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

Three-party Authentication Key Agreement Protocol Based on Chaotic Maps in the Standard Model with Privacy Preserving

Hong-Feng Zhu, Hui-Yan Liu, Yi-Feng Zhang and Yan Zhang

Software College, Shenyang Normal University
No.253, HuangHe Bei Street, HuangGu District, Shenyang, P.C 110034-China
zhuhongfeng1978@163.com; {64356340; 1548452125; 1505733680}@qq.com

Received February, 2015; revised June, 2015

Abstract. *Nowadays, several three-party authenticated key agreement protocols based on Chebyshev chaotic maps have been proposed. Most of them can provide heuristic security, which means that once the weaknesses of these protocols are found, they are either modified or abandoned. Under this circumstance, some protocols which defined in the standard security models have been proposed. These protocols establish a session key to authenticate each other with the help of a trusted server. Usually, users share their passwords and identities with the trusted server in these protocols. Users cannot protect their privacy information. In our paper, we propose a novel authentication key agreement protocol with user anonymity in the standard model, through applying public key encryption based on Chebyshev chaotic maps and pseudorandom function. In the design of our paper, we follow the ideas in the protocol of Lai et al. The proposed protocol not only can achieve various securities, but also can provide user anonymity.*
Keywords: Chaotic maps, Standard model, Pseudorandom function, Privacy Preserving

1. **Introduction.** In the research literature, chaotic system has many distinctive characteristics such as overly sensitivity to initial conditions, unpredictability, boundness, etc; chaotic sequence generated by chaotic system is characterized by non-periodicity and pseudo-randomness. These topping characteristics show excellent properties including diffusion and confusion, which is particularly essential in secret key cryptosystems. There are many protocols used for a signal server environment, however, if a remote user wants various services, it is trouble to repeatedly register new identities and passwords. So, proposed protocols applied to multi-server environment are more practical. In 2006, Alvarez et al. [1] provided a common framework of basic guidelines, which could benefit every new cryptosystem. The suggested guidelines aimed at assisting new cryptosystem designers to present their work in a more systematic and rigorous way to achieve some basic cryptographic requirements. From then on, more and more authors [2-9, 13-16] make their attention on the original and practical key agreement protocols based on Chebyshev chaotic maps. In 2007, based on the semi-group property of Chebyshev chaotic maps and some improvements of their original protocol, Xiao et al. [2] proposed a novel key agreement protocol which was proved to be secure, feasible and extensible. Unfortunately, soon after, [2] was proved to be the existence of faults. In 2008, Han [3] presented two attacks on [2], and proved that [2] could not establish a secure session key for the server and the users. In 2009, Xiang et al. [4] proved that [2] could not resist the stolen-verifier

attack and the offline guessing attack. In the same year, Tseng et al. [5] proposed a novel chaotic maps-based key agreement protocol with user anonymity. They claimed that their proposed protocol could provide mutual authentication between server and users, and allow the user to anonymously interact with the server to establish a shared session key. However, in 2011, Niu et al. [6] pointed out that [5] could not ensure the user anonymity and provide perfect forward secrecy, and then proposed a trusted third party into their protocol designing. Unfortunately, in 2012, Xue et al. [7] pointed out that the protocol of [6] is found to have several unsatisfactory drawbacks, and given some improvements to meet the original security and performance requirements. Meanwhile, [7] also overcame the security flaws of [5]. In recent years, more and more three-party authentication key agreement have been widely proposed and used. In 2012, Yang et al. [8] proposed a provably secure three-party password authenticated key exchange protocol in the standard model. They claimed that their protocol had stronger security and the better security properties such as semantic security, mutual authentication, key privacy, resistance to various known attacks and so on. However, in 2014, according to the ideas in [8], Lai et al. [9] firstly proposed a provably secure three-party key agreement protocol using Chebyshev chaotic maps in the standard model. In the same year, Farash and Attari [13] proposed a chaotic maps-based 3PAKE protocol in the random oracle model with no need for smart cards to login into the server, the servers public key to ensure the identity of it and symmetric cryptosystems to encrypt the messages, their protocol has better performances including communication, computation and security. Hu et al. [15] pointed that the protocol of Lee et al. cannot resist man-in-the-middle attack and provide user anonymity, and proposed an enhanced protocol to overcome the holes and improve the efficiency of it. In 2015, Lee et al. [14] proposed a new chaotic maps-based 3PAKE protocol with privacy protection without using passwords table. In the same year, Li et al. [16] proposed a chaotic maps-based 3PAKE protocol without password and clock synchronization, which can avoid the holes coming from the password-based key agreement. According to the security analysis using Burrows-Abadi-Needham logic and the performance and functionality comparison with other related protocols, the proposed protocol is efficient and practical. They used a public encryption based on enhanced Chebyshev chaotic maps and pseudo-random function ensembles to achieve security properties and the ability against various attacks. In this paper, we propose a novel authentication key agreement protocol with user anonymity on chaotic maps cryptosystem in the standard model. Our main contributions are shown as below: (1) We firstly put forward an authentication key agreement protocol with user anonymity on chaotic maps cryptosystem in the standard model (2) Our scheme can real resist active attacks, passive attacks, even the offline dictionary. (3) Our schemes practicability, stability, security is better than the related papers. Our paper is organized as follows: In the next section, we give the concepts of Chebyshev chaotic maps, pseudo-random function ensembles. Section 3 introduces our protocol in detail. Section 4 describes the standard model in our protocol. Section 5 discusses the security of our protocol in detail. The paper is concluded in section 6.

2. Theoretical concepts.

In this section, we introduce some basic concepts of Chebyshev chaotic maps, pseudo-random function ensembles in detail.

2.1. Chebyshev chaotic maps.

(1) Chebyshev polynomial [10] of degree $n (n \in \mathrm{N})$ is defined as

$$T_n(x) = \cos(narccos(x)), where\ \{x| -1 \leq x \leq 1\} \tag{1}$$

According to (1), the recurrence relation is defined as

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), n \geq 2, where T_0(x) = 1 \; and \; T_1(x) = x \tag{2}$$

(2) Chebyshev polynomial has two properties:
The chaotic property: When $n \geq 1$, Chebyshev polynomial map $T_n(x) : [-1, 1] \to [-1, 1]$ of degree n is a chaotic map with its invariant density $f^*(x) = 1/(\pi\sqrt{1-x^2})$, for positive Lyapunov exponent $\ln n$.
The semi-group property [11]: The semi-group property of Chebyshev polynomial defined on the interval $(-\infty, +\infty)$ holds, as follows:

$$T_n(x) \equiv (2xT_{n-1}(x) - T_{n-2}(x)) \bmod p \tag{3}$$

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

$$T_r(T_s(x)) \equiv T_{rs}(x) \equiv T_s(T_r(x)) \bmod p \tag{4}$$

Besides, the following problems are assumed to be intractable within polynomial time. (3) Chaotic Maps-based Discrete Logarithm problem (CMDLP): Given two variables x and y, it is intractable to find the integer s ,such that $T_s(x) = y$.

2.2. **Pseudo-random function ensembles.** Assume that $F : K \times D \to R$ is an ensemble of functions, A is a probabilistic polynomial time (PPT) algorithm [8, 9]. The PPT algorithm inputs an oracle for a random function $f : D \to R$ and outputs a bit b. If $n(n \in \mathrm{N})$ is large enough, for a probabilistic polynomial oracle λ,we know that

$$Adv^F(\lambda) = \left|\Pr\left[\lambda^{F_n}(1^n) = 1\right] - \Pr\left[\lambda^{H_n}(1^n) = 1\right]\right| < \varepsilon(n) \tag{5}$$

Where $H = \{H_n\}_{n\in\mathrm{N}}$ is a uniformly distributed function ensemble; $\varepsilon(\cdot)$ is a negligible function; $Adv^F = \max_\lambda\{Adv^F(\lambda)\}$ denotes all oracle λ; And $Adv^F(\lambda)$ represents the accessible maximum.

3. The proposed protocol.

In this section, we introduce our protocol in detail. Our protocol is made up of four phases: the initialization phase, user registration phase, authentication key agreement phase, password changing phase, respectively.
We introduce the notations used in the proposed scheme. Notations are shown in **Table 1**.

Table 1. Notations

Notation	Definition
A, ID_A, PW_A	the user A, the identity and password of A, respectively
B, ID_B, PW_B	the user B, the identity and password of B, respectively
S, ID_S	the server, the identity of S, respectively
$(x, T_k(x)), k$	the public key and the secret key of S, respectively
F	pseudo-random function ensembles
$E_K(\cdot), D_K(\cdot)$	secure symmetric encryption/decryption algorithm with key K
$\oplus$, $\|\|$	XOR operation, concatenation operation，respectively

3.1. **Initialization phase.** In this subsection, a server S chooses its public key and secret key $(x, T_k(x)), k$ based on Chebyshev chaotic maps, a secure symmetric encryption/decryption algorithm $E_K(\cdot)/D_K(\cdot)$ with key K. Additionally, the user A chooses his/her identity ID_A and password PW_A, and the user B chooses his/her identity ID_B and password PW_B, respectively.

3.2. **User registration phase.** **Fig.1** shows the user registration phase as below (Taking an example of the user A): (1)A inputs his/her identity and password ID_A, PW_A, computes $M_A = F_{PW_A}(ID_A||PW_A)$,and then chooses a random number a, computes $K_A = T_aT_k(x)$, $C_A = E_{K_A}(M_A)$, and then sends $C_A, T_a(x)$ to server S. (2)S computes $K_A^{'} = T_kT_a(x)$ decrypts C_A by $K_2^{'}$, obtains M_A, and then comput $R_A = F_{M_A}(M_A||k)$, $Z_A = R_A \oplus M_A$, and then sends Z_A to A. (3)A makes the value of Z_A public.

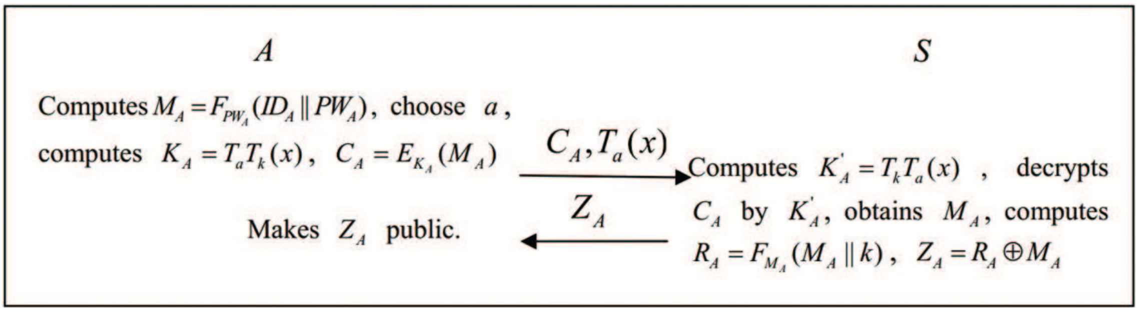

FIGURE 1. User registration phase

3.3. **Authentication key agreement phase.** **Fig.2** shows the authentication key agreement phase as below:
(1)A inputs his/her identity and password ID_A, PW_A, computes $M_A^{'} = F_{PW_A}(ID_A||PW_A)$ and then checks whether $M_A^{'} \stackrel{?}{=} M_A$. If it holds, A chooses a random number a,computes $K_1 = T_aT_k(x)$,$C_1 = E_{K_1}(M_A^{'}, Z_A, Z_B)$,and sends $C_1, T_a(x)$ to server S.
(2)S computes $K_1^{'} = T_kT_a(x)$,decrypts C_1 by $K_1^{'}$, obtains $M_A^{'}, Z_A, Z_B$.
Then S computes $R_A^{'} = F_{M_A^{'}}(M_A^{'}||k)$,and checks whether $R_A^{'} \stackrel{?}{=} Z_A \oplus M_A^{'}$.
If it holds, S sends Z_B to$B$ as a session request.
(3)B inputs his/her identity and password ID_B, PW_B, computes $M_B^{'} = F_{PW_B}(ID_B||PW_B)$ and then checks whether $M_B^{'} \stackrel{?}{=} M_B$. If it holds, B chooses a random number b,computes $K_2 = T_bT_k(x)$, $C_2 = E_{K_2}(M_B^{'}, Z_B)$,and sends $C_2, T_b(x)$ to server S.
(4)S computes $K_2^{'} = T_kT_b(x)$,decrypts C_2 by $K_2^{'}$,obtains $M_B^{'}, Z_B$.Then S computes $R_B^{'} = F_{M_B^{'}}(M_B^{'}||k)$,and checks whether $R_B^{'} \stackrel{?}{=} Z_B \oplus M_B^{'}$.If it holds, S chooses two random numbers t_1, t_2, computes $T_1 = T_{t_1}(x)$, $T_2 = T_{t_2}(x)$, $T_1^* = T_1T_{R_A^{'}}(T_k(x))$, $T_2^* = T_2T_{R_B^{'}}(T_k(x))$,and then sends Z_A, Z_B, ID_S, T_1^*, Z_A, Z_B, ID_S, T_2^* to A and B, respectively.
(5)After receiving Z_A, Z_B, ID_S, T_1^*, A chooses two random numbers x_1, x_2,computes $X_1 = T_{x_1}(x)$, $X_2 = T_{x_2}(x)$, $X^{'} = X_1T_{x_2}(T_k(x))$,and then computes $T_1 = T_1^*/(T_{Z_A \oplus M_A}(T_k(x)))$, $\alpha = T_{x_1}(T_1)$,$\alpha K_{AS} = F_\alpha(Z_A||Z_B||ID_S||X^{'})$.Then A sends $X_2, X^{'}, \alpha K_{AS}$ to S. In the same way, after receiving Z_A, Z_B, ID_S, T_2^*, B chooses two random numbers y_1, y_2,computes $Y_1 = T_{y_1}(x)$, $Y_2 = T_{y_2}(x)$, $Y^{'} = Y_1T_{y_2}(T_k(x))$,and then computes $T_2 = T_2^*/(T_{Z_B \oplus M_B}(T_k(x)))$, $\beta = T_{y_1}(T_2)$, $\beta K_{BS} = F_\beta(Z_A||Z_B||ID_S||Y^{'})$.
Then B sends $Y_2, Y^{'}$
, βK_{BS} to S.

A $\qquad$ S $\qquad$ B

Computes $M_A^{'} = F_{PW_A}(ID_A \| PW_A)$, checks whether $M_A^{'} \overset{?}{=} M_A$. If it holds, chooses a, computes $K_1 = T_a T_k(x)$, $C_1 = E_{K_1}(M_A^{'}, Z_A, Z_B)$

$$\xrightarrow{C_1, T_a(x)}$$

Computes $K_1^{'} = T_k T_a(x)$, decrypts C_1 by $K_1^{'}$, obtains $M_A^{'}, Z_A, Z_B$. Then computes $R_A^{'} = F_{M_A^{'}}(M_A^{'} \| k)$, checks whether $R_A^{'} \overset{?}{=} Z_A \oplus M_A^{'}$. If it holds, sends Z_B to B as a session request.

$$\xrightarrow{Z_B}$$

Computes $M_B^{'} = F_{PW_B}(ID_B \| PW_B)$, checks whether $M_B^{'} \overset{?}{=} M_B$. If it holds, chooses b, computes $K_2 = T_b T_k(x)$, $C_2 = E_{K_2}(M_B^{'}, Z_B)$

$$\xleftarrow{C_2, T_b(x)}$$

Computes $K_2^{'} = T_k T_b(x)$, decrypts C_2 by $K_2^{'}$, obtains $M_B^{'}, Z_B$, computes $R_B^{'} = F_{M_B^{'}}(M_B^{'} \| k)$, checks whether $R_B^{'} \overset{?}{=} Z_B \oplus M_B^{'}$. If it holds, chooses t_1, t_2, computes $T_1 = T_{t_1}(x)$, $T_2 = T_{t_2}(x)$, $T_1^* = T_1 T_{R_A^{'}}(T_k(x))$, $T_2^* = T_2 T_{R_B^{'}}(T_k(x))$,

$$\xleftarrow{Z_A, Z_B, ID_S, T_1^*} \qquad \xrightarrow{Z_A, Z_B, ID_S, T_2^*}$$

Chooses x_1, x_2, computes $X_1 = T_{x_1}(x)$, $X_2 = T_{x_2}(x)$, $X^{'} = X_1 T_{x_2}(T_k(x))$, and then computes $T_1 = T_1^* / (T_{Z_A \oplus M_A}(T_k(x)))$, $\alpha = T_{x_1}(T_1)$, $\alpha K_{AS} = F_\alpha(Z_A \| Z_B \| ID_S \| X^{'})$.

Chooses y_1, y_2, computes $Y_1 = T_{y_1}(x)$, $Y_2 = T_{y_2}(x)$, $Y^{'} = Y_1 T_{y_2}(T_k(x))$, and then computes $T_2 = T_2^* / (T_{Z_B \oplus M_B}(T_k(x)))$, $\beta = T_{y_1}(T_2)$, $\beta K_{BS} = F_\alpha(Z_A \| Z_B \| ID_S \| Y^{'})$.

$$\xrightarrow{X_2, X^{'}, \alpha K_{AS}} \qquad \xleftarrow{Y_2, Y^{'}, \beta K_{BS}}$$

Computes $X_1 = X^{'} / (T_k(X_2))$, $Y_1 = Y^{'} / (T_k(Y_2))$, $\alpha = T_{t_1}(X_1)$, $\beta = T_{t_2}(Y_1)$, checks whether $F_\alpha(Z_A \| Z_B \| ID_S \| X^{'}) \overset{?}{=} \alpha K_{AS}$, $F_\beta(Z_A \| Z_B \| ID_S \| Y^{'}) \overset{?}{=} \beta K_{BS}$. If they hold, chooses s, computes $X^* = T_s(X_1)$, $Y^* = T_s(Y_1)$, $\alpha K_A = F_\alpha(ID_S \| Z_A \| Z_B \| Y^*)$, $\beta K_B = F_\beta(ID_S \| Z_A \| Z_B \| X^*)$

$$\xleftarrow{Y^*, \alpha K_A} \qquad \xrightarrow{X^*, \beta K_B}$$

If $Y^* \neq 1$, checks whether $F_\alpha(ID_S \| Z_A \| Z_B \| Y^*) \overset{?}{=} \alpha K_A$. If it holds, computes $CS = T_{x_1}(Y^*)$, $SK = F_{CS}(1)$

If $X^* \neq 1$, checks whether $F_\beta(ID_S \| Z_A \| Z_B \| X^*) \overset{?}{=} \beta K_B$. If it holds, computes $CS = T_{y_1}(X^*)$, $SK = F_{CS}(1)$

Figure 2. Authentication key agreement phase

(6)When S receives both the messages from A and B,computes $X_1 = X'/(T_k(X_2))$, $Y_1 = Y'/(T_k(Y_2))$, $\alpha = T_{t_1}(X_1)$, $\beta = T_{t_2}(Y_1)$, and checks whether $F_\alpha(Z_A||Z_B||ID_S||X') \stackrel{?}{=} \alpha K_{AS}$, $F_\beta(Z_A||Z_B||ID_S||Y') \stackrel{?}{=} \beta K_{BS}$.If they hold, S chooses a random number s, and computes $X^* = T_s(X_1)$, $Y^* = T_s(Y_1)$, $\alpha K_A = F_\alpha(ID_S||Z_A||Z_B||Y^*)$, $\beta K_B = F_\beta(ID_S||Z_A||Z_B||X^*)$, and then sends $Y^*, \alpha K_A$ and $X^*, \beta K_B$ to A and B,respectively.
(7)When A and B receives the messages $Y^*, \alpha K_A$ and $X^*, \beta K_B$ from S.respectively, they firstly check whether Y^*, X^* are equal to 1.
If not, they check whether $F_\alpha(ID_S||Z_A||Z_B||Y^*) \stackrel{?}{=} \alpha K_A$, $F_\beta(ID_S||Z_A||Z_B||X^*) \stackrel{?}{=} \beta K_B$.
If they hold, A and B compute $CS = T_{x_1}(Y^*) = T_{y_1}(X^*)$, $SK = F_{CS}(1)$,and then accept and terminate the protocol.

3.4. **Password changing phase.** **Fig.3** shows the password changing phase as below (Taking an example of the user A):
(1)A opens the changing process, inputs his/her old identity and password ID_A, PW_A,and new password PW_A^{new},checks whether $M_A' = F_{PW_A}(ID_A||PW_A) \stackrel{?}{=} M_A$.If it holds, A computes $M_A^{new} = F_{PW_A^{new}}(ID_A$
$||PW_A^{new})$,chooses a random number c,and computes $K_c = T_cT_k(x)$, $C_c = E_{K_c}(M_A', Z_A, M_A^{new})$, and then sends $C_c, T_c(x)$ to server S.
(2)S computes $K_c' = T_kT_c(x)$,decrypts C_c by K_c',obtains M_A', Z_A, M_A^{new}.
Then S computes $R_A' = F_{M_A'}(M_A'||k)$,
and checks whether $R_A' \stackrel{?}{=} Z_A \oplus M_A'$. If it holds, S computes $R_A^{new} = F_{M_A^{new}}(M_A^{new}||k)$, $Z_A^{new} = R_A^{new} \oplus M_A^{new}$,and sends Z_A^{new} to A.
(3)A replaces M_A by M_A^{new} stored in secret in it, and replaces Z_A by Z_A^{new} made public.

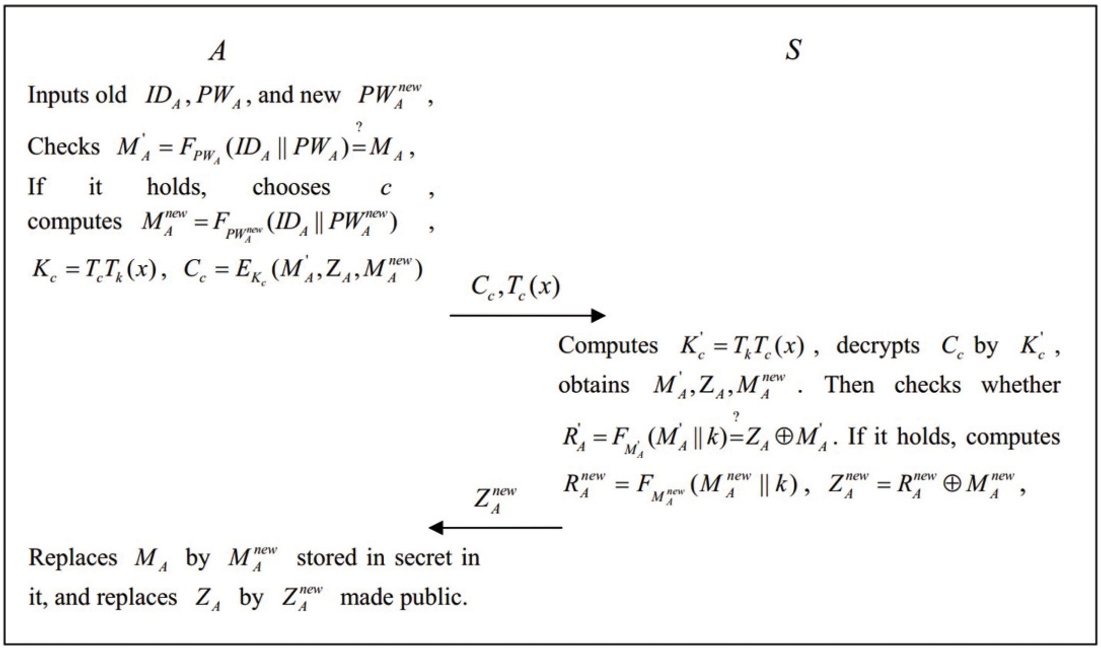

FIGURE 3. Password changing phase

4. **The provable security of the proposed scheme.** In this section, we introduce the standard model adopted in our paper. The standard model follows the ideas in work of

[8, 9, 12] for our proposed protocol.
The basic descriptions are shown in Table 2.

TABLE 2. Descriptions the model of Canetti and Krawczyk

Symbol	Definition
parties $P_1,\dots P_n$	Modeled 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 sessiosn matching	If the outgoing messages of one are the incoming messages of the other

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 the 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 4.1. *A key exchange protocol* Π_1 *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, these sessions produce the same key as output;
(2)$Advantage^{\Lambda}(k)$ is negligible.

Theorem 4.1. *Under the CMBDHP assumption, using the 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 [8].*

Proof: The proof is based on the proof given by Refs.[8]. 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 4.1**. 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.

Algorithm 1 CMBDHP distinguisher

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

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 Alice send $\{i, T_a(x), C_1\}$ to a server, and let a server send $\{i, Z_B\}$ to Bob. Then let Bob send $\{i, T_b(x), C_2\}$ to the server, where i is the session identifier. The server can compute two authenticator messages $\{T_1^*, T_2^*\}$ locally after authenticating each other 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**

8: $d \xleftarrow{R} \{0,1\}$.

9:**end if**

Output: d

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 4.2. *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*,*

(3)If two uncorrupted parties have completed matching sessions with pre-distributed parameter, these sessions produce the same key as output;
(4)$Advantage^{\Lambda}(k)$ is negligible.

Theorem 4.2. *Under the CMBDHP assumption, using the Algorithm 2 to compute session key is session-key secure in the adversarial model of Canetti and Krawczyk [8].*

Proof: The proofs process is similar to **Theorem 4.1**. The protocol Π_2 is the continuous instance of protocol multiple Π_1.All the messages of the process on protocol Π_2 are Under the CMBDHP assumption which is 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 our protocol.

Algorithm 2 Proposed protocol simulator

Input: $F, E_K()/D_K(), (x, T_k(x)), Z_A, Z_B$

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 the server send $\{Z_A, Z_B, ID_S, T_1^*\}/\{Z_A, Z_B, ID_S, T_2^*\}$ to Alice/Bob based on the protocol Π_1. Next, let Alice/Bob send $\{X_2, X', \alpha K_{AS}\}/\{Y_2, Y', \beta K_{BS}\}$ to the server. Finally, let the server send $\{Y^*, \alpha K_A\}/\{X^*, \beta K_B\}$ to Alice/Bob.

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**

8: $d \xleftarrow{R} \{0,1\}$.

9:**end if**

Output: d

5. **Security analysis of the proposed protocol.** In this section, we provide analysis to prove that our protocol is secure in the standard model. We mainly explain how our protocol achieves user anonymity in the standard model. Usually, in the random oracle model, if a user wants to protect personal sensitive information, one of the methods is hidden personal information in a pseudo-random function, and then the personal information is transferred over the channel in the way of a message which is an output result of a pseudo-random function. In addition, in [9] or some related protocols, the identities of users are allotted by the certificated server. Usually, in this condition, there is an identity table of users stored in the certificated server.

Once the server is invaded, the identities of users will be leaked. To solve these problems, we propose a novel method to achieve user anonymity in the standard model. We use a pseudo-random function, a secure symmetric encryption/decryption algorithm and Chebyshev chaotic maps to achieve our method (See **Fig. 4-1**). In addition, on the premise of achieving the user anonymity, our proposed protocol can still satisfy the security goals and resist various common attacks. Because of using the similar Chebyshev chaotic maps to solve the DDH assumption and discrete logarithm, the analysis of the security goals and the security proof of our proposed protocol are similar to that in [8, 9], therefore, it is omitted here. **Table 3** shows the security comparison between our proposed protocol and related protocols.

Table 4 shows the cost comparison between our proposed protocol and related protocols. According to **Table 3** and **Table 4**, we can know that our protocol gives the process in detail, and compared with related protocols, even though needing some more operations, our protocol is acceptable.

6. **Conclusion.** In our paper, according to the ideas of [9], we propose a novel ID-based authentication key agreement protocol with user anonymity on chaotic maps cryptosystem in the standard model.On the premise of achieving the user anonymity, our proposed protocol can still satisfy the security goals and resist various common attacks. Even though needing some more operations, our proposed protocol gives more detailed implementation

TABLE 3. Security comparisons

Security comparisons										
	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
[9]	Y	Y	Y	Y	Y	Y	Y	Y	N	Y
[10]	Y	Y	Y	Y	Y	N	N	Y	N	N
Our protocol	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y

Annotation: S1: Key privacy; S2: Mutual authentication;
S3: Client-to-server authentication; S4: Prefect forward security;
S5: Security against unknown key-share attack;
S6: Security against password compromised impersonation attack;
S7: Security against off-line dictionary attack;
S8: Security against undetectable on-line dictionary attack;
S9: User anonymity; S10: Standard model;
Y/N: Support/Not support

TABLE 4. Cost comparisons

Cost comparisons			
	C1(A/B/S)	C2(A/B/S)	C3(A/B/S)
[9]	------	6T+4F/ 6T+4F/ 10T+2F	------
[10]	------	2T+4H+2ED/ 2T+4H+2ED/ 2T+4H+4ED	------
Our protocol	1F+1T+1ED/ 1F+1T+1ED/ 1F+1T+1ED	4F+7T+1ED/ 4F+7T+1ED/ 6F+14T+2ED	2F+1T+1ED/ 2F+1T+1ED/ 2F+1T+1ED

Annotation: T: a Chebyshev chaotic maps operation;
ED: a Symmetric encryption/decryption operation;
F: a pseudo-random operation; H: a secure one-way hash operation;
C1: Communication cost in the user registration phase;
C2: Communication cost in the authentication key agreement phase;
C3: Communication cost in the password changing phase;
A: Participant A; B: Participant B; S: Server;
------: Not mentioned or not involve

process including the user registration phase and the password changing phase, compared with related protocols, our proposed protocol is acceptable.

REFERENCES

[1] G. Alvarez, S. Li, Some basic cryptographic requirements for chaos-based cryptosystems. *International Journal of Bifurcation and Chaos in Applied Sciences and Engineering,* vol. 16, no. 8, pp. 2129-2152, 2006.

[2] D. Xiao, X.F. Liao, and S.J. Deng, A novel key agreement protocol based on chaotic maps. *Information Sciences,* vol.177, no. 4, pp. 1136-1142,2007.

[3] S. Han, Security of a key agreement protocol based on chaotic maps. Chaos, *Solitons and Fractals,* vol. 38, no. 3, pp. 764-768, 2008.

[4] T. Xiang, K. W. Wong, X. F. Liao, On the security of a novel key agreement protocol based on chaotic maps. Chaos, *Solitons and Fractals,* vol.40, no. 2, pp. 672-675, 2009.

[5] H.R. Tseng, R.H. Jan,and W. Yang, A chaotic maps-based key agreement protocol that preserves user anonymity. *IEEE International Conference on Communications, ICC09*, Dresden, Germany, pp. 1-6, 2009.

[6] Y.J. Niu, X.Y. Wang, An anonymous key agreement protocol based on chaotic maps. *Communications in Nonlinear Science and Numerical Simulation,* vol.16, no. 4, pp. 1986-1992,2011.

[7] K.P. Xue, P.L. Hong, Security improvement on an anonymous key agreement protocol based on chaotic maps. *Communications in Nonlinear Science and Numerical Simulation*, vol.17, no. 7, pp. 2969-2977, 2012.

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

[9] H. Lai, M.A.Orgun, J.H.Xiao, J. Pieprzyk, L.Y. Xue, Y.X.Yang, Provably secure three-party key agreement protocol using Chebyshev chaotic maps in the standard model. *Nonlinear Dynamics,* vol. 77, no. 4, pp. 1427-1439, 2014.

[10] Q. Xie, J. M.Zhao, X. Y. Yu, Chaotic maps-based three-party password-authenticated key agreement scheme. *Nonlinear Dynamics,* vol. 74, no.4, pp. 1021-1027,2013.

[11] L.H. Zhang, Cryptanalysis of the public key encryption based on multiple chaotic systems. *Chaos Solitons Fract* ,2008.

[12] J.J. Zhao, and D.W.Gu, Provably secure three-party password-based authenticated key exchange protocol, *Information Sciences*, vol. 184, no. 1, pp. 310-323, 2012.

[13] M.S. Farash, M.A. Attari, An efficient and provably secure three-party password-based authenticated key exchange protocol based on Chebyshev chaotic maps. Nonlinear Dynamics, vol.77 no. 1-2, pp. 399-411(2014).

[14] C.C. Lee, C.T. Li, , S.T. Chiu, Y.M. Lai, A new three-party authenticated key agreement scheme based on chaotic maps without password table. *Nonlinear Dynamics,* vol. 79, no. 4, pp. 2485-2495, 2015.

[15] X.X. Hu, Z.F.Zhang, Cryptanalysis and enhancement of a chaotic maps-based three-party password authenticated key exchange protocol, *Nonlinear Dynamics*, vol. 78, no. 2, pp. 1293-1300, 2014.

[16] X. Li, J. W. Niu, S. Kumari, K. K. Khan,J. G. Liao, and W. Liang, Design and analysis of a chaotic maps-based three-party authenticated key agreement protocol, *Nonlinear Dynamics*, vol. 80, no. 3, pp. 1209-1220, 2015.