Provably Secure Multi-server Privacy-Protection System Based on Chebyshev Chaotic Maps without Using Symmetric Cryptography

Hongfeng Zhu, Yifeng Zhang, and Yang Sun
(Corresponding author: Hongfeng Zhu)

Software College, Shenyang Normal University
No. 253, HuangHe Bei Street, HuangGu District, Shenyang 110034, P. R. China
(Email:zhuhongfeng1978@163.com, {1548452125, 17247613}@qq.com)
(Received Mar. 15, 2015; revised and accepted May 5 & July 13, 2015)

Abstract

Most of the privacy-protection schemes adopting chaotic maps are usually by symmetric cryptography for guaranteeing identity hiding. This will lead to a high calculated amount. So, the paper will wipe out the symmetric cryptography, and only use chaotic maps, a secure one-way hash function to construct a provable privacy-protection system (PPS) which can achieve two kinds of privacy-protection and switch between them optionally by users: The first is anonymous scheme which can make nobody know the user's identity, including the server and the registration center (RC), and they only know these users are legal or paying members. The other is hiding scheme which owns also privacy-protection property, because the user's identity is not transferred during the process of the proposed protocol, and only the server and the *RC* know the user's identity. About practical environment, we adopt multi-server architecture which can allow the user to register at the *RC* once and can access all the permitted services provided by the eligible servers. Then a new PPS authenticated key agreement protocol is given based on chaotic maps. Security of the scheme is based on chaotic maps hard problems and a secure one way hash function. Compared with the related literatures recently, our proposed scheme can not only own high efficiency and unique functionality, but is also robust to various attacks and achieves perfect forward secrecy. Finally, we give the security proof and the efficiency analysis of our proposed scheme.

Keywords: Chaotic maps, key agreement, multi-server architecture, privacy-protection system

1 Introduction

Authenticated key exchange (AKE) is one of the most important cryptographic components which is used for establishing an authenticated and confidential communication channel. Based on the number of participants, we can divide AKE protocols into three categories: two-party AKE protocols [10], three-party AKE protocols [13], and N-party AKE protocols [3, 14, 25]. Furthermore, based on the respective features in detail, the previous AKE protocols [1, 2, 8, 11, 15, 17, 18, 20, 21, 22, 23] can be classified many categories, we use two-party AKE protocols to set an example: such as password-based [10], chaotic map-based [2], ID-based [25], anonymity [13, 23], secret sharing [21] and so on. Recently many researchers achieve AKE in the multi-server environment called multi-server authenticated key agreement (MSAKA) protocols. MSAKA protocols allow the user to register at the registration center (RC) once and can access all the permitted services provided by the eligible servers. In other words, users do not need to register at numerous servers repeatedly. MSAKA protocols mainly want to solve the problems in a traditional single server with authentication schemes [11] which lead to the fact that user has to register to different servers separately. About MSAKA protocols, the pioneer work in the field was proposed by Li et al. [15] in 2001. However, Lin et al. [17] 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. Next stage, the main work is amended repeatedly. For example, Tsai [22] proposed an efficient multi-server authentication scheme based on one-way hash function without a verification table. Because Tsais 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, the literature [20] pointed out that Tsais scheme is also vulnerable to server spoofing attacks by an insider server and privileged insider attacks, and does not provide forward secrecy. At the present stage, the research emphasis shifts to functionality and user experi-

ence. 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 [20].

However, there are many scenes need not mutual authentication at all and we just need one-way authentication. For example, readers act upon the perceived reputation of a news source, so reputation is a valuable commodity for journalists. No further authentication is required and since the information is public, channel secrecy is not required and does not affect the actions of either party. Another example, on Internet, patients requiring medical advice may wish to do so anonymously, while still ensuring the confidentiality of their request and assurance that the medical advice received comes from an authentic, qualified source. The key idea of one-way AKE is that one party wishes for no one to be able to determine his/her identity, including all the authorities. However, only a few protocols have considered the problem of one-way authentication. Goldberg [8] gave a specialized one-way AKE security definition for the Tor authentication protocol. The literature [1] described an identity-based anonymous authenticated key exchange protocol but with a limited session key secrecy definition based on key recovery, not indistinguishability. Morrissey et al. [18] analyzed the security of the Transport Layer Security (TLS) protocol in the context of one-way authentication, but with specialized security definitions. Recently, Goldberg and Stebila [9] provided an intuitive set of goals and present a formal model that captures these goals. Usually, public key encryption can be used for one-way AKE protocols, for example by having the client encrypt a session key under the server's public key. This mechanism is widely used, for example in the RSA-based cipher suites in TLS [6] and in the KAS1 protocol in NIST SP800-56B [19].

All above-mentioned scenes do not include a new scene of application: A user wants to consult with an authenticated expert anonymously or explicitly, and the expert does not want to provide the free service because of limited time or energy. Both mutual authenticated key agreement [10] and one-way authenticated key agreement protocol [27] cannot provide the solutions about this scene. Even for mutual authenticated key agreement protocol with privacy protection cannot solve it, because the scene needs transformation flexibly between anonymity and hiding identity. Therefore I propose the concept about privacy-protection system to solve the problem. In a meaning, the mutual authenticated key agreement protocol with privacy protection is the subset of the privacy-protection system.

The main contributions are shown as below: The paper firstly presents a new provable privacy-protection system towards multi-server architecture. Furthermore, the proposed protocol is mainly based on chaotic maps without using modular exponentiation and scalar multiplication on an elliptic curve. In Security aspect, the protocol can resist all common attacks, such as impersonation attacks, man-in-the-middle attacks, etc. About functionality, the protocol also has achieved some well-known properties, such as perfect forward secrecy and execution efficiency. The rest of the paper is organized as follows: Some preliminaries are given in Section 2. Next, a privacy-protection system towards multi-server architecture 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.

1.1 Multi-server Architecture

In the multi-server environment [15], each user must perform authentication procedure to login the server for a transaction. If the user is in a single authentication architecture, then the user must register at various servers and memorize the corresponding identifications and passwords, which could not be convenient for a user. In order to make the registration to various servers easier for users, each user must register with the registration center to obtain a secure account. Then the user uses the secure account to perform the login and authentication procedures with various servers.

1.2 Security Requirements

Secure communication schemes for remote one-way authentication and session key agreement for the multi-server architecture should provide security requirements [20, 27]:

1) Authentication: Anonymous authentication or hiding identity authentication in different phase in our protocol. Anonymous authentication: the server or a expert knows that he serves for a premium user but does not know the user's identity. Hiding identity authentication: only the *RC* and the server know the user's identity.

2) Impersonation attack: An impersonation attack is an attack in which an adversary successfully assumes the identity of one of the legitimate parties in a system or in a communications protocol.

3) Man-in-the-middle attack: The man-in-the-middle attack is a form of active eavesdropping in which the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker.

4) Replay attack: A replay attack is a form of network attack in which a valid data transmission is repeated or delayed maliciously or fraudulently.

5) Known-key security: Known-key security is that a protocol can protect the subsequent session keys from disclosing even if the previous session keys are revealed by the intendant user.

Table 1: Notations

Symbol	Definition
SID_A	A temporary session;
S_i, ID_{S_i}	The i^{th} server and the identity of the i^{th} server, respectively;
Ano_{S_i}	The identifier of anonymity;
a, r_a, r_i	Nonces;
$(x, T_k(x))$	Public key based on Chebyshev chaotic maps;
k	Secret key based on Chebyshev chaotic maps;
RC, ID_{RC}	Registration center and its identity;
H	A secure one-way hash function;
$\|\|$	Concatenation operation.

6) Perfect forward secrecy: An authenticated key establishment protocol provides perfect forward secrecy if the compromise of both of the node secret keys cannot results in the compromise of previously established session keys.

7) Session key security: A communication protocol exhibits session key security if the session key cannot be obtained without any long-term secrets.

8) Resistance to stolen-verifier attacks: An adversary gets the verifier table from servers or RC by a hacking way, and then the adversary can launch any other attack which called stolen-verifier attacks.

9) No verification table: there is no verification table at the RC or the server at all.

10) Securely chosen password and time synchronization: Guarantee securely chosen password and no need for time synchronization among parties.

1.3 Kinds of Authentication

Anonymity ensures that a user may use a resource or service without disclosing the user identity completely.

ID hiding usually means that a user may use a resource or service without disclosing the user identity during the protocol interaction, which is a kind of privacy protection partly. A pseudonym is an identifier of a subject other than one of the subject real names. ID hiding usually uses pseudonym to realize. Because the server may store the user identity.

OTP (one-time password) usually means that the password can be used only once but the ID is plaintext during the protocol interaction, so there is no privacy protection.

The above-mentioned terms related with authentication called anonymous authentication, hiding identity authentication and OTP authentication.

2 The Proposed Privacy-Protection System with Multi-Server Architecture

In this section, under the multi-server architecture, a chaotic maps-based one-way authentication key agreement scheme is proposed which consists of five phases: server registration phase, user registration phase, Anonymous authenticated key agreement phase, Hiding identity authenticated key agreement phase, Password changing phase.

2.1 Notations and Chebyshev Chaotic Maps

In this section, any server i has its identity ID_{S_i}. Only RC has its identity ID_{RC} and public key $(x, T_k(x))$ and a secret key k based on Chebyshev chaotic maps and a secure one-way hash function $H(\cdot)$. The concrete notations used hereafter are shown in Table 1.

Let n be an integer and let x be a variable with the interval $[-1, 1]$. The Chebyshev polynomial $T_n(x)$: $[-1, 1] \rightarrow [-1, 1]$ is defined as $T_n(x) = \cos(ncos^{-1}(x))$ [24]. Chebyshev polynomial map T_n: $R \rightarrow R$ of degree n is defined using the following recurrent relation:

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

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

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

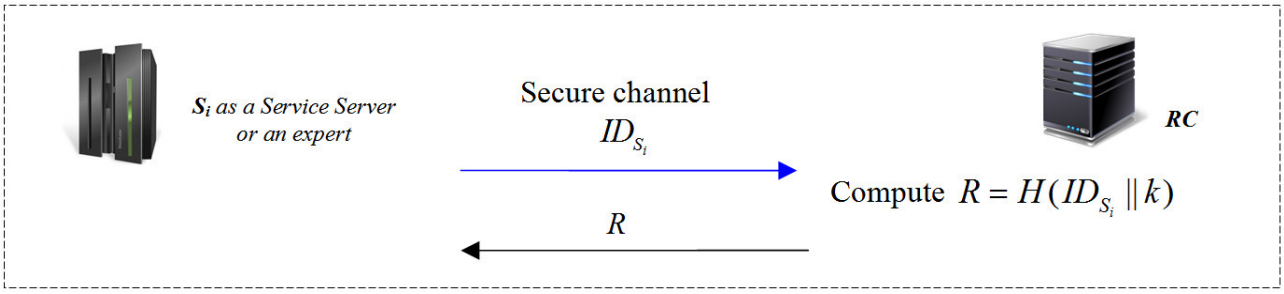

Figure 1: Server or a authenticated expert registration phase

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 [26] 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 of Chebyshev polynomials:*

$$\begin{aligned} T_{rs}(x) &= T_r(T_s(x)) \\ &= \cos(r\cos^{-1}(s\cos^{-1}(x))) \\ &= \cos(rs\cos^{-1}(x)) \\ &= T_s(T_r(x)) \\ &= T_{sr}(x). \end{aligned}$$

Definition 2. *Given x and y, it is intractable to find the integer s,such that $T_s(x) = y$. It is called the Chaotic Maps-Based Discrete Logarithm problem (CMBDLP).*

Definition 3. *Given x, $T_r(x)$ and $T_s(x)$, it is intractable to find $T_{rs}(x)$. It is called the Chaotic Maps-Based Diffie-Hellman problem (CMBDHP).*

2.2 Server Registration Phase

The business architecture of our proposed protocol: (1) The *RC* is a platform for users and servers/experts. In other words, anyone can register at the *RC* as a user or an expert. (2) If a user wants to consult with an expert, the *RC* must help him find an authenticated expert and charge fee. After ending the consultation, the user will give the evaluation for the expert, and the *RC* and the expert will share the fee in some percentage. (3) The expert must be authenticated by real name. (4) The user can consult with an expert anonymously or not. (5) Accumulative assessment will affect the expert's reputation.

Concerning the fact that the proposed scheme mainly relies on the design of Chebyshev chaotic maps-based in multi-server architecture, it is assumed that the servers can register at the registration center in some secure way or by secure channel. The same assumption can be set up for servers. Figure 1 illustrates the server registration phase.

Step 1. When a server(or an expert) wants to be a new legal service provider, she chooses her identity ID_{S_i} with her identification card in law. Then the server submits ID_{S_i} to the *RC* via a secure channel.

Step 2. Upon receiving ID_{S_i} from the server, the *RC* computes $R = H(ID_{S_i}||k)$, where k is the secret key of *RC*. Then the server stores R in a secure way via a secure channel.

2.3 User Registration Phase

Figure 2 illustrates the user registration phase.

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

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

2.4 Anonymous Authenticated Key Agreement Phase

In this phase, the anonymous authentication has three meanings: (1) The server and the *RC* authenticated each other; (2) The *RC* will help the server to authenticate the premium user, but no one knows (including the server and the *RC*) the premium user's identity. (3) The *RC* will help the premium user to authenticate the server. This concrete process is presented in Figure 3.

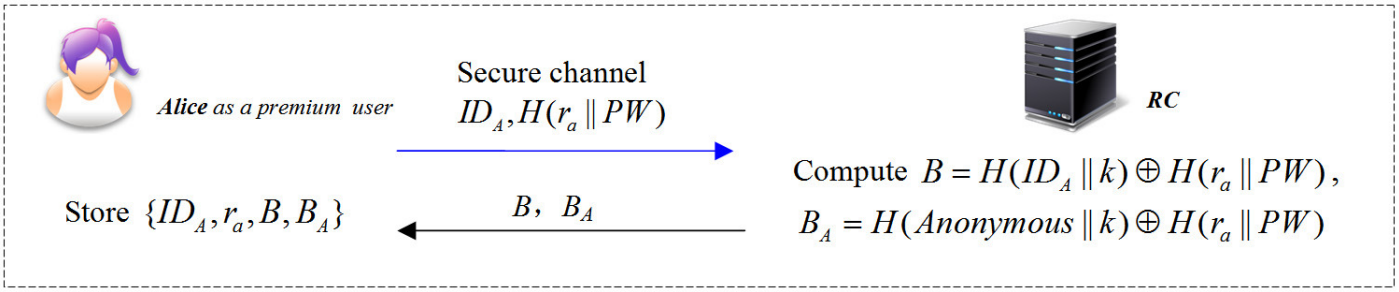

Figure 2: The user registration phase

Step 1. If Alice(assume Alice as a premium user) wishes to consult some personal issues establish with S_i (or an expert) in a anonymous way, she will input password and compute $B_A^* = B_A \oplus H(r_a||PW)$, n choose a random integer number a and compute $K_{A-RC} = T_a T_k(x)$, $H_A = H(B_A^*||ID_{S_i}||T_a(x))$. After that, Alice sends $m_1 = \{Ano_{S_i}, T_a(x), H_A\}$ to S_i where she wants to get the server's service.

Step 2. After receiving the message $m_1 = \{Ano_{S_i}, T_a(x), H_A\}$ from Alice, S_i will do the following tasks to ask RC for helping Alice to authenticate itself: S_i selects random r_i and computes $T_{r_i}(x)$ and $C_1 = H(ID_{S_i}||m_1||R||T_{r_i}(x))$. And then sends the message m_2 to RC.

Step 3. Next, RC will help Alice to authenticate S_i and verify the temporary information by helping them to compute the session key. After receiving the message $m_2 = \{ID_{S_i}, T_{r_i}(x), C_2, m_1\}$, RC will do the following tasks:

1) Authenticate S_i; Based on ID_{S_i}, RC can compute $R^{'} = H(ID_{S_i}||k)$. Then RC computes $C_1^{'} = H(ID_{S_i}||m_1||R^{'}||T_{r_i}(x))$ and check if $C_1^{'}? = C_1$. If above equation holds, that means S_i is legal participant in this instance because only S_i owns R.
2) Anonymous authenticate Alice: RC computes $B_A^* = H(Anonymous||k)$, $H_A^{'} = H(B_A^*||ID_{S_i}||T_a(x))$ and verifies if $H_A^{'} = H_A$ holds. If above equation holds, that means Alice is a legal premium user in this instance because only a legal premium user can retrieve the information $H(Anonymous||k)$.
3) Confirm S_i is the server that Alice wants to consult with: RC computes computes $H_A^{'} = H(B_A^*||ID_{S_i}||T_a(x))$. RC verifies $H_A^{'}? = H_A$. If holds, that means S_i is the server that Alice wants to consult with.
4) Help S_i and Alice to get the session key: RC computes $C_2 = H(ID_{RC}||ID_{S_i}||m_1||R||T_{r_i}(x))$ and $C_3 = H(B_A^*||ID_{S_i}||ID_{RC}||T_{r_i}(x))$. Then RC sends the message $\{ID_{RC}, C_3\}$ to Alice and sends the message $\{ID_{RC}, C_2\}$ to S_i.

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

Step 4. For Alice: After receiving the message $\{ID_{RC}, C_3\}$, Alice computes $C_3^{'} = H(B_A^*||ID_{S_i}||ID_{RC}||T_{r_i}(x))$. Check if $C_3^{'} = C_3$. If holds, Alice computes $SK = T_a T_{r_i}(x)$.

For S_i: After receiving the message $\{ID_{RC}, C_2\}$, S_i computes $C_2^{'} = H(ID_{RC}||ID_{S_i}||m_1||R||T_{r_i}(x))$ and checks if $C_2^{'} = C_2$. If holds, then S_i computes $SK = T_{r_i} T_a(x)$.

Remark 1: We can view the servers and the RC as an integrated system for the user, so from the perspective of the user, we adopt anonymous authentication, that means only user authenticated the integrated system (the server and the RC) but there is a anonymous authentication for the user. However, from the inside integrated system, for providing the reliable service in multi-server architecture, and we must make the server and the RC to authenticate each other, that is the mutual authentication.

2.5 Hiding Identity Authenticated Key Agreement Phase

Simply speaking, a premium user also can as a legal and hiding ID to interact with an expert. The two differences between hiding identity authenticated and anonymous authenticated are:

1) The user uses the B to login at the RC so that the server or the expert can know the users positive identity.
2) We construct an efficient method to covered identity or some important information instead of using symmetric cryptography. Without loss of generality, we assume Party i sends a covered message to Party j using $(x, T_{K_j}(x))$ for covering ID_i but only Party j can recover the ID_i. Party i selects a large and random integer t, and computes $T_t(x)$, $C_t = T_t T_{K_j}(x) ID_i$, $H(C_t||T_t(x))$.

   Then Party i sends $\{T_t(x), C_t, H(C_t||T_t(x))\}$ to Party j. After receiving the message $\{T_t(x), C_t, H(C_t||T_t(x))\}$ from Party i, Party j will use $T_t(x)$

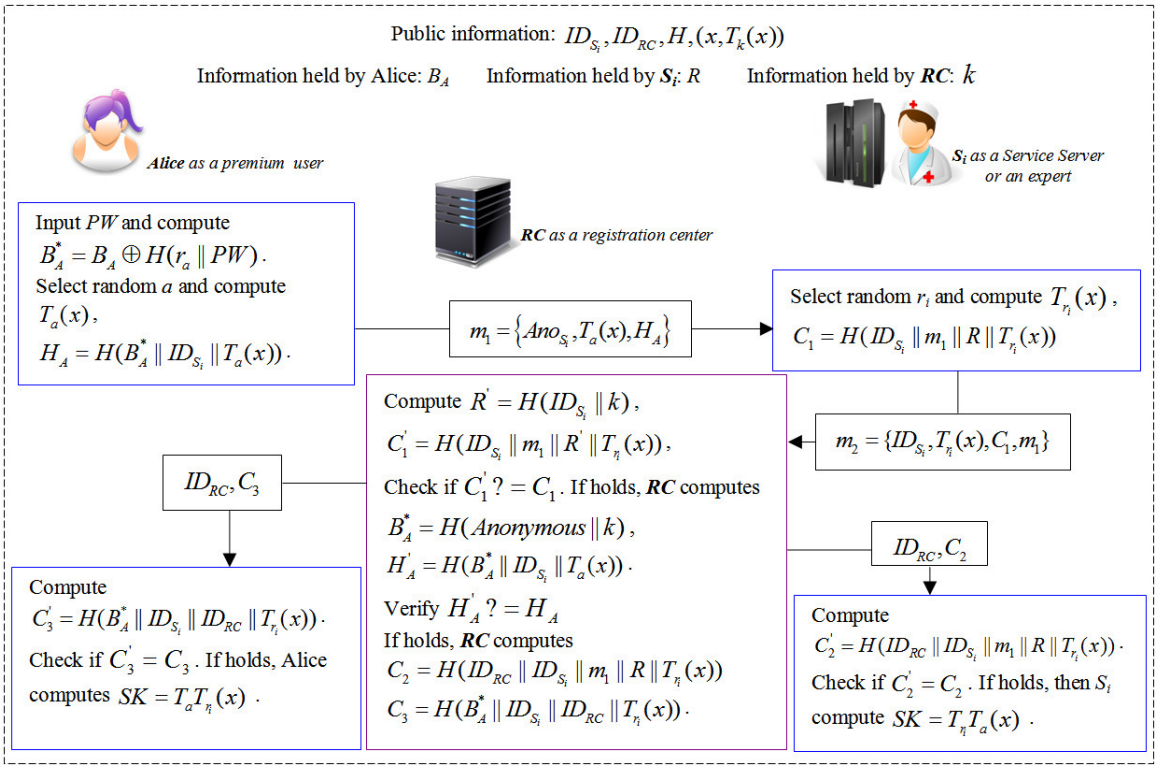

Figure 3: Anonymous authenticated key agreement phase for multi-server environment

and his own secret key K_j to recover $ID_i = C_t/T_{K_j}T_t(x) = C_t/T_tT_{K_j}(x)$. Then Party j check if the two hash values are equal. If above equation holds, Party j deems ID_i is legal identity. Otherwise, Party j terminates the session.

For the sake of simplicity, the paper only provides the process of hiding identity authenticated key agreement phase (Figure 4).

2.6 Password Changing Phase

Figure 5 illustrates the password changing phase.

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

Step 2. Upon receiving $m_1 = \{T_a(x), C_1, C_2, H_A\}$ from Alice, RC computes $K_{RC-A} = T_kT_a(x)$ and recovers $ID_A = C_1/K_{RC-A}$, $H(r_a^{'}||PW^{'}) = C_2/K_{RC-A}$. Next, RC computes $B^* = H(ID_A||k)$ and $H_A^{'} = H(B^*||ID_{RC}||T_a(x)||C_1||C_2)$. Then, RC checks $H_A^{'} = H_A$ or not. If holds, RC computes

$$\begin{aligned} B^{'} &= H(ID_A||k) \oplus H(r_a^{'}||PW^{'}), \\ B_A^{'} &= H(Anonymous||k) \oplus H(r_a^{'}||PW^{'}), \\ H_{RC} &= (ID_{RC}||ID_A||B^{'}||B_A^{'}), \\ C_3 &= B^{'} \times K_{RC-A}, \\ C_4 &= B_A^{'} \times K_{RC-A}, \end{aligned}$$

where k is the secret key of RC. Finally RC sends $\{ID_{RC}, C_3, C_4, H_{RC}\}$ to Alice.

Step 3. Upon receiving $\{ID_{RC}, C_3, C_4, H_{RC}\}$, Alice uses K_{A-RC} to decrypt C_3, C_4 to get $B^{'}, B_A^{'}$. Then Alice computes locally $H_{RC}^{'} = (ID_{RC}||ID_A||B^{'}||B_A^{'})$ to compare with H_{RC}. If they are equal, Alice stores $\{ID_A, r_a^{'}, B^{'}, B_A^{'}\}$ in a secure way.

3 Security Analysis

The section analyzes the security of our proposed protocol. Let us assume that there are three secure components, including the two problems CMBDLP and CMBDHP cannot be solved in polynomial-time and a secure one-way hash function. Assume that the adversary has full control over the insecure channel including eavesdropping, recording, intercepting, modifying the transmitted messages.

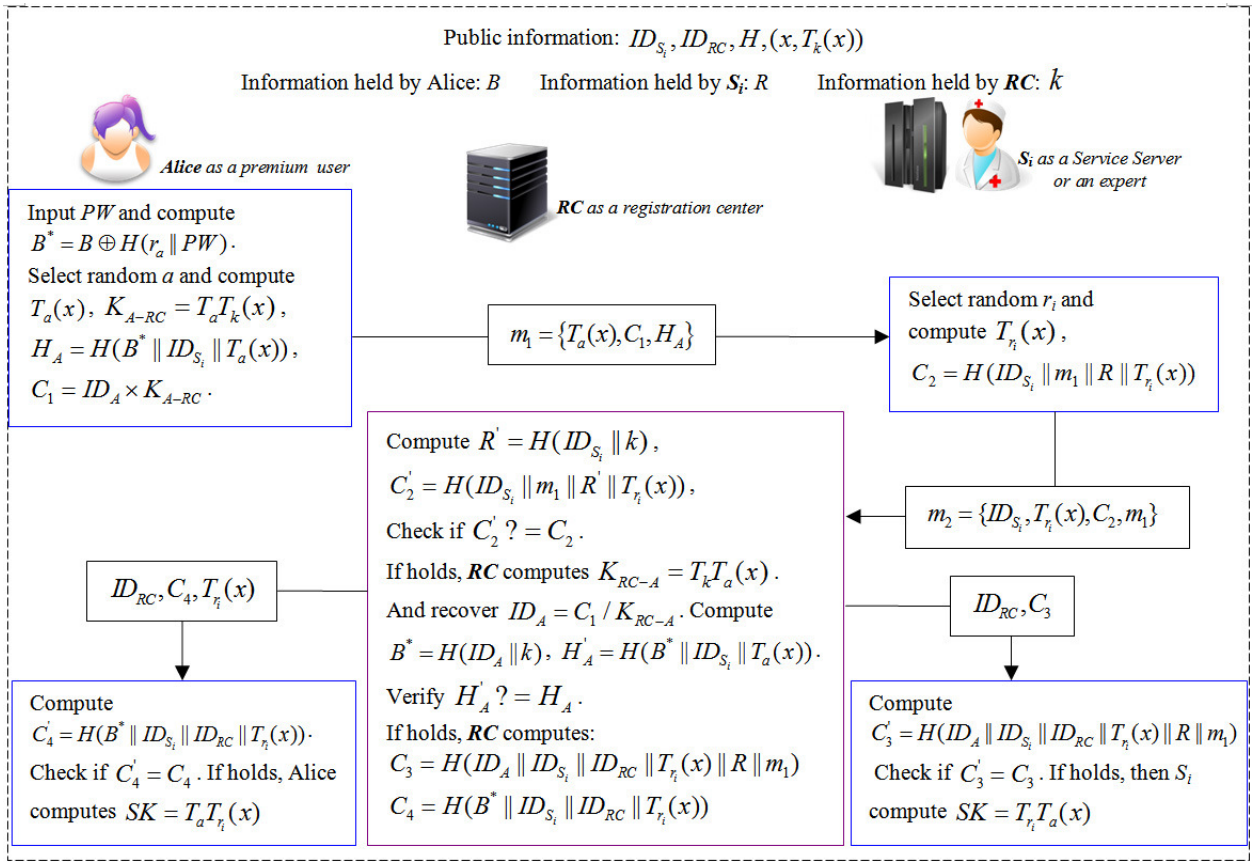

Figure 4: Hiding identity authenticated key agreement phase for multi-server environment

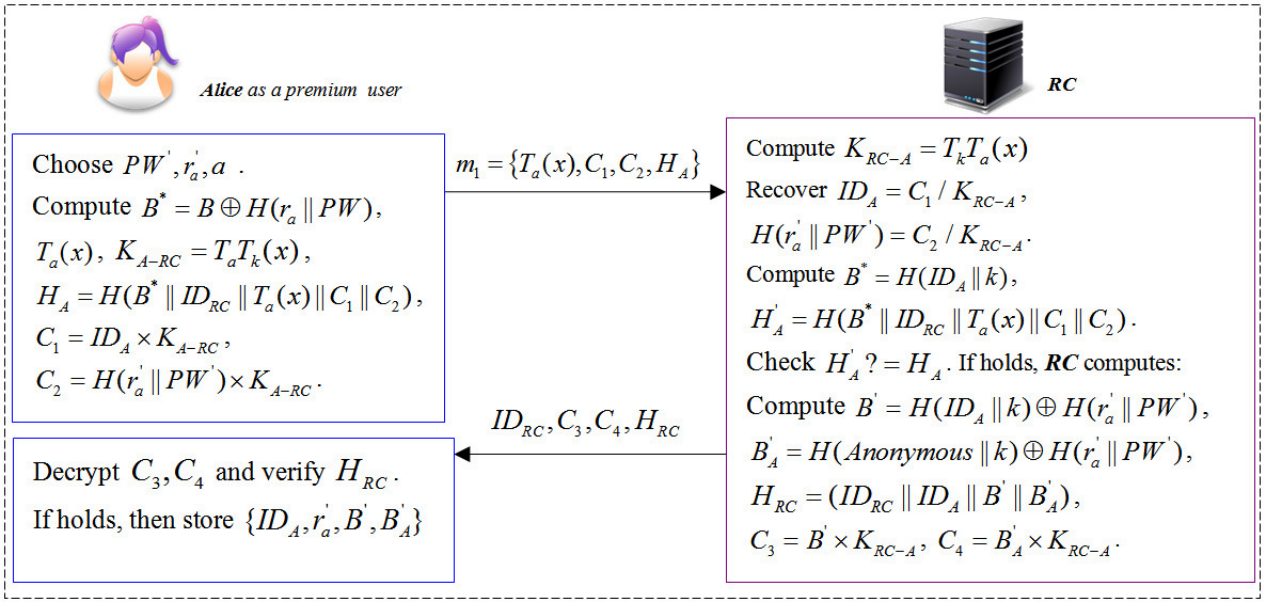

Figure 5: The password changing phase

3.1 Security Proof of the Proposed Scheme

In this subsection, we give a definition and simplified proof of various kinds of security and attacks.

Anonymous Authentication and Key Agreement.

Definition: Anonymous authentication and key agreement refers to authenticate each other for two peers/system, but only one peer knows the other peer's identity and getting the session key simultaneously.

Simplified Proof: Alice authenticates RC: Because only RC has the secret k, RC can computes $K_{RC-A} = T_k T_a(x)$ which equals to $K_{A-RC} = T_a T_k(x)$. So if Alice computes C_3 and check if $C_3' = C_3$. RC and S_i authenticate each other: We can use the shared key R to achieve the task. Firstly, based on ID_{S_i}, RC can compute $R' = H(ID_{S_i}||k)$ by its private key k. Then RC computes $C_1' = H(ID_{S_i}||m_1||R'||T_{r_i}(x))$ and checks if $C_1' = C_1$. If above equation is equal, then that means RC authenticates S_i. After receiving the messages $\{ID_{RC}, C_2\}$, S_i computes $C_2' = H(ID_{RC}||ID_{S_i}||m_1||R||T_{r_i}(x))$ and checks if $C_2' = C_2$. As for the key agreement, after authenticating each other, the temporary $T_a(x)$, $T_{r_i}(x)$ and the $SID_A||ID_{S_i}||ID_{RC}$ were already authenticated by RC. So finally Alice and S_i can make the key agreement simultaneously. The hiding identity authenticated key agreement can be proof in some analogous way.

Impersonation Attack.

Definition: An impersonation attack is an attack in which an adversary successfully assumes the identity of one of the legitimate parties in a system or in a communications protocol.

Simplified Proof: An adversary cannot impersonate anyone of the S_i and RC. The proposed scheme has already authenticated each other between S_i and RC, and Alice authenticates S_i and RC based on the secrets k, R and the nonces a, r_i. So there is no way for an adversary to have a chance to carry out impersonation attack.

Man-in-the-middle Attack.

Definition: The man-in-the-middle attack is a form of active eavesdropping in which the attacker makes independent connections with the victims and relays messages between them, making them believe that they are talking directly to each other over a private connection, when in fact the entire conversation is controlled by the attacker.

Simplified Proof: Because $C_i (1 \leq i \leq 3)$ contain the participants' identities or an anonymous identifier, a man-in-the-middle attack cannot succeed.

Replay Attack.

Definition: A replay attack is a form of network attack in which a valid data transmission is repeated or delayed maliciously or fraudulently.

Simplified Proof: That any message of Alice was replayed by an adversary is meaningless. Because "Alice" is an anonymous user, the adversary can as an anonymous user to initiate the protocol legally as his wish. Furthermore, if the adversary wants to launch the replay attack successfully, it must compute and modify $T_a(x)$, $T_{r_i}(x)$ and $C_i (1 \leq i \leq 3)$ correctly which is impossible.

Known-key Security.

Definition: Known-key security is that a protocol can protect the subsequent session keys from disclosing even if the previous session keys are revealed by the intendant user.

Simplified Proof: Since the session key is depended on the random nonces a and r_i, and the generation of nonces is independent in all sessions, an adversary cannot compute the previous and the future session keys when the adversary knows one session key.

Perfect Forward Secrecy.

Definition: An authenticated key establishment protocol provides perfect forward secrecy if the compromise of both of the node's secret keys cannot results in the compromise of previously established session keys.

Simplified Proof: In the proposed scheme, the session key is related with a and r_i, which were randomly chosen by Alice and the server S_i respectively. So any session key has not related with the secret key (such as k) of each of participants. Furthermore, because of the intractability of the CMBDLP and CMBDHP problem, an adversary cannot compute the previously established session keys.

Session Key Security.

Definition: A communication protocol exhibits session key security if the session key cannot be obtained without any long-term secrets.

Simplified Proof: In the authenticated key agreement phase, a session key SK is generated from a and r_i. These parameter values are different in each session, and each of them is only known by Alice and S_i. Additionally, since the values

Table 2: Security of our proposed protocol

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12
[5] (2013)	Yes	S21	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes
[22] (2008)	Yes	S21	Yes	Yes	Yes	No	No	No	No	No	Yes	No
[16] (2009)	Yes	No	Yes	Yes	Yes	S61	No	No	No	Yes	Yes	No
[7] (2009)	Yes	No	Yes	No	Yes	S61	No	No	No	Yes	Yes	No
Our Scheme	Yes	S22	Yes	Yes	Yes	S62	Yes	Yes	Yes	Yes	Yes	Yes

S1: Single registration; S2: Authentication; S21: Mutual Authentication; S22: Privacy-protection system; S3: No verification table; S4: Securely chosen password; S5: Session key agreement; S6: Privacy protection for a user; S61: ID hiding; S62: Anonymity or ID hiding; S7: Freedom from time synchronization; S8: Session key secrecy; S9: Perfect forward secrecy; S10: Resistance to replay attack; S11: Resistance to stolen-verifier attack; S12: Resistance to masquerading attack Yes/No: Support/Not support the security.

Table 3: Descriptions the model of Canetti and Krawczyk

Symbol	Definition
Parties $P_1, \cdots, P_n$	Modelled by probabilistic Turing machines.
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.
Send query	The adversary can control over Parties' outgoing messages via the *Send* query. Parties can be activated by the adversary launching *Send* queries.
Two sessions matching	If the outgoing messages of one are the incoming messages of the other

a and r_i of the random elements are very large, attackers cannot directly guess the values a and r_i of the random elements to generate session key. Therefore, the proposed scheme provides session key security.

Resistance to Stolen-verifier Attacks.

Definition: An adversary gets the verifier table from servers or RC by a hacking way, and then the adversary can launch any other attack which called stolen-verifier attacks.

Simplified Proof: In the proposed scheme, neither the server nor the registration center maintains any verification table. Thus, the stolen-verifier attack is impossible to initiate in the proposed scheme.

From Table 2, 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.

Remark 2: Some qualitatively discuss about the difference between the proposed scheme and [5, 7, 16, 22] as followed: (1) Our protocol is one way authentication AKE for users, so only servers need to registration at the RC. (2) About authentication, one-way authentication for users and mutual authentication for server and RC (see Remark 1). (3) Our proposed protocol can hold the security S1-S12, but the [5, 7, 16, 22] have some defects. (4) Our protocol is anonymity, and [7, 16] only assure ID hiding, and [5, 22] have no privacy protect at all.

3.2 The Provable Security of the Proposed Scheme

We recall the definition of session-key security in the authenticated-links adversarial model of Canetti and Krawczyk [4]. The basic descriptions are shown in Table 3.

We allow the adversary access to the queries **SessionStateReveal, SessionKeyReveal**, and **Corrupt**.

1) SessionStateReveal(s): This query allows the adversary to obtain the contents of the session state, including any secret information. s means no further output.

2) SessionKeyReveal(s): This query enables the adversary to obtain the session key for the specified session s, so long as s holds a session key.

3) Corrupt(P_i):This query allows the adversary to take over Party P_i, including long-lived keys and any session-specific information in P_i's memory. A corrupted party produces no further output.

4) Test(s): This query allows the adversary to be issued at any stage to a completed, fresh, unexpired session s. 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 can then continue to issue queries as desired, with the exception that it cannot expose the test session. At any point, the adversary can try to guess b. Let $GoodGuess^{\Lambda}(k)$ be the event that the adversary Λ correctly guesses b, and we define the advantage of adversary Λ as $Advantage^{\Lambda}(k) = \max\{0, |\Pr[GoodGuess^{\Lambda}(k)] - \frac{1}{2}|\}$, where k is a security parameter.

A session s is locally exposed with P_i:if the adversary has issued SessionStateReveal(s), SessionKeyReveal(s), Corrupt(P_i) before s is expired.

Definition 4. *An authenticator exchange protocol Π_1 in security parameter k is said to be authentication secure in the adversarial model of Canetti and Krawczyk if for any polynomial-time adversary Λ,*

1) If two uncorrupted parties have completed matching sessions, these sessions produce the same key as output;

2) $Advantage^{\Lambda}(k)$ is negligible.

Theorem 1. *Under the CMBDHP assumption, using Algorithm 1 to compute two authenticator messages can be deemed as session keys which are session-key secure in the adversarial model of Canetti and Krawczyk [4].*

Proof. The proof is based on the proof given by [4]. There are two-two uncorrupted parties (Alice and the server, Bob and the server) in matching sessions output the same authenticator messages, and thus the first part of Definition 4. is satisfied. To show that the second part of the definition is satisfied, assume that there is a polynomial-time adversary Λ with a non-negligible advantage ε in standard model. We claim that Algorithm 1 forms a polynomial-time distinguisher for CMBDHP having non-negligible advantage. □

Probability analysis. It is clear that Algorithm 1 runs in polynomial time and has non-negligible advantage. There are two cases where the r^{th} session is chosen by Λ as the test session: (1) If the r^{th} session is not the test session, then Algorithm 1 outputs a random bit, and thus its advantage in solving the CMBDHP is 0. (2) If the r^{th} session is the test session, then Λ will succeed with advantage ε, since the simulated protocol provided to Λ is indistinguishable from the real protocol. The latter case occurs with probability $1/k$, so the overall advantage of the CMBDHP distinguisher is ε/k, which is non-negligible.

Definition 5. *A composable key exchange protocol Π_2 in security parameter k is said to be session-key secure in the adversarial model of Canetti and Krawczyk if for any polynomial-time adversary Λ,*

1) If two uncorrupted parties have completed matching sessions with pre-distributed parameter, these sessions produce the same key as output;

2) $Advantage^{\Lambda}(k)$ is negligible.

Theorem 2. *Under the CMBDHP assumption, using Algorithm 2 to compute session key is session-key secure in the adversarial model of Canetti and Krawczyk [4].*

Proof. The proof's process is similar to Theorem 1. The protocol Π_2 is the composable instance of protocol multiple Π_1. Since Theorem 1 is session-key secure, the protocol Π_2 is also session-key secure. □

Probability analysis. It is similar to Algorithm 1. If we assume that Algorithm 2 forms a polynomial-time distinguisher for CMBDHP having non-negligible advantage, the overall advantage of the proposed protocol simulator with authenticated parameter is ε/k which is also non-negligible. Because the protocol Π_2 chooses different parameters to structure session keys in different phase which are secure independence of protocol Π_1.

4 Efficiency Analysis

Compared to RSA and ECC, Chebyshev polynomial computation problem offers smaller key sizes, faster computation, as well as memory, energy and bandwidth savings. In our proposed protocol, no time-consuming modular exponentiation and scalar multiplication on elliptic curves are needed. However, Wang [24] 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 [12].

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 [12]. Moreover, the computational cost of XOR operation could be ignored when compared with other operations.

Table 4 shows performance comparisons between our proposed scheme and the literature of [5, 7, 16, 22] in

Algorithm 1 CMBDHP distinguisher

Input: $H, E_K()/D_K(), (x, T_k(x))$

1: $r \xleftarrow{R} \{1,...,k\}$, where k is an upper bound on the number of sessions activated by Λ in any interaction.
2: Invoke Λ and simulate the protocol to Λ, except for the $r-th$ activated protocol session.
3: For the $r-th$ session, let a user send $\{i, Ano_{S_i}, T_a(x), H_A\}$ to a server $\boldsymbol{S_i}$, and let the server $\boldsymbol{S_i}$ send $\{i, ID_{S_i}, T_{r_i}(x), C_1, m_1\}$ to the ***RC***, where i is the session identifier. The ***RC*** can compute the encrypted messages $\{C_2, C_3\}$ with the authenticators locally after authenticating the server $\boldsymbol{S_i}$ by one-round messages and public information.
4: **if** the $r-th$ session is chosen by Λ as the test session **then**
5: Provide Λ as the answer to the test query.
6: $d \leftarrow \Lambda$'s output.
7:**else** $d \xleftarrow{R} \{0,1\}$.
8:**end if**
Output: d

Algorithm 2 Proposed protocol simulator

Input: $H, (x, T_a(x)), (x, T_{r_i}(x), (x, T_k(x))$

1: $r \xleftarrow{R} \{1,...,k\}$, where k is an upper bound on the number of sessions activated by Λ in any interaction.
2: Invoke Λ and simulate the protocol to Λ, except for the $r-th$ activated protocol session.
3: For the $r-th$ session, After running the protocol Π_1, the ***RC*** can compute the encrypted messages $\{C_2, C_3\}$ with the authenticators locally. Then the ***RC*** continues to send messages $\{ID_{RC}, C_3\}$ and $\{ID_{RC}, C_2\}$ to the user and the server $\boldsymbol{S_i}$ respectively. Both the user and the server can compute the session key $SK = T_a T_{r_i}(x)$ locally after authenticating each other by ***RC***'s messages and public information.
4: **if** the $r-th$ session is chosen by Λ as the test session **then**
5: Provide Λ as the answer to the test query.
6: $d \leftarrow \Lambda$'s output.
7:**else** $d \xleftarrow{R} \{0,1\}$.
8:**end if**
Output: d

Table 4: Efficiency of our proposed scheme

Phase	[5] (2013)	[22] (2008)	[16] (2009)	[7] (2009)	Ours
A	$2T_{hash} + 1T_{XOR}$	$2T_{hash} + 1T_{XOR}$	$5T_{hash} + 2T_{XOR}$	$8T_{hash} + 4T_{XOR}$	$3T_{hash}$
B	$1T_{hash}$	$1T_{hash}$	$1T_{hash}$	$1T_{hash}$	$1T_{hash}$
C	$2T_{hash} + 1T_{XOR} + 1T_{Exp}$	$1T_{hash} + 2T_{XOR}$	$6T_{hash} + 3T_{XOR}$	$7T_{hash} + 7T_{XOR}$	N/A
D1-User	$1T_{hash} + 1T_{Exp}$	$4T_{hash} + 3T_{XOR}$	$3T_{hash}$	$2T_{hash}$	$3T_{hash} + 1T_{CH}$
D1-Server	$2T_{hash} + 2T_{Exp}$	$6T_{hash} + 7T_{XOR}$	$6T_{hash} + 3T_{XOR}$	$8T_{hash} + 6T_{XOR}$	$2T_{hash} + 1T_{CH}$
D1-RC	$6T_{hash}$	$6T_{hash} + 5T_{XOR}$	0	$5T_{hash} + 7T_{XOR}$	$6T_{hash} + 2T_{CH}$
D1-Total	$9T_{hash} + 3T_{Exp}$	$16T_{hash} + 15T_{XOR}$	$9T_{hash} + 3T_{XOR}$	$15T_{hash} + 13T_{XOR}$	$11T_{hash} + 4T_{CH}$
D2-User	N/A	N/A	N/A	N/A	$3T_{hash} + 1T_{CH}$
D2-Server	N/A	N/A	N/A	N/A	$2T_{hash} + 1T_{CH}$
D2-RC	N/A	N/A	N/A	N/A	$6T_{hash} + 2T_{CH}$
D2-Total	N/A	N/A	N/A	N/A	$11T_{hash} + 4T_{CH}$
E	$2T_{hash} + 2T_{XOR}$	$2T_{hash} + 2T_{XOR}$	$4T_{hash} + 5T_{XOR}$	$4T_{hash} + 4T_{XOR}$	$8T_{hash} + 2T_{CH}$
F	4 rounds	7 rounds	3 rounds	5 rounds	3 rounds

A: User registration; B: Server registration; C: Login phase; D1: Hiding identity authentication phase; D2: Anonymous authentication phase; E: Password change phase; F: Communication cost; N/A: No support.

multi-server architecture. Therefore, as in Table 4 the concrete comparison data as follows: The total computation cost of our proposed protocol is lower than the literatures [5]. The main reason is that the literatures [5] adopted modular exponentiation computation. At the same time, the literatures [5] cannot provide privacy protection for a user. The total computation cost of our proposed protocol is higher than the literatures [7, 16, 22]. Furthermore, the communication round of our proposed protocol is superior to the literature [7, 22] and is equal to the literature [16]. The reasons are: one reason is our protocol mainly adopts Chebyshev chaotic maps but the literatures [7, 16, 22] mainly adopts one way hash function. At the same time, Chebyshev chaotic maps has more attributes which leading to reduce communication rounds. Furthermore, from the perspective of security, our protocol is more secure than the literatures [7, 16, 22]. From Table 2, we can see that the literatures [5, 7, 16, 22] cannot resist many attacks and the literatures [7, 16] cannot afford any authentication method. Therefore, as in Table 2 and Table 4, we can draw a conclusion that the proposed scheme has achieved the balance of efficiency and security.

5 Conclusion

We only use chaotic maps and a secure one-way hash function to construct a provable privacy-protection system (PPS) which provides a provable privacy-protection system towards multi-server architecture. The core ideas of the proposed system are the mutual authentication between the servers and *RC* and the anonymity or hiding identity for the users. Subsequently, we explain the practical motivations for authentication and secrecy assurances of parties engaging in AKE protocols and some related terms. Based on our discussion we proposed a suitable protocol that covers those goals and offered an efficient protocol that formally meets the proposed security definition. Finally, after comparing with related literatures (multi-server schemes and privacy-protection system) respectively, we found our proposed scheme has satisfactory security, efficiency and functionality. Therefore, our protocol is more suitable for practical applications.

References

[1] K. Aniket, G. M. Zaverucha, and G. Ian, "Pairing-based onion routing with improved forward secrecy," *ACM Transactions on Information and System Security*, vol. 13, no. 4, pp. 29, 2010.

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

[3] E. Bresson, O. Chevassut and D. Pointcheval, "Group Diffie-Hellman key exchange secure against dictionary attack," in *Advances in Cryptography (Asiacrypt'02)*, LNCS 2501, pp. 497–514, Springer, 2002.

[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] T. Y. Chen, C. C. Lee, M. S. Hwang and J. K. Jan, "Towards secure and efficient user authentication scheme using smart card for multi-server environments," *The Journal of Supercomputing*, vol. 66, no. 2, pp. 1008–1032, 2013.

[6] T. Dierks and A. Christopher, *The TLS Protocol Version 1.0*, RFC 2246, 1999. (`http://www.ietf.org/rfc/rfc2246.txt`)

[7] H. C. Hsiang, W. K. Shih, "Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment," *Computer Standard & Interfaces*, vol. 31, no. 6, pp. 1118–1123, 2009.

[8] G. Ian, "On the security of the Tor authentication protocol," in *Privacy Enhancing Technologies*, LNCS 4258, no. 2, pp. 316–331, Springer, 2006.

[9] G. Ian, D. Stebila, and U. Berkant, "Anonymity and one-way authentication in key exchange protocols," in *Design Codes and Cryptography*, vol. 67, no.5, pp. 245–269, 2013.

[10] J. Katz, P. MacKenzie, G. Taban, V. Gligor, "Two-server password-only authenticated key exchange," in *Applied Cryptography and Network Security*, LNCS 3531, pp. 1-16, Springer, 2005.

[11] M. K. Khan, J. Zhang, "Improving the security of a flexible biometrics remote user authentication scheme," *Computer Standard & Interfaces*, vol. 29, no. 1, pp. 82–85, 2007.

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

[13] 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.

[14] H. Li, C. K. Wu, J. Sun, "A general compiler for password-authenticated group key exchange protocol," *Information Processing Letters*, vol. 110, no. 4, pp. 160–167, 2010.

[15] L. H. Li, I. C. Lin, and M. S. Hwang, "A remote password authentication scheme for multi-server architecture using neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1498–1504, 2001.

[16] Y. P. Liao, S. S. Wang, "A secure dynamic ID based remote user authentication scheme for multiserver environment," in *Computer Standard & Interfaces*, vol. 31, no. 1, pp. 24–29, 2009.

[17] I. C. Lin, M. S. Hwang, and L. H. Li, "A new remote user authentication scheme for multi-server architecture," *Future Generation Computer Systems*, vol. 19, no. 1, pp. 13–22, 2003.

[18] P. Morrissey, N. P. Smart, and B. Warinschi, "A modular security analysis of the TLS handshake protocol," in *Advances in Cryptology*, LNCS 5350, pp. 55–73, Springer, 2008.

[19] NIST, *Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography*, NIST National Institute of Standards and Technologe, pp. 800, 2009.

[20] R. S. Pippal, C. D. Jaidhar, S. Tapaswi, "Robust smart card authentication scheme for multi-server architecture," *Wireless Personal Communications*, vol. 72, no. 1, pp. 729–745, 2013.

[21] M. D. Raimondo, R. Gennaro, "Provably secure threshold password-authenticated key exchange," in *Journal of Computer and System Sciences*, vol. 72, no. 6, pp. 978–1001, 2006.

[22] J. L. Tsai, "Efficient multi-server authentication scheme based on one-way hash function without verification table," *Computers & Security*, vol. 27, no. 3-4, pp. 115–121, 2008.

[23] H. Tseng, R. Jan, W. Yang, "A chaotic maps-based key agreement protocol that preserves user anonymity," in *IEEE International Conference on Communications (ICC'09)*, pp. 1–6, 2009.

[24] 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.

[25] T. Y. Wu, Y. M. Tseng, and T. T. Tsai, "A revocable ID-based authenticated group key exchange protocol with resistant to malicious participants," *Computer Networks*, vol. 56, no. 12, pp. 2994–3006, 2012.

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

[27] H. F. Zhu, "A provable one-way authentication key agreement scheme with user anonymity for multi-server environment," *KSII Transactions on Internet And Information Systems*, vol. 9, no. 2, pp. 811–829, 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 papers (SCI or EI journals) 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 six articles in EI journals.

Yang Sun obtained his master degree in Information Science and Engineering from Northeastern University. Yang Sun is a full associate professor of the Kexin software college at Shenyang Normal University. He is also a department head of network engineering. He has research interests in wireless networks, mobile computing, cloud computing, social networks and network security. Yang Sun had published more than 15 international journal and international conference papers on the above research fields.