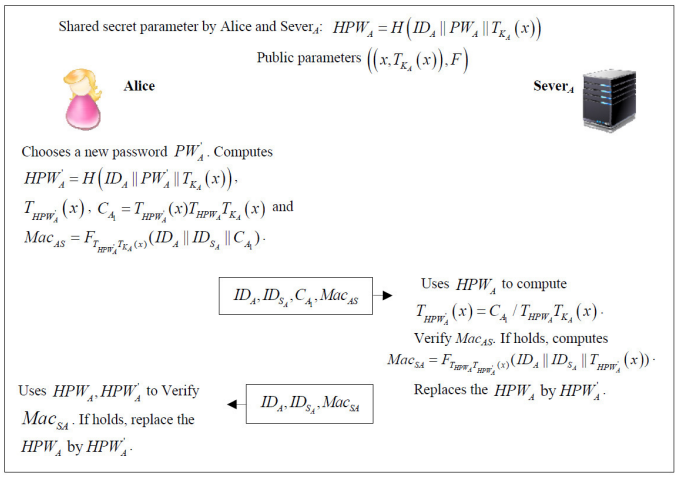

Figure 4: Password update phase

Step 2. After receiving the message: ID_A, ID_{S_A}, C_{A_1}, Mac_{AS} from Alice, **Server**$_A$ will do the following tasks:

1) **Server**$_A$ uses HPW_A to compute $T_{HPW'_A}(x) = C_{A_1}/T_{HPW_A}T_{K_A}(x)$.
2) **Server**$_A$ examines whether $Mac_{AS} = F_{T_{HPW'_A}T_{K_A}(x)}(ID_A||ID_{S_A}||C_{A_1})$ is valid in terms of the $(ID_A||ID_{S_A}||C_{A_1})$.
3) If holds, **Server**$_A$ computes $Mac_{SA} = F_{T_{HPW_A}T_{HPW'_A}(x)}(ID_A||ID_{S_A}||T_a(x))$ and sends ID_A, ID_{S_A}, Mac_{SA} to Alice. Replaces the HPW_A by HPW'_A.

Step 3. After receiving the message: ID_A, ID_{S_A}, Mac_{SA} from **Server**$_A$, Alice will uses HPW_A, HPW'_A to compute $Mac'_{SA} = F_{T_{HPW_A}T_{HPW'_A}(x)}(ID_A||ID_{S_A}||T_{HPW'_A}(x))$ to verify Mac_{SA}. If holds, Alice replaces the HPW_A by HPW'_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 [4, 11, 19]. The basic descriptions and some queries

are shown in Table 2.

Table 2: Descriptions the model and the queries

Symbol	Definition
parties $P_1,...P_n$ or $(C_1...C_n, S_1,...S_n)$	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 MSTSA-PAKE 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 $(\Pi^i_{U_1}, S^j, S^j, \Pi^j_{U_2})$	This query returns the messages that were communicated in the course of an honest execution of the protocol among $\Pi^i_{U_1}, S^j, S^j, \Pi^j_{U_2}$.
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.

4.2 Security Proof

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

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

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} 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 [19] and our proposed protocol. Then we give the differences between the literature [11] and our proposed protocol. Finally, we will get Theorem 2.

1) The differences between the literature [19] 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 [19] should be replaced by $T_{x_2}(x), rT_{x_1}(x), ZT_{x_1}(x)$ and $T_{x_1}(x)T_{x_2}(h)$, 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 [11] and our proposed protocol. We convert the low entropy secret password PW to high entropy cryptography key by a one-way hash function $HPW_A = H(ID_A||PW_A||T_{K_A}(x))$ which is more secure way than the literature [11] only stored password in the server database.

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

Our proposed protocol sets up in multiple servers to server architecture which has only one password with the RC server. That means one Send-Client query will test only one password in the same set. So in our protocol, when relating with N (N is the password dictionary D's size), and it is must be multiplied by $1/2$.

The detailed descriptions of these games and lemmas are analogous to those in literature [11], 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*

$$Adv^{k_p}_D(\Lambda_{k_p}) \leq 4q_s Adv^{DDH}_{x,T_u} + 2q_e Adv^F$$

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 [19] and Theorem 3 in [11]. The difference between

our proposed protocol and the literature [19] 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 [11] is that our proposed protocol is designed in different realm with different password, so some changed details can be described in 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 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	KP	MA	AR	FS	UDOD	UKS	PCI	OFD
Our protocol	S	Yes	Yes	MSTSA	Yes	Yes	Yes	Yes	Yes
Yang and Cao's protocol [23]	S	Yes	Yes	C2S	Yes	Yes	Yes	Yes	Yes
Lai et al.'s protocol [24]	S	Yes	Yes	C2S	Yes	Yes	Yes	Yes	Yes
Yoon-Jeon's protocol [25]	N	No	Yes	C2S	No	No	Yes	No	No
Xie et al.'s protocol [26]	N	Yes	Yes	C2S	Yes	Yes	Yes	No	No
Lee et al.'s protocol [27]	N	Yes	Yes	C2S	No	No	Yes	Yes	No

S standard model, *N* nonstandard model, *KP* key privacy, *MA* mutual authentication, *AR* architecture, *C2S* client-to-server, *MSTSA* multiple servers to server architecture, *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 among Our Scheme and Other Multi-server Architecture 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 [16] proposed several methods to solve the Chebyshev polynomial computation problem. For convenience, some notations are defined as follows.

T_{hash}**:** The time for executing the hash 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_{CH}**:** The time for executing the $T_n(x)modp$ in Chebyshev polynomial using the algorithm in the literature [10].

To be more precise, on an Intel Pentium4 2600 MHz processor with 1024 MB RAM, where n and p are 1024 bits long, the computational time of a one-way hashing operation, a symmetric encryption/decryption operation, an elliptic curve point multiplication operation and Chebyshev polynomial operation is 0.0005s, 0.0087s, 0.063075s and 0.02102s separately [10]. Moreover, the computational cost of XOR operation could be ignored when compared with other operations.

For simplicity, the literatures [3, 6, 13, 15] in the different realms architecture, we omit the comparisons table detailed. The reason is that our proposed protocol are mainly based on chaotic maps algorithms which is more efficient than the other algorithms, such as RSA and ECC, in the literatures [3, 6, 13, 15].

5.2 The Comparisons among Our Scheme and Other Algorithms

Table 4 shows performance comparisons between our proposed scheme and the literature of [11, 12, 18, 19, 20] in three-party architecture with chaotic maps.

Table 4: Cost comparison existing protocols for 3PAKE based on Chebyshev chaotic maps and our protocol

The others paper / Our protocol	R	RN ($A/B/S$) ($A/S_A/S_B$)	PKE ($A/B/S$) ($A/S_A/S_B$)	SKE ($A/B/S$) ($A/S_A/S_B$)	*T* ($A/B/S$) ($A/S_A/S_B$)	*H* ($A/B/S$) ($A/S_A/S_B$)	*D* ($A/B/S$) ($A/S_A/S_B$)	*F* ($A/B/S$) ($A/S_A/S_B$)
Our protocol	5	1/1/2	0/1/1	0/0/0	3/6/2	0/0/0	0/0/0	2/2/1
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 [27]	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$_A$ as ***RC***, 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 Conclusion

In this paper, we conduct a comprehensive and general study of PAKE protocol over standard model using chaotic maps towards multiple servers to server architecture. 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 distributed network environment based on chaotic maps has been proposed. However, through our exploration, we firstly clarify that the PAKE scheme using chaotic maps towards multiple servers to server architecture 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," in *Physics Letters A*, vol. 240, no. 1, pp. 50–54, 1998.

[2] J. W. Byun, Ik R. Jeong, D. H. Lee, C. S. Park, "Password-authenticated key exchange between clients with different passwords," in *Information and Communications Security (ICICS'02)*, LNCS 2513, pp. 134–146, Springer, 2002.

[3] J. W. Byun, D. H. Lee, J. I. Lim, "EC2C-PAKA: An efficient client-to-client password-authenticated key agreement, *Information Sciences*, vol. 177, no. 19, pp. 3995–4013, 2007.

[4] R. Canetti, and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Advances in Cryptography (EUROCRYPT'01)*, LNCS 2045, pp. 453–474, Springer, 2001.

[5] L. Chen, *A Weakness of the Password-authenticated Key Agreement between Clients with Different Passwords Scheme*, ISO/IEC JTC 1/SC27 N3716.

[6] D. Denning, G. Sacco, "Timestamps in key distribution protocols," *Communications of the ACM*, vol. 24, no. 8, pp. 533–536, 1981.

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

[8] J. C. Jiang, Y. H. Peng, "Chaos of the Chebyshev polynomials," *Natural Science Journal of Xiangtan University*, vol. 19, no. 3, pp. 37–39, 1996.

[9] J. Kim, S. Kim, J. Kwak, D. Won, "Cryptanalysis and improvements of password authenticated key exchange scheme between clients with different passwords," in *Proceedings of ICCSA'04*, LNCS 3044,pp. 895–902, Springer, 2004.

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

[11] H. Lai, M. A. Orgun, J. Xiao, et al, "Provably secure three-party key agreement protocol using Chebyshev chaotic maps in the standard model," *Nonlinear Dynamics*, vol. 77, pp. 1427–1439, 2014.

[12] C. C. Lee, C. T. Li, C. W. Hsu, "A three-party password based authenticated key exchange protocol with user anonymity using extended chaotic maps," *Nonlinear Dynamics*, vol. 73, pp. 125–132, 2013.

[13] R. C. W. Phan, B. Goi, "Cryptanalysis of an improved client-to-client password-authenticated key exchange (C2C-PAKE) scheme," in *Proceedings of ACNS'05*, LNCS 3531, pp. 33–39, Springer, 2005.

[14] V. Shoup, *Sequences of Games: A Tool for Taming Complexity in Security Proofs*, Report 2004/332, International Association for Cryptographic Research (IACR), 2004.

[15] S. Wang, J. Wang, M. Xu, "Weakness of a password-authenticated key exchange protocol between clients with different passwords," in *Proceedings of ACNS'04*, LNCS 3089, pp. 414–425, Springer, 2004.

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

[17] T. Wu, "The secure remote password protocol," in *Internet Society Network and Distributed System Security Symposium (NDSS'98)*, pp. 97–111, 1998.

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

[19] J. H. Yang, T. J. Cao, "Provably secure three-party password authenticated key exchange protocol in the standard model," *Journal of System Software*, vol. 85, pp. 340–350, 2012.

[20] E. J. Yoon, I. S. Jeon, "An efficient and secure Diffie-Hellman key agreement protocol based on Chebyshev chaotic map," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, pp. 2383–2389, 2011.

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

[22] M. Zhang, "New approaches to password authenticated key exchange based on RSA," in *Advances in Cryptology (ASIACRYPT'04)*, LNCS 3329, pp. 230–244, Springer, 2004.

[23] H. Zhu, X. Hao, Y. 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, Mar. 2015.

[24] H. Zhu, M. Jiang, X. Hao and Y. Zhang, "Robust biometrics-based key agreement scheme with smart cards towards a new architecture," in *Journal of Information Hiding and Multimedia Signal Processing*, vol. 6, no. 1, pp. 81–98, Jan. 2015.

[25] H. Zhu, Y. 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, Mar. 2015.

Hongfeng Zhu obtained his Ph.D. degree in Information Science and Engineering from Northeastern University. Hongfeng Zhu is a full associate 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 and international conference papers on the above research fields.

Yifeng Zhang, 24 years old, an undergraduate from Shenyang Normal University, major in information security management. During the four years of college, after completing her studies, he enjoys reading the book

related to this major. Under the guidance of the teacher, he has published two articles in EI journals.

Yu Xia is the graduate student of Shenyang Normal University in Computer science and technology. He once served as chairman of the student union, and won the national scholarship for the two times and several times to get a scholarship. He was named outstanding graduates of Liaoning Province, Shenyang outstanding college students and other honors. He was a member of the basketball team and participated in the track and field competition, and won the championship in the 1500 meters and 3000 meters of the project. He has research interests in network security, computer application, cloud computing. He published a monograph and four EI international journals on the above research fields.

Haiyang Lee, graduate, graduated from Liaoning University Population Research Institute, Master demographic now at Shenyang Normal University Dean's Office Examination Management Division, lecturers title. He researches on labor and social security, wireless computer networks, network security.