Journal of Information Hiding and Multimedia Signal Processing ©2016 ISSN 2073-4212
Ubiquitous International Volume **7**, Number **1**, January **2016**

A Multi-server Authenticated Key Agreement Protocol with Privacy preserving Based on Chaotic Maps in Random Oracle Model

Hongfeng Zhu

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

Dan Zhu

School of Foreign Languages
Shenyang Jianzhu University
No.9, HunNan East Street, HunNan District, Shenyang, P.C 110168 China
zhudan413@163.com

Yan Zhang

Software College
Shenyang Normal University
No.253, HuangHe Bei Street, HuangGu District, Shenyang, P.C 110034 China
1505733680@qq.com

Received October, 2014; revised September, 2015

Abstract. *In the research literature, a typical authenticated key agreement protocol is designed to realize mutual authentication and key secrecy. In a network, as a crucial cryptographic primitive, key agreement can establish secure communication channels for communication entities. Meanwhile, the users privacy is particularity important, especially for multi-server authentication environment. Network privacy security means that the personal data and online data are not peep, intrusion, interference, illegal collection and utilization. In our paper, we propose a robust chaotic maps-based authentication key agreement scheme with privacy protection using smart cards for multi-server authentication environment. The key idea of our proposed scheme is to adopt chaotic maps for mutual authentication, not to encrypt/decrypt messages transferred between user and server, which can make our proposed scheme much more efficient. In addition, our protocol can realize the users privacy preserving and various common security features. As a whole, compared with other related protocol, our proposed protocol is more secure and practical.*
Keywords: Chaotic maps, Biometrics, Multi-server, Privacy preserving, Smart card.

1. **Introduction.** As a special form of motion, Chaos means that in a certain nonlinear system can appear similar to the behavior of random phenomena without needing any random factors. Chaotic system is very sensitive to initial parameters, thus the chaotic sequence produced by chaos has the nature of non-periodicity and pseudo-randomness. Analogously, chaotic system has the characteristics of certainty, boundedness, sensibility to initial parameters and unpredictability, etc. Because of the similar characteristics of chaotic system and cryptosystem, chaos theory has been widely noted and used by

cryptographic circle. Biological technology is a technology based on our own inherent physiological or behavioral characteristics. Due to the unrepeatable uniqueness of the biometrics characteristics, they are extremely difficult to be copied or stolen. In addition, biometrics authentication has a very good experience, thus is being widely used. As is known to all, smart card has powerful information confidentiality and flexible portability. It is so convenient. However, it also has many disadvantage factors: easily forgotten, replication, lost, stolen, etc. To address this problem, combined the smart card with biometrics technology, the security of the smart card can be improved. In 2000, Lee et al. [1] proposed an anonymous identification protocol to successfully provide authentication and anonymity for multi-server environment. In 2001, Tsaur [2] proposed a smart card based remote login authentication scheme for multi-server Internet environments, which can certificate a single password for logging multiple authorized servers without using any password verification table. Unfortunately, Kim et al. [3] pointed out that the authentication scheme of [2] is vulnerable to the off-line guessing attack. However, Kim et al. did not propose any improvement method about [2]. In 2005, Tsaur et al. [4] proposed an improvement scheme using the RSA cryptosystem and Lagrange interpolating to overcome the weaknesses of [2]. In 2008, Tsai [5] proposed an alternative multi-server authentication scheme using smart cards, which was based on the nonce, used one-way hash function, and did not store any verification table in the server and registration center. Unfortunately, Wang et al. [6] pointed out that the scheme of [5] cannot resist server spoofing attack and impersonation attack in 2009, and proposed a remote user authentication scheme to withstand these attacks. What is a pity is that the schemes of [5] and [6] cannot achieve the prefect forward secrecy. In 2013, Chen et al. [13] pointed that in the both schemes of [5] and [6], there were some flaws, and proposed a new improving scheme.

Recently, many schemes have been proposed for multi-server environment. In 2011, Chang et al. [7] proposed a smart card based remote authentication mechanism for multi-server environment and claimed that their proposed protocol could achieve secure communication. In 2012, Wang et al. [9] proposed a smart card based efficient and secured multi-server authentication scheme which is validated and verified by Colored Petri Nets. Unfortunately, Yeh et al. [8] pointed that the protocol of [7] had a defect of session key disclosure in 2013, and proposed a novel authentication scheme which was given the formal security analysis proofs. In the same year, He et al. [10] pointed that the scheme of [9] was vulnerable to the server spoofing attack, the impersonation attack, the privileged insider attack and the off-line password guessing attack. In addition, Pippal et al. [11] proposed a robust multi-server authentication scheme using smart card, and claimed that their proposed scheme was secure which has been validated by using BAN logic in 2013. However, in 2014, Guo et al. [12] pointed that the scheme of [11] could not resist the impersonation attack and the off-line password guessing attack, and proposed an improved multi-server authentication scheme which could preserver user anonymity. Nowadays, chaos theory has been widely noted and used by cryptographic circle. Recently, in 2013, Guo et al. [13] proposed a chaotic maps-based password-authenticated key agreement protocol with smart cards which avoids modular exponential computing or scalar multiplication on an elliptic curve. In the same year, Xie et al. [14] firstly proposed a chaotic maps-based three-party password-authenticated key agreement (3PAKA) scheme without using a timestamp. In this paper, we propose a provable biometrics-based multi-server authenticated key agreement protocol with privacy preserving on chaotic maps cryptosystem.

Our contribution has the following several aspects: (1) In a multi-server environment, user privacy can be effectively protected. (2) The modular exponential computing or

scalar multiplication can be avoided. (3) The chaos theory is only used for communication between entities of mutual authentication, rather than encrypting or decrypting messages. (4) Combining biometrics authentication with chaos theory, the proposed protocol has better experience and security. (5) Proposed protocol can satisfy the common security requirements.

The rest of the paper is organized as follows: In the next section, we review some preliminaries. Sect. 3 describes our proposed scheme. Sect. 4, 5 and 6 discuss the security, functionality and efficiency of the proposed scheme. Finally, the paper is concluded in Sect. 7.

2. **Preliminaries.** The concepts of Chebyshev chaotic maps, biometrics authentication are introduced in below, respectively.

2.1. **Chebyshev chaotic maps.** Chebyshev polynomial and chaotic maps [14] have the following properties:
(1) Let n be an integer and let x be a variable with the interval $[-1, 1]$. The Chebyshev polynomial $T_n(x) : [-1, 1] \to [-1, 1]$ is defined as

$$T_n(x) = \cos(ncos^{-1}(x)). \tag{1}$$

In terms of (1), the recurrence relation of Chebyshev polynomial is defined as

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

(2) The properties of Chebyshev polynomial:

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 [15]: The semi-group property of Chebyshev polynomial defined on the interval $(-\infty, +\infty)$ holds, as below:

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

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

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

Besides, assume that the following problems are 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 $T_s = y$.
(4) Chaotic Maps-Based DiffieHellman problem (CMDHP): Given x, $T_r(x)$, $T_s(x)$, it is intractable to find $T_{rs}(x)$, such that $T_r(T_s(x)) = T_{rs}(x)$ or $T_s(T_r(x)) = T_{rs}(x)$.

2.2. **Biometrics certification.** Fig.1 shows the flow chart of biometrics certification in detail. In the biometrics collection model, user inputs the biometrics in a biometric sensor, and then the system performs biometrics collection, detail information extraction, and stores it in the biometrics database. In the certification model, after extracting detail information, the system submits it to the database and compares it with the stored information, and then outputs the result.

3. **The proposed protocol.** In this paper, we propose a biometrics-based multi-server key agreement protocol with privacy preserving on chaotic maps cryptosystem. In this part, we describe the proposed protocol which is composed of four phases: user registration phase, server registration phase, authenticated key agreement phase, password and biometrics changing phase, respectively.

In Table.1, the signs used in the proposed protocol are shown in detail as below:

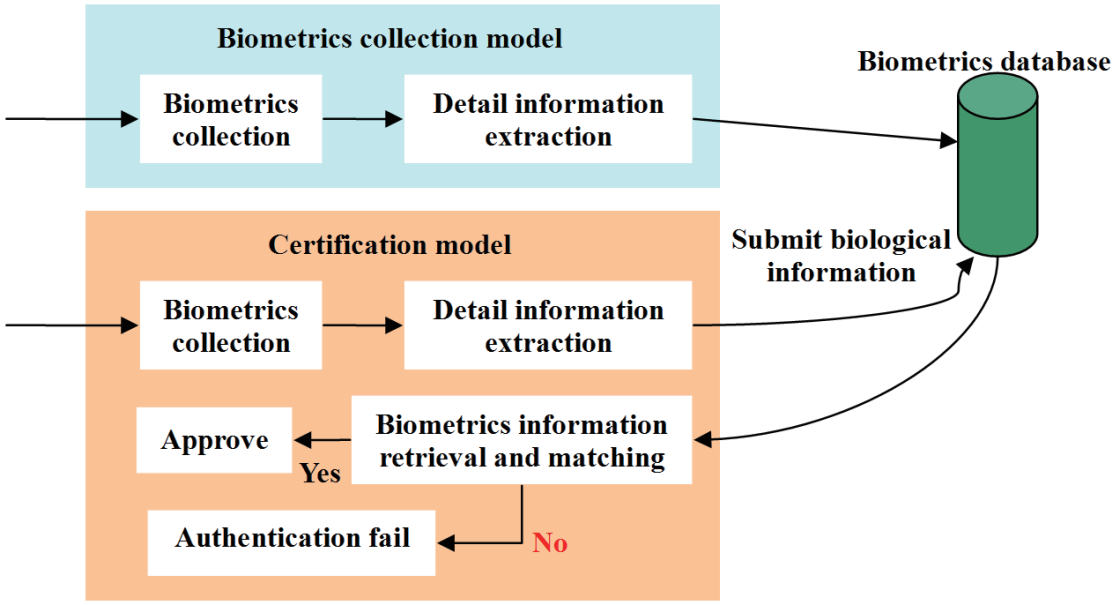

FIGURE 1. The flow chart of biometrics certification

TABLE 1. Signs

Sign	*Definition*
$U_i, ID_{U_i}, PW_{U_i}, B_{U_i}$	The ith user; the identity, password, biometrics sample of the ith user, respectively
S_j, ID_{S_j}	The jth server， the identity of the jth server, respectively
RC	Registration center
$d(\cdot)$	Symmetric parametric function
τ	Predetermined threshold for biometrics certification
$(x, T_k(x)), k$	The public key and secret key of RC
u, s, rc	Random integer number
sk	Session key
$h(\cdot)$	Secure one-way hash function
$\oplus, \|\|$	XOR operation, concatenation operation, respectively

3.1. **User registration phase.** A user U_i must be certificated by registration center before he/she communicates with the servers $S_j(1 \leq j \leq n)$ in a multi-server environment. Fig.2 expounds the user registration phase as below:

Step.1 U_i optionally chooses his/her identity ID_{U_i}, password PW_{U_i}, and collects his/her biometrics sample B_{U_i} through a biological sensor. Then U_i computes $M_{U_i} = h(ID_{U_i}||PW_{U_i})$, $U_{U_i} = M_{U_i} \oplus h(B_{U_i})$, and sends $\{U_{U_i}, h(B_{U_i})\}$ to RC.

Step.2 RC computes $R_{U_i} = h(h(B_{U_i})||k)$, $Z_{U_i} = R_{U_i} \oplus U_{U_i}$, and then stores $\{Z_{U_i}, U_{U_i}, h(\cdot), d(\cdot), \tau\}$ in a smart card and gives it to U_i via a secure channel. When U_i obtains the smart card, he/she stores B_{U_i} in it. Need to add that the sign $d(\cdot)$ is symmetric parametric function and the sign τ is predetermined threshold for biometrics certification.

3.2. **Server registration phase.** In a multi-server environment, if a server $S_j(1 \leq j \leq n)$ wants to provide services for U_i, the server $S_j(1 \geq j \geq n)$ must have been certificated by the registration center RC. Fig.3 expounds the server registration phase as below:

Step.1 S_j chooses its identity ID_{S_j} and sends it to RC via a secure channel.

Step.2 RC computes $R_{S_j} = h(h(ID_{S_j}))$, $S_{S_j} \oplus h(ID_{S_j})$, and sends S_{S_j} to S_j via secure

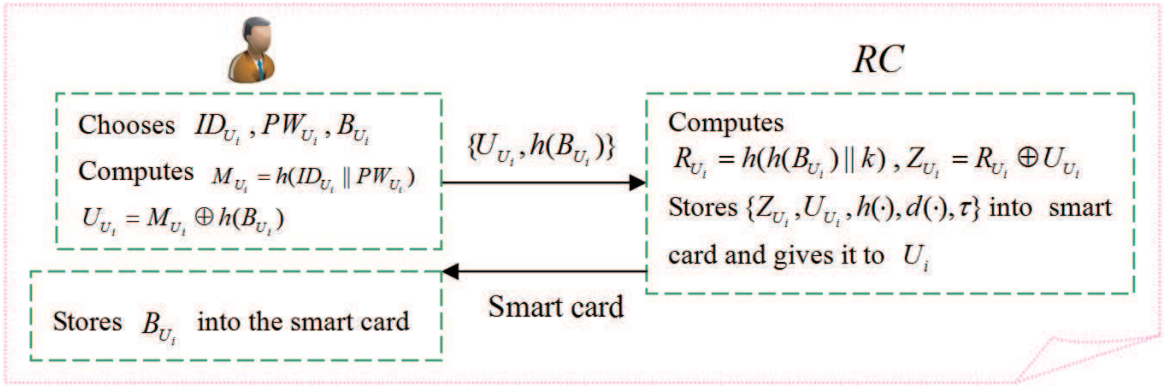

FIGURE 2. User registration phase

channel.

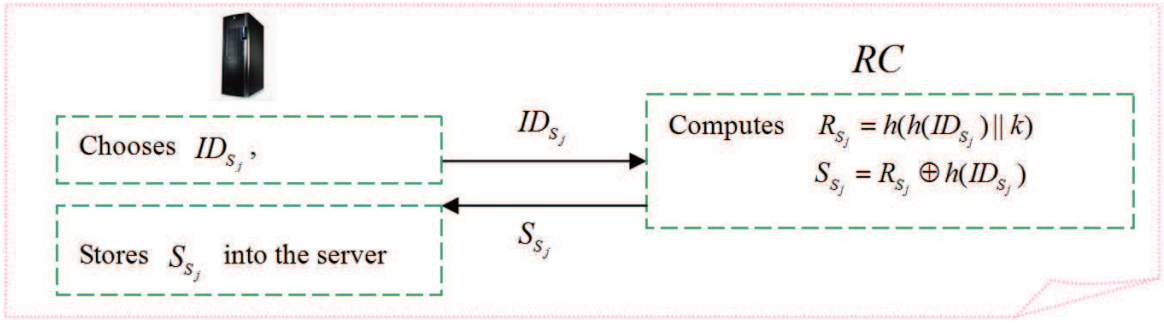

FIGURE 3. Server registration phase

3.3. **Authenticated key agreement phase.** In the authenticated key agreement phase, U_i and S_j can authenticate each other and establish the session key sk, meanwhile, U_i can protect his/her privacy. Fig.4 expounds the authenticated key agreement phase as below:

Step.1 U_i inputs his/her smart card into a smart card reader, opens the access software, starts the biosensor, imprints his/her biometric $B^{\gamma}_{U_i}$, and then the biometrics certification program compares $B^{\gamma}_{U_i}$ with B_{U_i} stored in the smart card. If $d(B^{\gamma}_{U_i}, B_{U_i}) \geq \tau$ holds, a refused response is given to U_i; if $d(B^{\gamma}_{U_i}, B_{U_i}) < \tau$ holds, an accepted response is given to U_i. Then U_i inputs ID_{U_i}, PW_{U_i}, computes $U^*_{U_i} = h(ID_{U_i}||PW_{U_i}) \oplus h(B_{U_i})$, and checks whether $U^*_{U_i} \stackrel{?}{=} U_{U_i}$. If it does not hold, U_i gets a Wrong password message; if it holds, U_i chooses a random integer number u, computes $R_{U_i} = Z_{U_i} \oplus U_{U_i}$, $C = T_u T_k(x)$, $V_i(R_{U_i}, C)$, and then sends $\{V_i, h(B_{U_i}), T_u(x)\}$ to S_j.

Step.2 S_j chooses a random integer number s, and computes $G = T_s T_k(x)$, $F_j = h(ID_{S_j}, G)$, and then sends $\{V_i, h(B_{U_i}), T_u(x), F_j, ID_{S_j}, ID_{S_j}, T_s(x)\}$ to RC.

Step.3 RC computes $R^*_{U_i} = h(h(B_{U_i})||k)$, $R^*_{S_j} = h(h(B_{U_i})||k)$, $C^* = T_k T_u(x)$, $V^*_i = h(R^*_{U_i}, C^*)$, $G^* = T_k T_s(x)$, $F^*_j = h(R^*_{S_j}, G^*)$, and checks whether $V^*_i \stackrel{?}{=} V_i$, $F^*_j \stackrel{?}{=} F_j$. If they do not hold, RC refuses the session request; if they hold, RC chooses a random integer number rc, and computes $Q = T_{rc} T_s(x)$, $Y = T_{rc} T_u(x)$,$C_{S_j} = h(F_j, Q)$, $C_{U_i} = h(V_i, Y)$, and sends $\{C_{S_j}, C_{U_i}, T_{rc}(x)\}$ to S_j.

Step.4 S_j computes $Q^* = T_s T_{rc}(x)$, $C^*_{S_j} = h(F_j, Q^*)$, and checks whether $C^*_{S_j} \stackrel{?}{=} C_{S_j}$. If it does not hold, S_j stops this session request; if it holds, S_j computes $sk = T_s T_u(x)$,

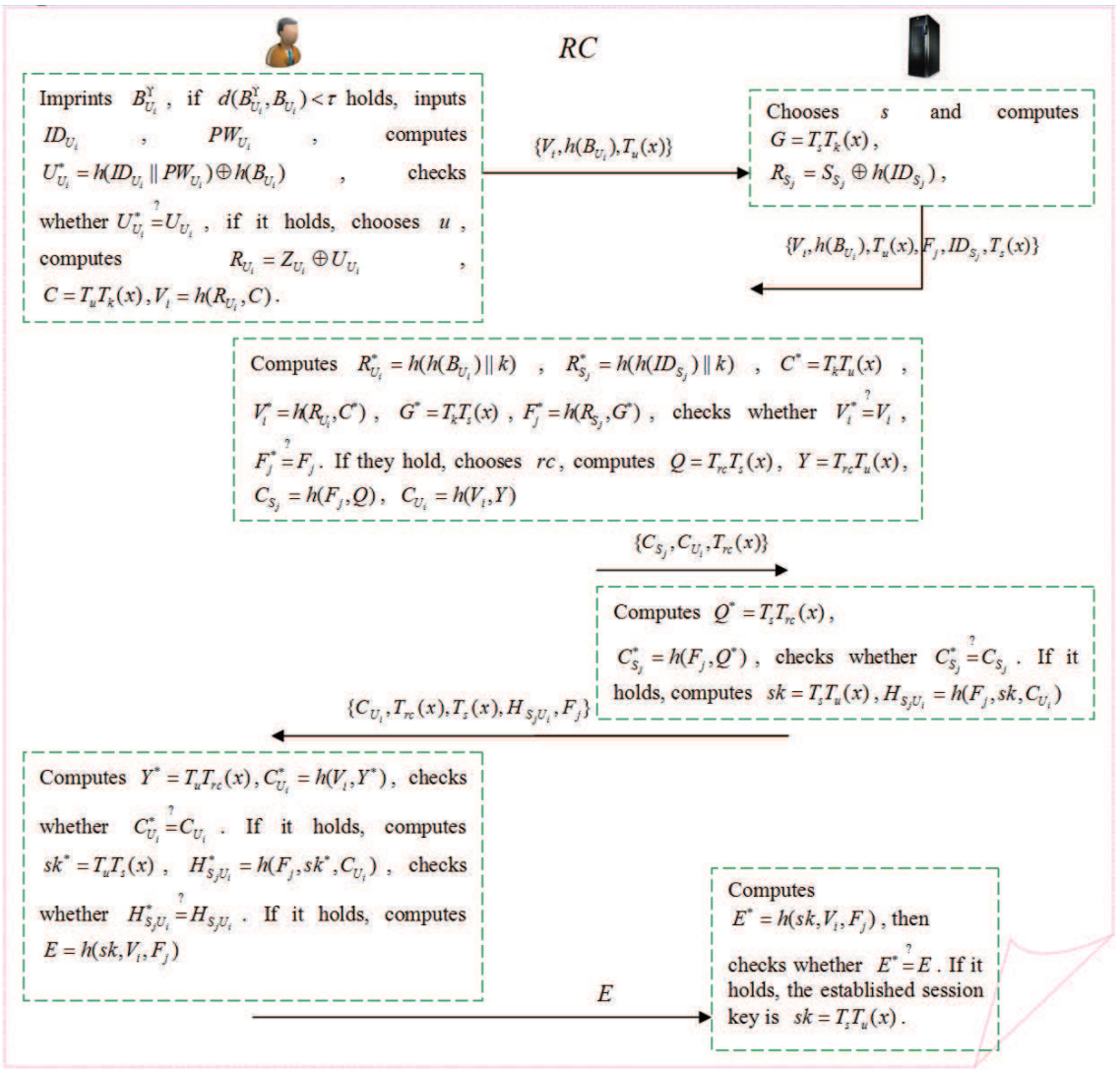

FIGURE 4. Authenticated key agreement phase

$H_{S_jU_i} = h(F_j, sk, C_{U_i})$, and sends $\{C_{U_i}, T_{rc}(x), T_s(x), H_{S_jU_i}, F_j\}$ to U_i.

Step.5 U_i computes $Y^* = T_u T_{rc}(x)$, $C_{U_i}^* = h(V_i, Y^*)$, and checks whether $C_{U_i}^* \stackrel{?}{=} C_{U_i}$. If it does not hold, U_i stops this session request; if it holds, U_i computes $sk^* = T_u T_s(x)$, $H_{S_jU_i}^* = h(F_j, sk^*, C_{U_i})$, and checks whether $H_{S_jU_i}^* \stackrel{?}{=} H_{S_jU_i}$. If it does not hold, U_i stops this session request; if it holds, U_i computes $E = h(sk, V_i, F_j)$, and sends E to S_j.

Step.6 S_j computes $E^* = h(sk, V_i, F_j)$, and checks whether $E^* \stackrel{?}{=} E$. If it holds, U_i and S_j authenticate each other and the established session key is $sk = T_s T_u(x)$.

3.4. **Password and biometrics changing phase.** Fig.5 expounds the password and biometrics changing phase as below:

U_i inputs his/her smart card into a smart card reader, opens the password and biometrics changing software, starts the biosensor, imprints his/her biometric $B_{U_i}^*$, and then the biometrics comparison program compares $B_{U_i}^*$ with B_{U_i} stored in the smart card. If $d(B_{U_i}^*, B_{U_i}) \geq \tau$ holds, a refused response is given to U_i; if $(B_{U_i}^*, B_{U_i}) < \tau$ holds, gives U_i the message that Please enter your identity and password:. U_i inputs ID_{U_i}, PW_{U_i}, the

smart card computes $U_{U_i}^* = h(ID_{U_i}||PW_{U_i}) \oplus h(B_{U_i})$, and checks whether $U_{U_i}^* \stackrel{?}{=} U_{U_i}$. If it does not hold, a Wrong password message is given to U_i; if it holds, a Allowed to change message is given to U_i.

Next, we describe the changing phase in the following three cases:

(1) Only changing the password

Step.1 U_i inputs his/her new password $PW_{U_i}^{new}$. The smart card automatically computes $U_{U_i}^{new} = h(ID_{U_i}||PW_{U_i}^{new}) \oplus h(B_{U_i}^{new})$, $Z_{U_i}^{new} = Z_{U_i} \oplus U_{U_i} \oplus U_{U_i}^{new}$, and then replaces $\{Z_{U_i}, U_{U_i}\}$ by $\{Z_{U_i}^{new}, U_{U_i}^{new}\}$ to be stored in it.

(2) Only changing the biometrics

Step.1 U_i inputs his/her new biometrics $B_{U_i}^{new}$, The smart card automatically computes $U_{U_i}^{new} = h(ID_{U_i}||PW_{U_i}) \oplus h(B_{U_i}^{new})$, and sends $\{Z_{U_i}, U_{U_i}, h(B_{U_i}), h(B_{U_i}^{new}), U_{U_i}^{new}\}$ to RC.

Step.2 RC checks whether $h(h(B_{U_i})||k) \stackrel{?}{=} Z_{U_i} \oplus U_{U_i}$. If it does not hold, RC refuses the changing request; if it holds, RC computes $R_{U_i}^{new} = h(h(B_{U_i}^{new})||k)$, $Z_{U_i}^{new} = R_{U_i}^{new} \oplus U_{U_i}^{new}$, and sends $\{Z_{U_i}^{new}, U_{U_i}^{new}\}$ to the smart card.

Step.3 The smart card replaces $\{Z_{U_i}, U_{U_i}, B_{U_i}\}$ by $\{Z_{U_i}^{new}, U_{U_i}^{new}, B_{U_i}^{new}\}$ to be stored in it.

(3) Changing the password and biometrics

Step.1 U_i inputs his/her new $PW_{U_i}^{new}$, $B_{U_i}^{new}$. The smart card automatically computes $U_{U_i}^{new} = h(ID_{U_i}||PW_{U_i}^{new}) \oplus h(B_{U_i}^{new})$, and sends $\{Z_{U_i}, U_{U_i}, h(B_{U_i}), h(B_{U_i}^{new}), U_{U_i}^{new}\}$ to RC. The following steps are same with the Step.2 and Step.3 of **(2)**.

4. Security analysis.

4.1. **Security analysis under the random oracle model.** Usually, the random oracle model is used to certificate the security of the key agreement protocol. In this subsection, we use it to analyze the security of the proposed scheme. In the random oracle model, each participant in the authenticated key agreement phase is treated as an oracle, and meanwhile, an adversary can obtain these oracles by sending some queries.

The adversarial model is introduced as below. Suppose that the multi-server environment includes three types of participants: n users $U = \{U_1, U_2, ..., U_n\}$, m servers $S = \{S_1, S_2, ..., S_n\}$ and a registration center RC. The ith instance of U is indicated as $\prod_U^i$ and the jth instance of S is indicated as $\prod_S^j$. Suppose that an adversary A is a probabilistic polynomial time machine, it is able to control all messages transferred in the proposed scheme via accessing to a set of oracles (as defined below). The public parameters are known by each participant.

(1) $Extract(ID_i)$ query: A can obtain the private key of ID_i In Extract query model.

(2) $Send(\prod_c^k, M)$ query: A can send a message M to the oracle $\prod_c^k$ in Send query model, where $c \in \{U, S\}$. After receiving the message M, $\prod_c^k$ responds to A according to the proposed scheme.

(3) $h(m_i)$ query: When A makes hash query with message m_i in the hash query model, the oracle $\prod_c^k$ returns a random number r_1 and records (m_i, r_1) into a list L_H which is initially empty.

(4) $Reveal(\prod_c^k)$ query: A can obtain a session key sk from the oracle $\prod_c^k$ in Reveal query model if the oracle $\prod_c^k$ has accepted. Otherwise, $\prod_c^k$ returns a null to A.

(5) $Corrupt(ID_i)$ query: A can issue this query to ID_i and gets back its secret key.

(6) $Test(\prod_c^k)$ query: When A asks a test query to an oracle $\prod_c^k$ in Test query model, the oracle chooses a random bit $b \in \{0, 1\}$. If $b = 1$, $\prod_c^k$ returns the session key. Otherwise, $\prod_c^k$ returns a random value. Test query can measure the semantic security of the session key.

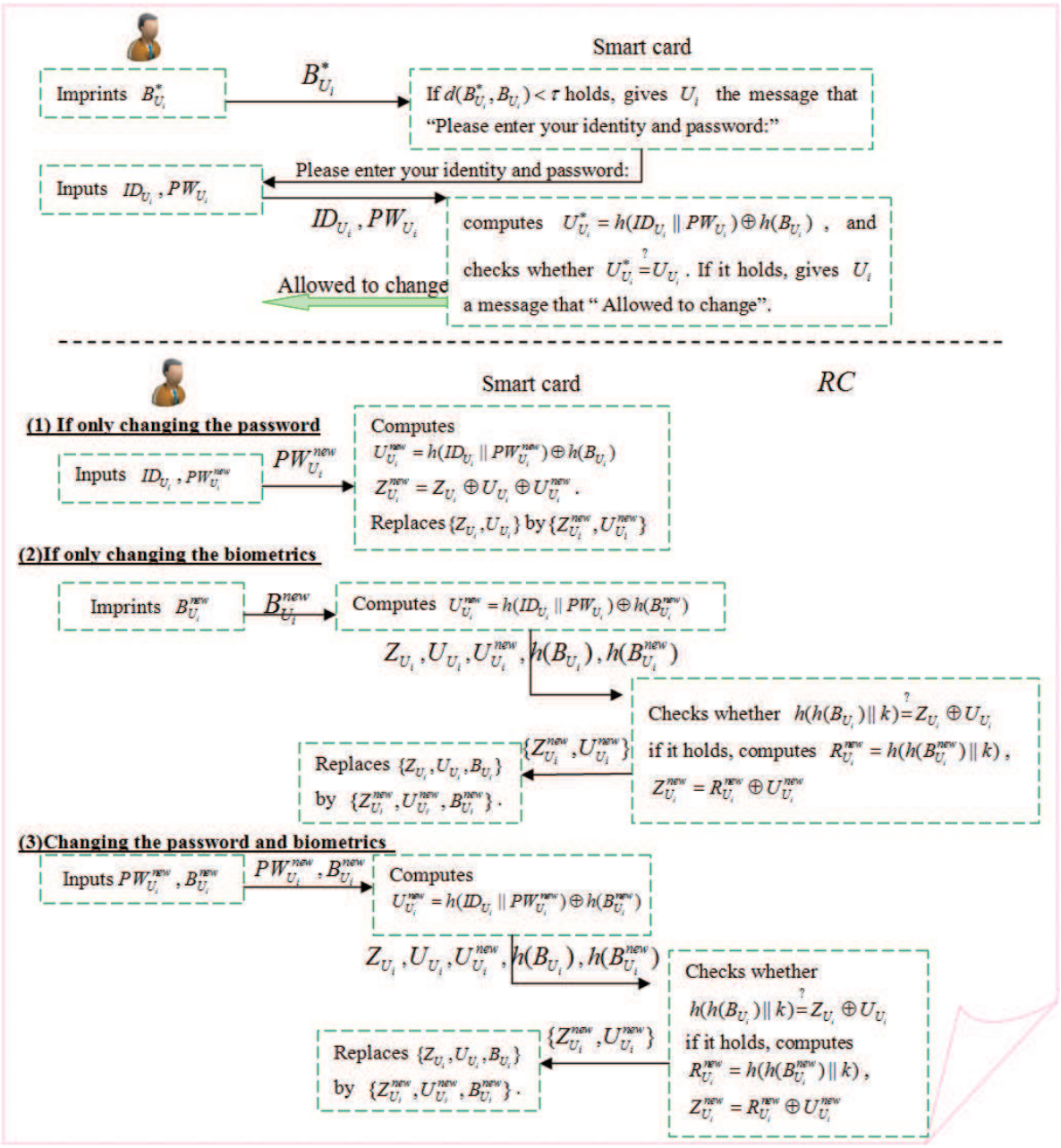

FIGURE 5. Password and biometrics changing phase

In this adversarial model, A can make *Send*, *Reveal*, *Corrupt* and *Test* queries. It be note worthy that the capabilities of the adversary can make finite queries under adaptive chosen message attacks.

Next, we introduce that the proposed scheme can provide the secure authenticated key agreement under the computational Chaotic Maps-based DiffieHellman problem (CMDHP) assumption.

Theorem1. Suppose that the adversary A can infringe the proposed scheme with a non-negligible advantage ϵ and makes at most q_u, q_s, q_h queries to the oracle of the user $\prod_U^i$, oracle of the server $\prod_S^j$, and h, respectively. Then we can construct an algorithm to solve the CMDHP with a non-negligible advantage.

Proof. We firstly assume that the types of attack are into two categories: impersonating the user to communicate with RC and impersonating the server to communicate with

RC. Then we can construct an algorithm to solve the CMDHP.

For an instance of CMDHP $\{x, P_1 = T_k(x), P_2 = k\}$, B simulates the system initializing algorithm to generate the system parameters $\{x, P_{pub-u} = P_1, h\}$, h is random oracles controlled by B. Then B gives the system parameters to A and interacts with A as follows.

$h - query$: B maintains a list L_h of tuples (str_i, h_i). When A queries the oracle h on (str_i, h_i), B responds as follows: If str_i is on L_h, B responds with h_i. Otherwise, B randomly chooses an integer h_i that is not found in L_h, and adds (str_i, h_i) into L_h, then responds with h_i.

$Reveal - query$: When the adversary A makes a $Reveal(\prod_c^m)$ query, B responds as follows. If $\prod_c^m$ is refused, B responds none. Otherwise, B examines the list L_h and responds with the corresponding h_i.

$Send - query$: When A makes a $Sedn(\prod_c^m, "start")$ query, B responds as follows. If $\prod_c^m = \prod_U^m$, B sets $T_u(x) \leftarrow P_1$, and randomly generates the values V_i and $h(B_{U_i})$. Otherwise, B generates a random number u^*, and computes $T_u(x) \leftarrow T_{u^*}(x)$, $C^* = T_{P_2}(T_{u^*}(x))$, $V_{i^*} = h(h(h(B_{U_i})||P_2), C^*)$, and responds with $\{V_{i^*}, h(B_{U_i}), T_{u^*}(x)\}$, where $h(B_{U_i})$ is generated by B. The simulation works correctly because A cannot distinguish whether $h(B_{U_i})$ is valid.

When A makes a $Send(\prod_c^m, \{V_{i^*}, h(B_{U_i}), T_{u^*}(x), F_{j^*}, ID_{S_j}, T_s(x)\})$ query, B responds as follows. If $\prod_c^m = \prod_U^m$, B stops the game. Otherwise, B computes $R_{U_i}^* = h(h(B_{U_i})||P_2)$, $R_{S_j}^* = h(h(ID_{S_j})||P_2)$, $C^* = T_{u^*}(T_{P_2}(x))$, $G^* = T_s(T_{P_2}(x))$, then checks whether $V_i = h(R_{U_i}^*, C^*) \stackrel{?}{=} V_{i^*}$, $F_j = h(R_{S_j}^*, G^*) \stackrel{?}{=} F_{j^*}$ to authenticate U_i and S_j. If they hold, B generates a random number rc^*, computes $Q^* = T_{rc^*}T_s(x)$, $Y^* = T_{rc^*}T_{u^*}(x)$, $C_{S_j}^* = h(F_{j^*}, Q^*)$, $C_{U_i}^* = h(V_{i^*}, Y^*)$, and then responds the corresponding message according to the description of the proposed scheme.

When A makes a $Send(\prod_c^m, \{C_{U_i}^*, T_{rc^*}(x), F_{j^*}, H_{S_j}U_i, T_s(x)\})$ query, B responds as follows. If $\prod_c^m = \prod_S^m$, B stops the game. Otherwise, B computes $Y^* = T_{u^*}T_{rc^*}(x)$, $C_{U_i}^* = h(V_{i^*}, Y^*)$, and checks whether $C_{U_i}^* \stackrel{?}{=} C_{U_i}$. If it holds, B computes $sk^* = T_{u^*}T_s(x)$.

If A can infringe a user to the RC authentication, it means that A can obtain the value of $R_{U_i}^* = h(h(B_{U_i})||P_2)$ from the list L_h, and then know the session key $sk^* = T_{u^*}T_s(x)$. It means that B is able to solve the CMDHP with non-negligible probability. In addition, P_2 is the secret key of RC and h is a secure one-way hash function. It is impossible for A to compute the value of $R_{S_j}^*$. From the above analysis, A can infringe the user to the RC authentication is negligible.

If A can infringe a server to the RC authentication, it means that A can obtain the value of $R_{S_j}^* = h(h(B_{S_j})||P_2)$ from the list L_h, and then know the session key $sk^* = T_{u^*}T_s(x)$. It means that B is able to solve the CMDHP with non-negligible probability. In the same reason, it is impossible for A to compute the value of $R_{S_j}^*$. From the above analysis, A can infringe the server to the RC authentication is negligible.

If B can win the game, B must have made the corresponding $h - query$ from the list L_h to find the correct h_i with non-negligible probability because h is a random oracle. From all the above analysis, it is a contradicting to the intractability of the CMDHP.

4.2. **Other security features. Perfect forward secrecy**

Perfect forward secrecy is that the following established session keys do not depend on the previously established session keys. Supposing that the adversary knows the previous parameters $T_u(x)$ and $T_s(x)$, the adversary cannot obtain the following session key because when establishing the following session key, U_i and S_i reselect the random parameters $T_{u^*}(x)$ and $T_{s^*}(x)$ which are not inferred from the previously parameters $T_u(x)$

and $T_s(x)$. Thus our proposed protocol can achieve perfect forward secrecy.

Known-key secrecy

Known-key secrecy is that even if a session key is known by the adversary, he/she also cannot know the previous and following session key. Supposing that the adversary intercepts a session key $sk = T_m(T_n(x))$ and knows random parameters m and n, he/she cannot obtain the previous and the future session keys because of unknown the corresponding random parameters m and n.

Mutual authentication and key agreement

Mutual authentication and key agreement means communication entities can authenticate each other and establish a session key. In the user registration phase and server registration phase, RC uses its secret key k to compute $R_{U_i} = h(h(B_{U_i})||k)$ and $R_{U_i} = h(h(B_{U_i})||k)$, respectively. Then U_i stores R_{U_i} in $\{Z_{U_i}, U_{U_i}\}$ and S_j stores R_{S_j} in S_{S_j}. In the authenticated key agreement phase, after receiving $\{V_{i^*}, h(B_{U_i}), T_u(x), F_j, ID_{S_j}, T_s(x)\}$ from S_j, RC computes $R^*_{U_i} = h(h(B_{U_i})||k)$, $R^*_{S_j} = h(h(ID_{S_j})||k)$, $C^* = T_k T_u(x)$, $G^* = T_k T_s(x)$, checks whether $V_i = h(R^*_{U_i}, C^*) \stackrel{?}{=} V_{i^*}$, $F_j = h(R^*_{S_j}, G^*) \stackrel{?}{=} F_{j^*}$, if they hold, it means that U_i and S_j are authenticated by RC; When S_j receives $\{C_{S_j}, C_{U_i}, T_{rc}(x)\}$ from RC, S_j computes $Q^* = T_s T_{rc}(x)$, checks whether $C^*_{S_j} = h(F_j, Q^*) \stackrel{?}{=} C_{S_j}$, if it holds, it means that RC is authenticated by S_j; When U_i receives $\{C_{U_i}, T_{rc}(x), T_s(x), H_{S_j}U_i, F_j\}$ from S_j, U_i firstly computes $Y^* = T_u T_{rc}(x)$, checks whether $C^*_{U_i} = h(V_i, Y^*) \stackrel{?}{=} C_{U_i}$, if it holds, it means that RC is authenticated by U_i, then U_i computes $sk^* = T_u T_s(x)$, checks whether $H^*_{S_jU_i} = h(F_j, sk^*, C_{U_i}) \stackrel{?}{=} H_{S_jU_i}$, if it holds, it means that S_j is authenticated by U_i; when S_j receives E from U_i, S_j checks whether $E^* = h(sk, V_i, F_j) \stackrel{?}{=} E$, if it holds, it means that U_i is authenticated by S_j. After U_i and S_j authenticate each other, the established session key is $sk = T_s T_u(x)$.

Secure password and biometrics update protocol

In our proposed protocol, the adversary can do nothing because the proposed protocol cannot work off-line. Otherwise, it is so easy to guess the correct password and change it. In addition, because the biometrics was stored in the smart card, the proposed update protocol can utilize the old password and biometrics to check the authenticity of the information of U_i.

Server spoofing attack

Our proposed scheme can resist server spoofing attack. The adversary cannot masquerade as a server to spoofing the legal user U_i or the registration center RC in our proposed protocol. If the adversary attempts to masquerade as a serve S_j to spoof RC, without knowing the value of , he/she cannot compute the correct value of F_j; in addition, even if the adversary can intercept the correct F_j transferred on the channel, he/she cannot obtain the correct value of G because of the CMDLP and CMDHP. Thus, the adversary fails to impersonate as S_j to cheat RC. Since the adversary cannot obtain a valid value C_{S_j}. Therefore, the adversary cannot cheat the legal user U_i, too.

Registration center spoofing attack

Our proposed scheme can resist registration center spoofing attack. If the adversary attempts to masquerade as the registration center RC to spoof the user U_i and the server S_j, he/she must will fail because the secret key k of the real RC is only known by itself, others cannot obtain the value of k in any way. Thus, the adversary is impossible to know the correct value of R_{S_j} and R_{U_i}, impossible to pass the authentication by S_j and U_i.

Insider attack

Our proposed scheme can resist insider attack. As a malicious insider adversary, he/she

always attempts to maliciously gain the personal information of users using his/her authorized access. However, in our proposed, PW_{U_i} and ID_{U_i} are protected in a secure hash function, the adversary is impossible to know PW_{U_i} and ID_{U_i} from the hash function.

Impersonation attack/Man-in-the-middle attack/ Replay attack

Our proposed scheme can resist impersonation attack, man-in-the-middle attack and replay attack. According to the proposed protocol, even if the adversary intercepts all the messages transferred on the channel, attempts to masquerade as the communication entities to pass the authentication with each other, and finally obtain the session key. He/she cannot successfully obtain the session key $sk = T_sT_u(x)$ because of the CMDLP and CMDHP. Meanwhile, when transferred on the channel, sk is always protected in a one-way hash function. Thus sk cannot be extracted.

All of above prove that our protocol is secure. Table 2 shows the security comparisons between our proposed scheme and related schemes.

Table 2. Security comparisons

Security comparisons										
	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
[13]	Y	Y	Y	Y/--	--	--	Y	--	--	Y
[16]	Y	Y	Y	Y/--	--	--	--	Y	Y	Y
[17]	Y	Y	Y	Y/--	Y	--	Y	Y	Y	--
Our scheme	Y	Y	Y	Y/Y	Y	Y	Y	Y	Y	Y

Annotation : S1: Perfect forward secrecy; S2: Known-key secrecy;
S3: Mutual authentication and key agreement;
S4: Secure password and biometrics update protocol
S5:Server spoofing attack S6:Registration center spoofing attack
S7: Resist insider attack; S8: Resist impersonation attack;
S9: Resist man-in-the-middle attack; S10: Resist replay attack
--: Not mentioned or not involve Y/N: Support/Not support

5. **Functionality analysis.** In this subsection, Table 3 shows the functionality comparisons between our proposed scheme and related schemes about three aspects as below:

No timestamp mechanism

Timestamp is a string produced by the current time of communication entities which can replace the random numbers at some nodes with a nonce. Unfortunately, if the adversary delays delivery of the message, the interval time for message transferred is equal or greater than $\triangle T$, then the protocol will be stopped.

Privacy preserving

Usually, personal information of users is easy to leak. To solve this problem, our proposed scheme makes the sensitive information PW_i and ID_i hidden in a secure hash function, even if the message transferred over the insecure channel is intercepted by the adversary, he/she cannot gain any useful information from the intercepted hash function.

TABLE 3. Functionality comparisons

Functionality comparisons				
	F1	F2	F3	F4
[13]	N	N	N	N
[16]	N	N	N	Y
[17]	Y	Y	N	N
Our scheme	Y	Y	Y	Y

Annotation : F1: No timestamp mechanism; F2: Privacy preserving;
F3: Biometrics certification; F4: Multi-server environment
--: Not mentioned or not involve Y/N: Support/Not support

6. **Efficiency analysis.** The efficiency of the proposed scheme is analyzed in this subsection. According to the required operations for communication entities, Table 4 summarizes the communication costs of our proposed scheme and related schemes in different phases.

TABLE 4. Communication costs

Communication costs										
	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
[13]	--	--	1H	1S+1T	2T+2H+2S	2H+3S+3T	--	2H+1S+1T	3S+1T	--
[16]	--	2H	1H	4H	6H+2E	2H+1E	3E+3H	7H+1E	--	3H
[17]	--	--	1H	1E	2H+2E	2H+5E	--	2S+2H	2S+1E	--
Our scheme	--	1H	2H	1H	6H+3T	3T+5H	7H+4T	2H/2H/2H	2H/2H/2H	--/3H/3H

Annotation :C1/C2:Communication cost of the serve/registration center in the server registration phase;
C3/C4:Communication cost of the user/ server & registration center in the user registration phase;
C5/C6/C7: Communication cost of the user/server/registration center in the authenticated key agreement phase;
C8/C9/C10: Communication cost of the user/server/registration center in the biometrics and password update phase
H: Hashing operation; T: Chebyshev chaotic maps operation; S: Symmetric encryption/decryption
E: Elliptic curve multiplication --: Not involve the operations

Chang et al. [18] showed that the average time of one time hash function operation was 0.605ms. Lee et al. [19] showed that one hash function operation was about one time faster than one Chebyshev chaotic maps operation. Thus the average time of one Chebyshev chaotic maps operation was about 1.21ms.

According to Table 4, in our proposed scheme, we use 26 times hash function operations and 10 times Chebyshev chaotic maps operations at least, the execution time of our proposed protocol is about 27.83ms. According to Table 4, compared with related schemes, the execution of our proposed scheme is acceptable, and our proposed scheme is more practical.

7. **Conclusion.** In this paper, we propose a provable biometrics-based multi-server authenticated key agreement protocol with privacy preserving on chaotic maps cryptosystem. Our protocol only uses chaos theory to authentication communication entities, rather than encrypting/decrypting messages which can increase the efficiency of it. In addition, our protocol refuses timestamp, modular exponentiation and scalar multiplication on an elliptic curve, and provides secure biometric authentication, chaotic maps-based authenticated key agreement, secure update protocol and protects user privacy. In the same time,

the proposed protocol can satisfy various common security requirements. Compared with related schemes, the proposed scheme is more practical.

REFERENCES

[1] W.B. Lee and C.C. Chang, User identification and key distribution maintaining anonymity for distributed computer networks, *International Journal of Computer Systems Science and Engineering*, vol. 15, no. 4, pp. 211-214, 2000..

[2] W.J. Tsaur, A flexible user authentication scheme for multi-server Internet services, *NetworkingICN 2001: Lecture Notes in Computer Science*, vol. 2093, pp. 174-183, 2001.

[3] S. Kim, S. Lim and D. Won, Cryptanalysis of flexible remote password authentication scheme of ICN'01, *Electronics Letters*, vol. 38, no. 24, pp. 1519-1520, 2002.

[4] W.J. Tsaur, C.C. Wu and W.B. Lee, An enhanced user authentication scheme for multi-server Internet services, *Applied Mathematics & Computation*, vol. 170, no. 1, pp. 258-266, 2005.

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

[6] R.C. Wang, W.S. Juang and C.L. Lei, User authentication scheme with privacy-preservation for multi-server environment, *Communications Letters*, IEEE. vol. 13, no. 2, pp. 157-159, 2009.

[7] C.C. Chang and T.F. Cheng, A robust and efficient smart card based remote login mechanism for multi-server architecture, *International Journal of Innovative Computing Information and Control*, vol. 7, no. 8, pp. 4589-4602, 2011.

[8] K.H. Yeh, K.Y. Tsai and J.L. Hou, Analysis and design of a smart card based authentication protocol, *Journal of Zhejiang University Science C*, vol. 14, no. 12, pp. 909-917, 2011.

[9] B. Wang and M.D. Ma, A smart card based efficient and secured multi-server authentication scheme, *Wireless Personal Communications*, vol. 68, no. 2, pp. 361-378, 2012.

[10] D.B. He and S.H. Wu, Security flaws in a smart card based authentication scheme for multi-server environment, *Wireless Personal Communications*, vol. 68, no. 2, pp. 361-378, 2013.

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

[12] D.L. Guo and F.T. Wen, Analysis and improvement of a robust smart card based-authentication scheme for multi-server architecture, *Wireless Personal Communications*, ol. 78, no. 1, pp. 475-490, 2014.

[13] C. Guo and C.C. Chang, Chaotic maps-based password-authenticated key agreement using smart cards, *Communications in Nonlinear Science and Numerical Simulation*, vol. 18, no. 6, pp. 1433-1440, 2013.

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

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

[16] J.S. Zhang, J. Ma, X. Li and W.D. Wang, A secure and efficient robust user authentication scheme for multi-server environments using ECC, *Transactions on Internet and Information Systems*, vol. 8, no. 8, pp. 2930-2947, 2014.

[17] T.H. Liu, Q. Wang and H.F. Zhu, A multi-function password mutual authentication key agreement scheme with privacy preserving, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 5, no. 2, pp. 165-178, 2014.

[18] C.C. Chang and C.Y. Sun, A Secure and Efficient Authentication Scheme for E-coupon Systems, *Wireless Personal Communications*, vol.77, no.4, pp. 2981-2996, 2014.

[19] C.C Lee, A simple key agreement scheme based on chaotic maps for VSAT satellite communications,*International Journal of Satellite Communications and Networking*, vol. 31, no. 4, pp. 177-186, 2013.