Journal of Information Hiding and Multimedia Signal Processing ©2015 ISSN 2073-4212
Ubiquitous International Volume **6**, Number **1**, January **2015**

Robust Biometrics-based Key Agreement Scheme with Smart Cards towards a New Architecture

Hongfeng Zhu, Man Jiang, Xin Hao and Yan Zhang

Software College
Shenyang Normal University
No.253, HuangHe Bei Street, HuangGu District, Shenyang, P.C 110034 - China
zhuhongfeng1978@163.com; 459589817@qq.com; haoxin20110202@163.com; 1505733680@qq.com

Received June, 2014; revised August, 2014

Abstract. *In a traditional single server authentication scheme, if a user wishes to access network services from different servers, the user has to register with these servers separately. To handle this issue, multi-server authentication scheme has been proposed. Multi-server authenticated key agreement (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. However, MSAKA schemes are created with defects about the centralized registration center architecture. This architecture will make the centralized registration center become unsafe and have to deal with many registered and authenticated tasks. So the paper spares no effort to eliminate three problems: single-point of security, single-point of efficiency and single-point of failure. Based on these motivations, it is firstly proposed a new multiple servers to server architecture (MSTSA) to solve the problems caused by centralized registration center. Then a provably secure and robust biometrics-based Multiple Servers to Server authentication with key agreement scheme is presented using chaotic maps with smart cards. Security of the protocol is based on the computational infeasibility of solving Chaotic Maps-Based Discrete Logarithm problem (CMBDLP), Chaotic Maps-Based Diffie-Hellman problem (CMBDHP) and a secure symmetric encryption. At the same time the proposed scheme can not only refrain from consuming modular exponential computing and scalar multiplication on an elliptic curve, but is also robust to various attacks and achieves perfect forward secrecy with adjusting different server as a registration center for adapting to different users interests.*
Keywords: Authentication, Chaotic maps, Key agreement, Multiple servers to server architecture

1. **Introduction.** With the rapid development of chaos theory related to cryptography [13], many key agreement protocols using a chaotic map have been studied widely. These protocols using a chaotic map can mainly be divided into three categories based on the number of participants: two-party authenticated key agreement protocols [723], three-party authenticated key agreement protocols [2431], and N-party authenticated key agreement protocols. Furthermore, based on the respective features in detail, the previous researches [731] on this subject can be classified many categories: such as password-based, using smart card, timestamp, anonymity and other security attributes. From the macroscopic point of view, the literatures [731] have two main traits: On one side, along with some new protocols putting forward, then some flaws will be found over a period of time, such as the flaws in the literatures [7, 9, 14] are found by the literatures [8, 10, 15]. On the other side, some new secure attributes and improving the efficiency can be found in the

literatures [12, 13, 17, 19]. In recent years, the three-party password-authenticated key agreement protocol using modular exponentiation or scalar multiplication on an elliptic curve has been addressed widely [29, 30]. However, these schemes need heavy computation costs and even the most recent literatures are still remain on three-party authenticated key agreement protocol [31]. Multi-server authenticated key agreement (MSAKA) protocols aim to register at the registration center for log in other servers without register repeatedly. MSAKA protocols mainly want to solve the problems in a traditional single server with authentication schemes [32-39] which lead to the fact that user has to register to different servers separately. On a macro level MSAKA protocols can be divided into three phases in chronological order:

(1) the creative phase: The pioneer work in the field was proposed by Li et al. [41] in 2001. However, Lin et al. [42] pointed out that Li *et al.*,'s scheme takes long time to train neural networks and an improved scheme based on ElGamal digital signature and geometric properties on the Euclidean plane has also been given.

(2) the development phase: the main work in this phase is amended repeatedly. For example, Tsai [43] also proposed an efficient multi-server authentication scheme based on one-way hash function without a verification table. Because Tsai's scheme only uses the nonce and one-way hash function, the problems associated with the cost of computation can be avoided in the distributed network environment. However, some researchers [44] pointed out that Tsai's scheme is also vulnerable to server spoofing attacks by an insider server and privileged insider attacks, and does not provide forward secrecy.

(3) the diversification phase: the research emphasis shifts to functionality. Therefore, identity-based MSAKA protocols, based on bilinear pairings or elliptic curve cryptosystem (ECC) MSAKA protocols, dynamic identity-based MSAKA protocols and other MSAKA protocols came up recently[44-46].

In this paper, a new flexible and password-authenticated key agreement scheme is proposed based on chaotic maps for multiple servers to server architecture. The main contributions are shown below:

(1) New architecture: The paper firstly presents the multiple servers to server architecture which can solve the weaknesses in the traditional multi-server communication architecture. That is also the first fundamental solution to transfer centralized registration center to distributed registration center. Furthermore, in the proposed architecture, a new solution of flexible and password-authenticated key agreement scheme is proposed based on chaotic maps.

(2) Computation: The proposed protocol is based on chaotic maps without using modular exponentiation and scalar multiplication on an elliptic curve.

(3) Security: The protocol can resist all common attacks, such as impersonation attacks, man-in-the-middle attacks, etc. (The details specified in section 2.5 of the article)

(4) Functionality: The protocol also has achieved some well-known properties, such as perfect forward secrecy and execution efficiency. Furthermore, the paper firstly presents the special properties about symmetry and transparency which will be set up in the proposed scheme. (The details specified in section 2.6 of the article).

The rest of the paper is organized as follows: Some preliminaries are given in Section 2. Next, A Chebyshev chaotic maps-based multiple servers to server scheme 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.

2. **Preliminaries.**

2.1. **Multiple servers to server architecture.** In the multi-server environment, 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, multi-server architecture schemes have been developed and proposed [41-46]. Basically, 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. Fig.1 shows the traditional multi-server environment. In the proposed

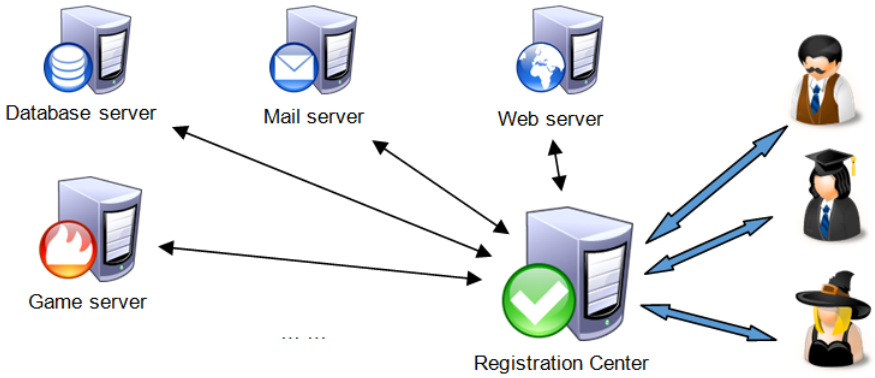

FIGURE 1. The traditional multi-server communication architecture

multiple servers to server communication architecture, the registration center is not fixed. In other words, any server can work as a registration center. However in multi-server authentication architecture, the single registration center will face to single-point of security, single-point of efficiency and single-point of failure problems. The proposed architecture can solve the problems under multi-server environment with only one registration center architecture, which means 'once security register for all registration' that is shown in Fig.2.

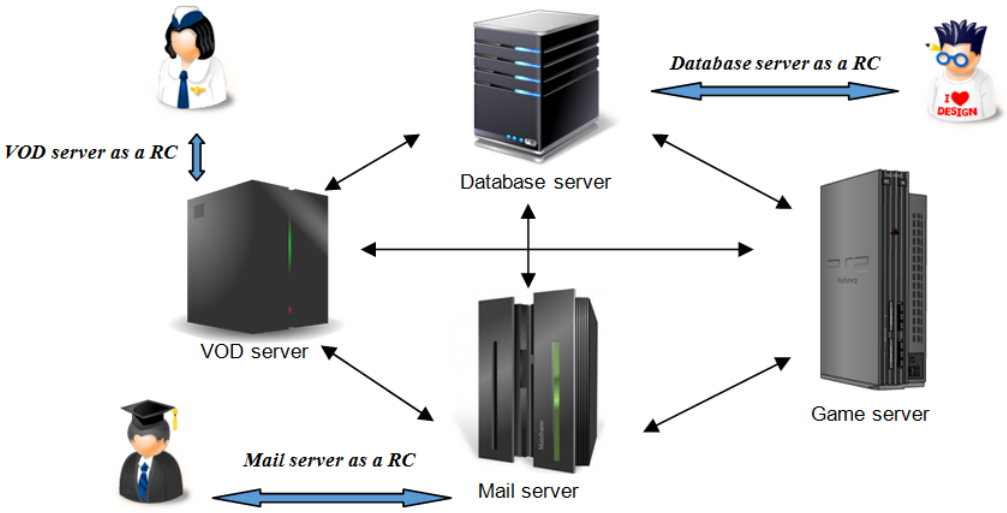

FIGURE 2. The proposed multiple servers to server communication architecture

2.2. **Biometric authentication.** Each user has their unique biometric characteristics, for example, voice, fingerprints, iris recognition. They have irreplaceable advantages: reliability, availability, non-repudiation, less cost and so on. Therefore, biometric authentication has widely used. Fig.3 is the flow diagram of biometric characteristics collection and authentication. During the biometric collection phase, a biometric sample is collected, processed by a computer, and stored which prepared for subsequent comparison (Fig.3). During the biometric authentication phase, the biometric system compares the stored sample with a newly captured sample (Fig.3). Obviously, smart card has powerful information confidentiality and flexible portability. When performing a biometric authentication process, a user inputs a smart card, and makes use of a simple touch with a finger or a glance at a camera to authenticate himself/herself [4-6].

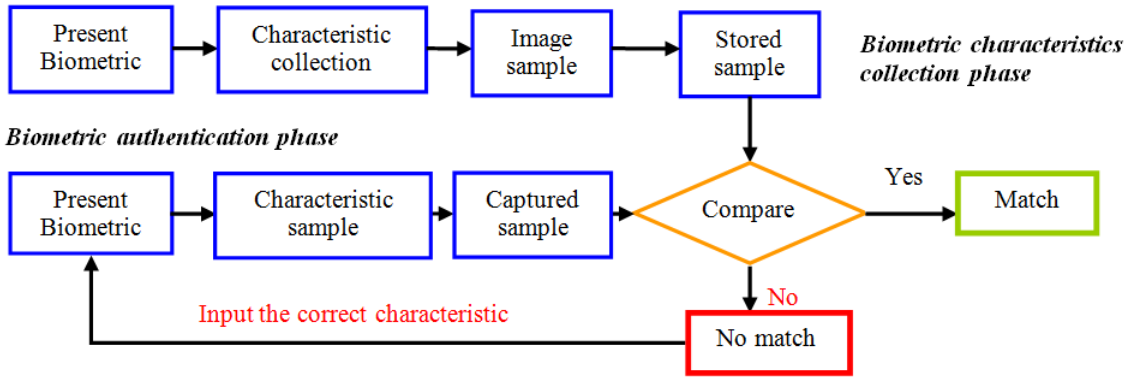

FIGURE 3. The flow diagram of Biometric characteristics collection and authentication

2.3. **Definition and properties of Chebyshev chaotic maps.** 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 $T_n(x) = \cos(n \arccos(x))$. Chebyshev polynomial map $T_n : R \to R$ of degree n is defined using the following recurrent relation [24]:

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

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

The first few Chebyshev polynomials are:

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

One of the most important properties is that Chebyshev polynomials are the so-called semi-group property which establishes that

$$T_r(T_s(x)) = T_{r \cdot s}(x) \tag{2}$$

An immediate consequence of this property is that Chebyshev polynomials commute under composition

$$T_r(T_s(x)) = T_s(T_r(x)) \tag{3}$$

In order to enhance the security, Zhang [48] proved that semi-group property holds for Chebyshev polynomials defined on interval $(-\infty, +\infty)$. The enhanced Chebyshev polynomials are used in the proposed protocol:

$$T_n(x) = (2xT_{n-1}(x) - T_{n-2}(x)) (\mathrm{mod} N) \tag{4}$$

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

$$T_{r \cdot s}(x) = T_r(T_s(x)) = T_s(T_r(x)) \tag{5}$$

Definition 1. Semi-group property of Chebyshev polynomials:

$$T_r(T_s(x)) = \cos(r\cos^{-1}(s\cos^{-1}(x))) = \cos(rs\cos^{-1}(x)) = T_{sr}(x) = T_s(T_r(x))$$

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).

2.4. **One-way Hash Function.** A secure cryptographic one-way hash function $h : a \to b$ has four main properties:

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

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

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

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

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

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

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

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

2.6. **Security requirements.** Secure communication schemes for remote mutual authentication and session key agreement for the multiple servers to server architecture should provide security requirements [41-46], such as mutual authentication and key agreement, impersonation attack, man-in-the-middle attack, replay attack, known-key security, perfect forward secrecy, data integrity, off-line guessing attack, session key security and key compromise impersonation. The definitions and proofs of above-mentioned security requirements will be illustrated in Appendix A. detailedly.

2.7. **Function requirements.** The steps as follows:

(1) Symmetry: It is symmetric design for all the servers in the proposed scheme. In other words, any server can work as a registration center. Furthermore, any user can regard the registered server as the registration center to log in the other servers without multiple registration. That is very convenient from the perspective of the client.

(2) Transparency: Transparency is defined as users need not know the operating mechanism of the servers. For example, the user has a Facebook account already: when a user log in another servers (VOD server, Game server, Database server and so on), the user only need to click the button of using Facebook account to log in.

(3) Simplicity: The user only needs to register at a certain server according to his favorite, and then treat this server as the registration center to access all the permitted services provided by the eligible servers.

(4) Expandability: In the multiple servers to server architecture, the servers are under the dynamic changing. In other words, any servers can join or leave easily.

(5) Other functions: The protocol should achieve some well-known properties, such as, no timestamp, and execution efficiency.

3. **The Proposed Chaotic Maps-based Scheme with Multiple Servers to Server.** In this section, under the multiple servers to server architecture, a chaotic maps-based password-authenticated key agreement scheme is proposed which consists of four phases: the multiple servers to server architecture setup phase, the user registration phase, authenticated key agreement phase and password and shared secret key among servers update phase. But firstly some notations are given which used in the proposed scheme.

3.1. **Notations.** In this phase, any participant i has its identity ID_i , and public key $(x, T_{k_i}(x))$ and a secret key k_i based on Chebyshev chaotic maps, a secure one-way hash function $H(\cdot)$[49], and a pair of secure symmetric encryption/decryption functions $E_K()/D_K()$ with key K. The concrete notation used hereafter is shown in Table1.

TABLE 1. Notations

Symbol	Definition
$ID_A \quad pw_A$	The identity of user, the password of user, respectively
S_j, ID_{S_j}	The jth server， the identity of the jth server, respectively
a, r_i, r_j	nonces
$(x, T_{k_i}(x))$	public key based on Chebyshev chaotic maps
k_i	secret key based on Chebyshev chaotic maps
RC	registration center (any server can be regarded as a RC)
$E_K() / D_K()$	a pair of secure symmetric encryption/decryption functions with the key K
B	the biometric sample of user
τ	predetermined threshold for biometric verification
$d()$	symmetric parametric function
H	A secure one-way hash function
$\|$	concatenation operation

3.2. **Multiple servers to server architecture setup phase.** Simply speaking, for all the servers $S_i(1 \le i \le n)$ shown in Fig.4., their public keys are $(x, T_{k_i}(x))(1 \le i \le n)$ and the corresponding secret keys are $k_i(1 \le i \le n)$.The setting mainly has two advantages:

(1) It is symmetric for all the servers based on chaotic maps which are suitable for our proposed scheme.

(2) It is expandable because any server can join or leave easily. Remark: It has to be emphasized that all the servers must be verified by the authorities before they provide services for users.

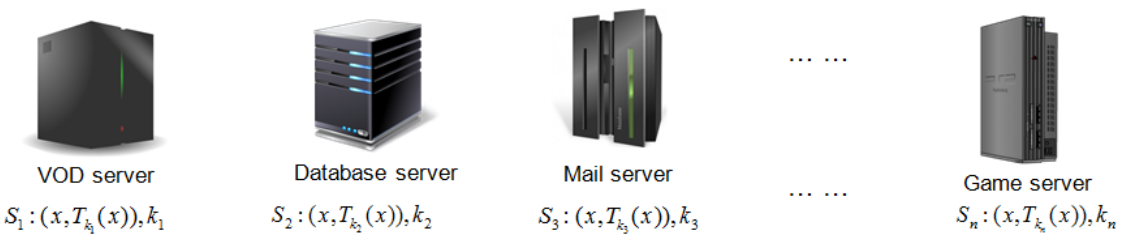

FIGURE 4. Multiple servers to server architecture setup phase

3.3. **User registration phase.** Concerning the fact that the proposed scheme mainly relies on the design of Chebyshev chaotic maps-based in multiple servers to server architecture, it is assumed that the user can register at his appointed server as the registration center in some secure way or by secure channel. The same assumption can be set up for servers Fig.5 illustrates the user registration phase. The steps are performed during the user registration phase as follows.

Step 1. When a user Alice wants to be a new legal user, she chooses her identityID_A, password pw_A at liberty, and also inputs her personal biometric image sample B at the sensor. Then Alice submits $\{ID_A, h(pw_A||B), B\}$ to server Sj as the RC via a secure channel.

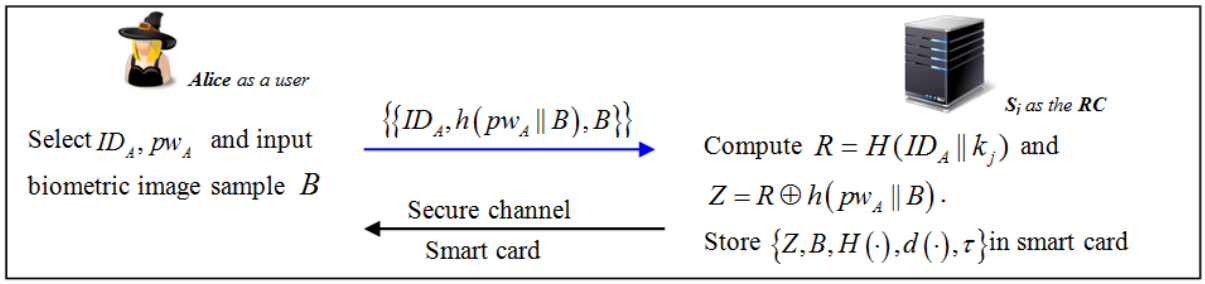

FIGURE 5. User registration phase

Step 2. After receiving the message $m_1 = \{ID_A, T_a(x), C_1\}$ from Alice, S_i will do the following tasks to ask S_j for helping to authenticated Alice: S_i selects random r_i and computes $T_{r_i}(x)$, $K_{S_iS_j} = T_{r_i}T_{k_j}(x)$, $H_{S_i} = H(ID_A||ID_{S_i}||T_{r_i}(x)||m_1)$ and $C_2 = E_{K_{S_iS_j}}(ID_A||ID_{S_i}||m_1||H_{S_i})$. And then sends the message $m_2 = \{ID_{S_i}, T_{r_i}(x), C_2\}$ to S_j.

Step 3. Next, S_j will help Alice and S_i to authenticate each other and verify the temporary information by helping them to compute the session key. After receiving the message $m_2 = \{ID_{S_i}, T_{r_i}(x), C_2\}$, S_j will Compute $K_{S_jS_i} = T_{k_j}T_{r_i}(x)$ to decrypt C_2. After getting the information in C_2, S_j can compute $H'_{S_i} = H(ID_A||ID_{S_i}||T_{r_i}(x)||m_1)$ and $C'_1 = H(H(ID_A||k_j)||T_a(x))$ to check if $H_{S_i} = H'_{S_i}$ and $C'_1 = C_1$. If above equations hold, that means Alice and Si are legal participants in this instance. Then S_j selects random r_j and computes $T_{r_j}(x)$, $K'_{S_jS_i} = T_{r_j}T_{k_i}(x)$, $C_4 = H(H(ID_A||k_j)||ID_A||ID_{S_i}||T_a(x)||T_{r_i}(x))$, $H_{S_j} = H(ID_A||ID_{S_i}||T_a(x)||T_{r_i}(x)||C_4)$, $C_3 = E_{K'_{S_jS_i}}(ID_A||ID_{S_i}||T_a(x)||T_{r_i}(x)||H_{S_j}||C_4)$ and sends the message $m_3 = \{ID_{S_j}, T_{r_j}(x), C_3\}$ to S_i. If not, S_j terminates it simply.

Step 4. After receiving the message $m_3 = \{ID_{S_j}, T_{r_j}(x), C_3\}$, Si firstly computes $K'_{S_iS_j} = T_{k_i}T_{r_j}(x)$ and uses $K'_{S_iS_j}$ to decrypt C_3 for getting the messages $ID_A||ID_{S_i}||T_a(x)||T_{r_i}(x)||H_{S_j}||C_4$. Then Si computes $H'_{S_j} = H(ID_A||ID_{S_i}||T_a(x)||T_{r_i}(x)||C_4)$. Check if $H'_{S_j} = H_{S_j}$. If the equation does not hold, S_i terminates it. Otherwise, that means S_j is authenticated and the information of user Alice is also authenticated messages. Finally S_i computes $SK = T_{r_i}T_a(x)$, $C_5 = H(C_4||SK)$ and sends $m_4 = \{ID_{S_i}, T_{r_i}(x), C_5\}$ to Alice.

Step 5. After receiving the message $m_4 = \{ID_{S_i}, T_{r_i}(x), C_5\}$ from S_i, Alice can use her secret $H(ID_A||k_j)$ to compute C4 and use her nonce a to compute $SK = T_aT_{r_i}(x)$.

Alice computes $C_5' = H(C_4||SK)$ and checks if $C_5' = C_5$. If any of above equations does not hold, Alice terminates it. Otherwise, Alice authenticates both S_i band S_j. Finally Alice computes $C_6 = H(C_4||SK||T_{r_i}(x))$ and sends $m_5 = \{C_6\}$ to S_i.

Step 6. When S_i obtains m_5, S_i computes $C_6' = H(C_4||SK||T_{r_i}(x))$ and verifies whether $C_6' = C_6$ or not. If it does not hold, S_i terminates it. Otherwise, Alice and S_i share the session key $SK = T_a T_{r_i}(x) = T_{r_i} T_a(x)$.

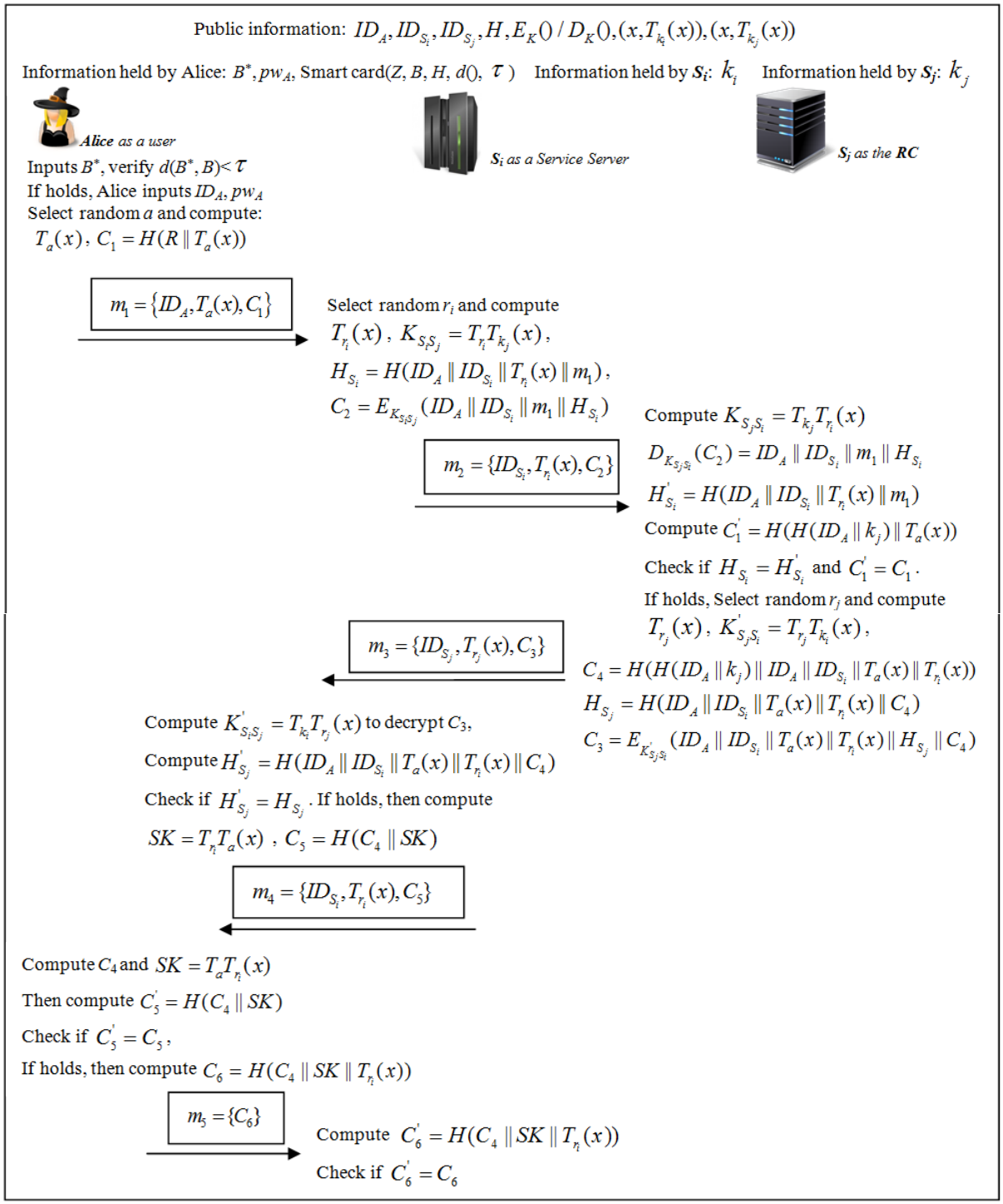

FIGURE 6. Authenticated key exchange phase

3.4. **Password and biometrics update phase**

. Fig.7 illustrates biometrics and password update phase. The steps are performed during the biometrics and password update phase as follows.

Step 1. Alice inputs the smart card into a card reader, opens the update application software, and imprints biometrics B^{new} at the sensor.

Step 2. Firstly, the biometrics authentication process compares B^{new} with B. If $d(B^{new}, B) \geq \tau$, that means Alice will get a connection refused response. If $d(B^*, B) < \tau$, that means Alice will get a connection accepted response. Then the smart card sends the password input request message to Alice.

Step 3. Alice inputs the old password pw_A and the new password pw_A^{new}.

Step 4. Smart card computes $Z^{new} = Z \oplus H(pw_A||B) \oplus H(pw_A^{new}||B^{new})$, and then replaces Z and by Z^{new} and B^{new}, and then stores Z^{new} and B^{new} into the smart card.

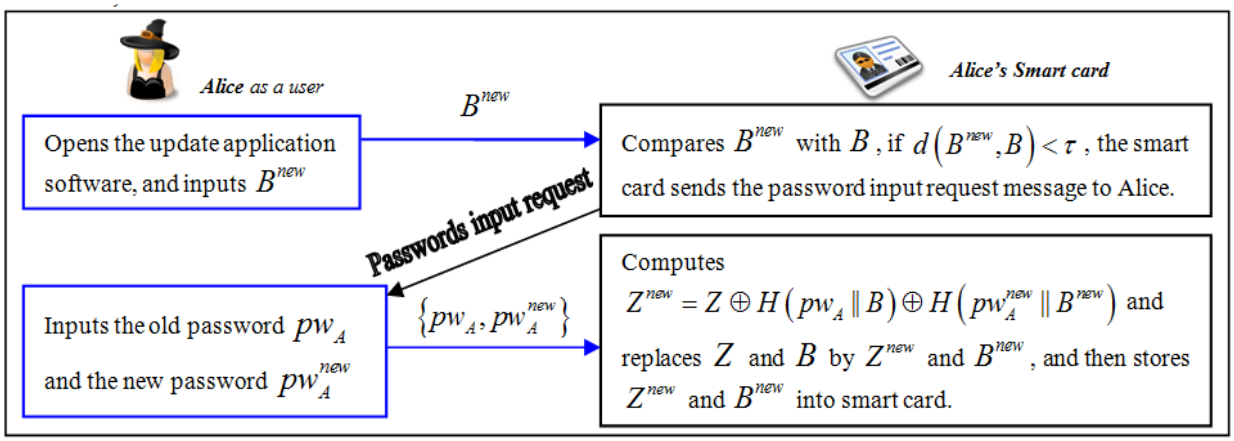

FIGURE 7. The biometric and password update phase

4. **Security Consideration**

. 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, a secure one-way hash function, and a secure symmetric encryption. Assume that the adversary has full control over the insecure channel including eavesdropping, recording, intercepting, modifying the transmitted messages. The definitions and analysis of the security requirements will be illustrated in Appendix A, and the provable security will be given in Appendix B. From Table 2, we can see that the proposed scheme not only provides secure session key agreement and perfect forward secrecy, but also prevents the KCI attacks. As a result, the proposed scheme is more secure and has much functionality compared with the recent related scheme.

TABLE 2. Architecture and security of our proposed protocol

Category		Eun-Jun Yoon's Scheme [46] (2013)	Our Proposed Scheme
Architecture		Multi-server communication architecture (Centralized)	Multiple servers to server communication architecture (Distributed)
Architecture properties and functionality	Single-point of security	N/A	**Provided**
	Single-point of efficiency	N/A	**Provided**
	Single-point of failure	N/A	**Provided**
	Symmetry	N/A	**Provided**
	Transparency	**	***
	Simplicity	*	***
	Expandability	**	***
	No timestamp	Provided	Provided
	Secure password update	Provided	Provided
Security requirements	No verification table	Provided	Provided
	Single registration	Provided	Provided
	Mutual authentication	Provided	Provided
	Impersonation attack	Provided	Provided
	Man-in-the-middle attack	Provided	Provided
	Replay attack	Provided	Provided
	Known-key security	Provided	Provided
	Perfect forward secrecy	Provided	Provided
	Data integrity	Provided	Provided
	Guessing attacks	Provided	Provided
	Session key security	Provided	Provided
	KCI attacks	N/A	**Provided**
	Stolen smart card attack	Provided	Provided
	Biometrics authentication	Provided	Provided

Notes:
N/A: "not available" or "not support".
*: provided but in low level.
**: provided but in middle level.
***: provided but in high level.

5. **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, Xiao *et al.*, [9] and Wang [24] proposed several methods to solve the Chebyshev polynomial computation problem.

In this section, we will compare our scheme with Eun-Jun Yoon's scheme of [46]. 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_{ECmul}: The time for executing the elliptic curve point multiplication;

T_{CHpol}: The time for executing the $T_{n(x)}$ mod p in Chebyshev polynomial using the algorithm in literature[52].

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 [52-54]. Moreover, the computational cost of XOR operation could be ignored when compared with other operations. Table 3 shows performance comparisons between our proposed scheme and Eun-Jun Yoons scheme of [46]. Therefore, as in Table 3, we can draw a conclusion that the proposed scheme has the lowest computational costs and is well suited to the smart card's applications.

TABLE 3. Efficiency of our proposed scheme

Communication costs		Eun-Jun Yoon's Scheme [46] (2013)	Our Proposed Scheme
User registration phase		$1T_{hash} \approx 0.0005s$	$1T_{hash} \approx 0.0005s$
Server registration phase		$1T_{hash} \approx 0.0005s$	-
Authentication phase	*User*	$5T_{hash}+2T_{EC-mul} \approx 0.12715s$	$4T_{hash}+1T_{CH-pol} \approx 0.02302s$
	S_i (server)	$5T_{hash}+2T_{EC-mul} \approx 0.12715s$	$3T_{hash}+2T_{sym}+2T_{CH-pol} \approx 0.06094s$
	S_j (*RC*)	$7T_{hash} \approx 0.0035s$	$5T_{hash}+2T_{sym}+2T_{CH-pol} \approx 0.06194s$
Password and biometrics update phase		$2T_{hash} \approx 0.001s$	$2T_{hash} \approx 0.001s$
Random numbers		2	3
Rounds of Authentication phase		5	5
Note: -: "no need".			

6. **Conclusions.** The article put forward a new architecture called multiple servers to server (MSTSA) to solve the problems caused by centralized registration center in traditional multi-server communication architecture. In MSTSA architecture, the paper proposes the first provably secure and flexible password-authenticated key agreement scheme based on chaotic maps which is a better algorithm than RSA and ECC. The core ideas of the proposed scheme are the symmetry (or called peer to peer) in the servers side and the transparency for the clients side. Our proposed scheme has satisfactory security, efficiency and functionality. Therefore, our protocol is more suitable for practical applications. Next, the proposed protocol in three aspects will be extended: *(1)* From the view of functionality, it is meaningful to research the fairness or entanglement and so on. *(2)* From the perspective of complex, diversified algorithms, especially for quantum security, are our interests.

References

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

[2] I. Hussain, T. Shah, M. Gondal, H. Mahmood, An efficient approach for the construction of LFT S-boxes using chaotic logistic map, *Nonlinear Dyn.*, vol. 71, pp. 133-140, 2013.

[3] M. Khan, T. Shah, H. Mahmood, M. Gondal, An efficient method for the construction of block cipher with multi-chaotic systems, *Nonlinear Dyn*, vol. 71, pp. 489492, 2013.

[4] N. Y. Lee, Y. C. Chiu, Improved remote authentication scheme with smart card, *Computer Standards and Interfaces*, vol. 27, no. 2, pp. 177-180, 2005.

[5] MK. Khan, JS. Zhang, Improving the security of a flexible biometrics remote user authentication scheme, *Computer Standards and Interfaces*, vol. 29, no. 1, pp. 82-85, 2007.

[6] CT. Li, MS. Hwang, An efficient biometrics-based remote user authentication scheme using smart cards, *Journal of Network and Computer Applications*, vol. 33, no. 1, pp. 1-5, 2010.

[7] F. zkaynak, S. Yavuz, Designing chaotic S-boxes based on time-delay chaotic system, *Nonlinear Dyn.*, doi:10.1007/s11071-013-0987-4, 2013.

[8] G. Alvarez, Security problems with a chaos-based deniable authentication scheme, *Chaos Solitons Fractals26*,vol. 7, no. 11, 2005.

[9] D. Xiao, X. Liao, S. Deng, A novel key agreement protocol based on chaotic maps, *Inf. Sci*, vol. 177, pp. 11361142, 2007.

[10] S. Han, Security of a key agreement protocol based on chaotic maps, *Chaos Solitons Fractals38*, pp. 764768, 2008.

[11] T. Xiang, K. Wong, X. Liao, On the security of a novel key agreement protocol based on chaotic maps, *Chaos Solitons Fractals40*, pp. 672675, 2009.
[12] D. Xiao, X. Liao, S. Deng, Using time-stamp to improve the security of a chaotic maps-based key agreement protocol, *Inf. Sci*, vol. 178, pp. 159811602, 2008.
[13] S. Han, E. Chang, Chaotic map based key agreement without clock synchronization, *Chaos Solitons Fractals39*, pp. 12831289, 2009.
[14] X. Guo, J. Zhang, Secure group key agreement protocol based on chaotic Hash, *Inf. Sci*, vol. 180, pp. 40694074, 2010.
[15] D. He, Cryptanalysis of a key agreement protocol based on chaotic Hash.eprint.iacr.org/2011/333.pdf.
[16] P. Gong, P. Li, W. Shi, A secure chaotic maps-based key agreement protocol without using smart cards, *Nonlinear Dyn.*, vol. 70, pp.24012406, 2012.
[17] H. Tseng, R. Jan, W. Yang, A chaotic maps-based key agreement protocol that preserves user anonymity, *IEEE International Conference on Communications (ICC09)*, pp. 16, 2009.
[18] Y. Niu, X. Wang, An anonymous key agreement protocol based on chaotic maps, *Commun. Nonlinear Sci. Numer. Simul.*, vol. 16, no. 4, pp. 19861992 (2011).
[19] K. Xue, P. Hong P, Security improvement on an anonymous key agreement protocol based on chaotic maps, *Commun. Nonlinear Sci. Numer. Simul.*, vol. 17, pp. 29692977, 2012.
[20] E. Yoon, Efficiency and security problems of anonymous key agreement protocol based on chaotic maps, *Commun. Nonlinear Sci. Numer. Simul.*, vol. 17, pp. 27352740, 2012.
[21] Z. Tan, A chaotic maps-based authenticated key agreement protocol with strong anonymity, *Nonlinear Dyn.*, vol. 72, pp. 311320, 2013.
[22] C. Lee, C. Hsu, A secure biometric-based remote user authentication with key agreement scheme using extended chaotic maps, *Nonlinear Dyn.*, vol. 71, pp. 201211, 2013.
[23] C. Guo, CC. Chang, Chaotic maps-based password-authenticated key agreement using smart cards, *Commun. Nonlinear Sci. Numer. Simul.*, doi:10.1016/j.cnsns.2012.09.032, 2012.
[24] X. Wang, J. Zhao, An improved key agreement protocol based on chaos, *Commun. Nonlinear Sci. Numer. Simul.*, vol. 15, pp. 40524057, 2010.
[25] E. Yoon, I. Jeon, An efficient and secure DiffieHellman key agreement protocol based on Chebyshev chaotic map, *Commun. Nonlinear Sci. Numer. Simul.*, vol. 16, pp. 23832389, 2011.
[26] H. Lai, J. Xiao, L. Li, Y. Yang, Applying semigroup property of enhanced Chebyshev polynomials to anonymous authentication protocol, *Math. Probl. Eng.*, doi:10.1155/2012/454823, 2012.
[27] F. Zhao, P. Gong, S. Li, M. Li, and P. Li, Cryptanalysis and improvement of a three-party key agreement protocol using enhanced Chebyshev polynomials, *Nonlinear Dyn.*, 2013.
[28] C. Lee, C. Li, C. Hsu, A three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps, *Nonlinear Dyn.*, vol. 73, pp. 125132, 2013.
[29] J. Yang, T. Cao, Provably secure three-party password authenticated key exchange protocol in the standard model, *J. Syst. Softw.*, vol. 85, pp. 340350, 2012.
[30] S. Wu, K. Chen, Q. Pu, and Y. Zhu, Cryptanalysis and enhancements of efficient three-party password-based key exchange scheme, *Int. J. Commun. Syst.*, doi:10.1002/dac.1362, 2012.
[31] Q. Xie, J. Zhao, X. Yu, Chaotic maps-based three-party password-authenticated key agreement scheme, *Nonlinear Dyn*, 74:10211027. DOI 10.1007/s11071-013-1020-7, 2013.
[32] L. Lamport, Password authentication with insecure communication, *Commun ACM*, vol. 24, no. 11, pp. 770772, 1981.
[33] T. Hwang, Y. Chen, CS. Laih, Non-interactive password authentication without password tables, *Proc of IEEE region conference on computer and communication system*, pp. 429431, 1990.
[34] HM. Sun, An efficient remote use authentication scheme using smart cards, *IEEE Trans Consum Electron*, vol. 46, no. 4, pp. 958961, 2000.
[35] CH. Lin, YY. Lai, A flexible biometrics remote user authentication scheme, *Comput. Stand Interfaces*, vol. 27, no. 1, pp. 1923, 2004.
[36] NY. Lee, YC, Chiu YC, Improved remote authentication scheme with smart card, *Comput Stand Interfaces*, vol. 27, no.2, pp. 177180, 2005.
[37] EJ. Yoon, EK. Ryu, KY. Yoo, An improvement of HwangLeeTangs simple remote user authentication scheme, *Comput. Secur.*, vol. 24, no. 1, pp. 5056, 2005.
[38] YF. Chang, CC. Chang, YW. Su, A secure improvement on the user-friendly remote authentication scheme with no time concurrency mechanism, *Proc. of 20th international conference on advanced information networking and applications (AINA06)*, IEEE Computer Society, Los Alamitos, pp. 741745, 2006.

[39] MK. Khan, J. Zhang, Improving the security of a flexible biometrics remote user authentication scheme, *Comput. Stand Interfaces*, vol. 29, no.1, pp. 8285, 2007.
[40] A. Stolbunov, Reductionist security arguments for public-key cryptographic schemes based on group action, *The Norwegian Information Security Conference (NISK)*, pp. 97-109, 2009.
[41] LH. Li LH, IC. Lin, and MS. Hwang, A remote password authentication scheme for multi-server architecture using neural networks, *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 14981504, 2001.
[42] IC. Lin, MS. Hwang, and LH. Li, A new remote user authentication scheme for multi-server architecture, *Future Generation Computer Systems*, vol. 19, no. 1, pp. 1322, 2003.
[43] JL. Tsai, Efficient multi-server authentication scheme based on one-way hash function without verification table, *Comput. Secur.*, vol. 27, no. 34, pp. 115121, 2008.
[44] SP. Ravi, CD. Jaidhar, T. Shashikala, Robust Smart Card Authentication Scheme for Multi-server Architecture, *Wireless Pers Commun.*, DOI 10.1007/s11277-013-1039-6, vol. 72, pp.729745, 2013.
[45] B. Wang, M. Ma, A smart card based efficient and secured multi-server authentication scheme, *Wireless Personal Communications*, doi:10.1007/s11277-011-0456-7, 2012.
[46] E. J. Yoon, K. Y. Yoo, Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem, *J. Supercomput*, vol. 63, pp. 235255, 2013.
[47] J. Katz, JS. Shin, Modeling insider attacks on group key-exchange protocols, *Proceedings of the 12th ACM Conference on Computer and Communications SecurityCCS05*, ACM. pp. 180189, 2005.
[48] Zhang L. Cryptanalysis of the public key encryption based on multiple chaotic systems. Chaos Solitons Fractals37(3), 669674 (2008).
[49] Xiao D, Shih F, and Liao X. A chaos-based hash function with both modification detection and localization capabilities, *Commun. Nonlinear Sci. Numer. Simul.*, vol. 15, pp. 22542261, 2010.
[50] J. Kar, B. Majhi, An Efficient Password Security of Three Party Key Exchange Protocol based on ECDLP, *12th International Conference on Information Technology 2009 (ICIT 2009)*, Bhubaneswar, India, Tata McGrow Hill Education Private Limited, pp. 75-78, 2009.
[51] C. Ran and K. Hugo, Analysis of key-exchange protocols and their use for building secure channels, In Birgit Ptzmann, editor, EUROCRYPT, *Lecture Notes in Computer Science*, vol. 2045, pp. 453-474, Springer, 2001.
[52] L. Kocarev, S. Lian, Chaos-Based Cryptography: Theory, Algorithms and Applications, Springer, Berlin, pp. 5354, 2011.
[53] W. Hsieh, J. Leu, Anonymous authentication protocol based on elliptic curve DiffieHellman for wireless access networks, *Wireless Communications and Mobile Computing*, doi:10.1002/wcm.2252, 2012.
[54] C. Li, M. Hwang, and Y. Chung, A secure and efficient communication scheme with authenticated key establishment and privacy preserving for vehicular ad hoc networks, *Computer Communication*, vol. 31, pp. 28032814, 2008.

APPENDIX A. Security proof of the proposed scheme

(1) Mutual authentication and key agreement

Definition A.1. Mutual authentication and key agreement refers to two parties authenticating each other suitably and getting the session key simultaneously.

Theorem A.1. *The proposed protocol can achieve mutual authentication and key agreement.*

Proof: S_j uses the secret key k_j to compute $C_4$and S_i uses the secret key k_i to compute SK, so the proposed scheme allows the Alice to authenticate the S_i and S_j simultaneously by checking whether $H(C_4||SK)$ equals C^5. S_i and S_j authenticate each other by opposite sides public key $(x, T_{k_j}(x))$ and $(x, T_{k_i}(x))$, because only using the secrets key k_i or k_j to decrypt the message C_3 or C_4.

If $H(H(ID_A||k_j)||T_a(x))$ equals C_1, which means that Alice was already authenticated by S_j. Because only after Alice was authenticated by inputting her personal biometric image sample B and pw_A, the smart card can compute the messages $H(ID_A||k_j)$ and $H(H(ID_A||k_j)||T_a(x))$.

If H'_{S_j} equals H_{S_j}, which also means that the information $T_a(x)$ of Alice was already authenticated by S_j. Because H_{S_j} contains $ID_A||ID_{S_i}||T_a(x)||T_{r_i}(x)||C_4$ and C_4 contains $H(ID_A||k_j)$. The trust flow is $S_i \rightarrow S_j \rightarrow$ Alice.

As for the key agreement, after authenticating each other, the temporary $T_a(x)$, $T_{S_r}(x)$ and the ID ID_A, ID_{S_i} were already authenticated by S_j. So finally Alice and S_i can make the key agreement simultaneously.

(2) Impersonation attack / Man-in-the-middle attack

Definition A.2. 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.

Definition A.3. 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.

Theorem A.2. *The proposed protocol can resist impersonation attack.*

Theorem A.3. *The proposed protocol can resist Man-in-the-middle attack.*

Proof: An adversary cannot impersonate anyone of the Alice, S_i and S_j. The proposed scheme has already authenticated each other among Alice, $S_i$and S_j (in section Appendix A.(1)) based on the secrets k_i, k_j, B, pw_A and the nonces a, r_i, r_j. So there is no way for an adversary to have a chance to carry out impersonation attack.

Because $C_i(1 \leq i \leq 6)$ contain the participants identities, a man-in-the-middle attack cannot succeed.

(3)Replay attack

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

Theorem A.4 *The proposed protocol can resist replay attack.*

Proof: An adversary cannot start a replay attack against our scheme because of the freshness of a, r_i, r_j in each session. If $T_a(x)$, $T_{r_i}(x)$ and $T_{r_j}(x)$ has appeared before or the status shows in process, any of the participants in instance protocol will reject the session request. If the adversary wants to launch the replay attack successfully, it must compute and modify $T_a(x)$, $T_{r_i}(x)$, $T_{r_j}(x)$ and $C_i(1 \leq i \leq 6)$ correctly which is impossible.

(4) Known-key security

Definition A.5 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.

Theorem A.5 The proposed protocol can achieve known-key security.

Proof: Since the session key $SK = T_a T_{r_i}(x) = T_{r_i} T_a(x)$ 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. And in the secrets update phase, any session key is only used once, so it has known-key security attribute.

(5) Perfect forward secrecy

Definition A.6 An authenticated multiple key establishment protocol provides perfect forward secrecy if the compromise of both of the nodes secret keys cannot results in the compromise of previously established session keys [50].

Theorem A.6 The proposed protocol can achieve perfect forward secrecy.

Proof: In the proposed scheme, the session key $SK = T_a T_{r_i}(x) = T_{r_i} T_a(x)$ 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_i, pw_A) of each of participants. Furthermore because of the intractability of the CMBDLP and CMBDHP problem, an adversary cannot compute the previously established session keys.

(6) Data integrity

Definition A.7 Authentication multiple key establishment protocol is said to achieve the property of data integrity, if there is no polynomial time algorithm that can alter or manipulate the transmitted messages.

Theorem A.7 *The proposed protocol can achieve data integrity property.*

Proof: While the each participant sends the sensitive data to another participant in the instance protocol by the communication channel, the adversary alter or manipulate the data and cheat one of the honest participants by relying on the wrong session keys.

If the adversary wants to alter or manipulate the message $m_1 = \{ID_A, T_a(x), C_1\}$ of **step1** for cheating S_i and S_j, the adversary will be detected in the **step3**. Because the adversary does not have the pw_A and the personal biometric image sample B of Alice, then the adversary cannot compute $C_1 = H(R||T_a(x))$, finally the adversary cannot pass the validation of S_j. Simply speaking, if the adversary wants to alter or manipulate the message m_2 (including m_1), the adversary will be detected in the **step3** by S_j. If the adversary wants to alter or manipulate the message m_3 and m_4, the adversary will be detected in the **step4** and **step5** by S_j and Alice relatively. As for m_5, the adversary cannot alter or manipulate it because m_5 is just a value of a secure hash and S_i can verify m_5 by the local information.

(7) Guessing attacks

Definition A.8 In an off-line guessing attack, an attacker guesses a password or long-term secret key and verifies his/her guess, but he/she does not need to participate in any communication during the guessing phase. In an undetectable on-line guessing attack, an attacker searches to verify a guessed password or long-term secret key in an on-line transaction and a failed guess cannot be detected and logged by the server.

Theorem A.8 *The proposed protocol can resist Guessing attacks.*

Proof: In our proposed scheme of the authenticated key exchange phase, the undetectable on-line guessing attack will fail since after **Step3**, S_j can authenticate Alice. The off-line guessing attack will not work against the proposed scheme since the password pw_A is not transmitted on the public channel at all. Even if the adversary gets the Alices smart card and wants to execute off-line guessing attack, he will fail because the information $Z = R \oplus h\,(pw_A||B)$ is protected by Alices personal biometric image sample B. Therefore, the proposed scheme can resist guessing attacks.

(8) Session key security

Definition A.9 A communication protocol exhibits session key security if the session key cannot be obtained without any long-term secrets.

Theorem A.9 *The proposed protocol can achieve session key security.*

Proof: In the authenticated key agreement phase and password and shared secret key among servers update 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. Whenever the communication ends between S_i and Alice, the key will immediately self-destruct and will not be reused. Therefore, assuming the attacker has obtained a

session key, Alice will be unable to use this session key to decode the information in other communication processes. Because the random point elements a and r_i are all generated randomly and are protected by the CMBDLP, CMBDHP, and the secure symmetric encryption, a known session key cannot be used to calculate the value of the next session key. Additionally, since the values 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.

(9) Key Compromise Impersonation Attacks (KCI attacks)

Definition A.10 An adversary is said to impersonate a party B to another party A if B is honest and the protocol instance at A accepts the session with B as one of the session peers but there exists no such partnered instance at B [47]. In a successful KCI attack, an adversary with the knowledge of the long-term private key of a party A can impersonate B to A. **Theorem A.10** *The proposed protocol can resist KCI attack.*

Proof: We assume that an adversary can know Alices secret password pw_A. Then, an adversary can impersonate S_i or S_j to cheat Alice, and to get the session key. But above-mentioned process will not be achieved and the attack course terminates at the beginning. Because an adversary cannot own the S_js secret key k_j, and then an adversary cannot compute $C_i(2 \leq i \leq 4)$ and finally the adversary cannot pass validation of S_j. So the key compromise impersonation attacks will fail.

(10) Stolen smart card attacks

Definition A.11 Anyone gets the smart card in some way to execute some kinds of attacks.

Theorem A.11 *The proposed scheme can resist stolen smart card attacks.*

Proof: It is very clear that the proposed scheme provides biometrics authentication. Anyone including an adversary cannot pass the biometric verification. Therefore, the proposed scheme can resist stolen smart card attacks.

APPENDIX B. 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 [51]. The basic descriptions are shown in Table 4.

TABLE 4. Descriptions the model of Canetti and Krawczyk

Symbol	Definition
parties $P_1, ...P_n$	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(Pi) : This query allows the adversary to take over the party P_i, including long-lived keys and any session-specific information in P_is 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 correctly guesses b, and we define the advantage of adversary 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 had issued SessionStateReveal(s), SessionKeyReveal(s), Corrupt(Pi) before s would have expired.

Definition B 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,

Algorithm 1 CMBDHP distinguisher

Input: $H, E_K()/D_K(), (x, T_{k_A}(x)), (x, T_{k_B}(x))$

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 $\{i, T_{R_A}(x), ID_A, ID_B, C_1\}$ to a server, and let a server send $\{i, T_{R_B}(x), ID_A, ID_B, C_2\}$ to a user, where i is the session identifier. Both the user and the server can compute the session key $SK = H(T_{R_A}T_{R_B}(x))$ 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** $d \xleftarrow{R} \{0,1\}$.

8:**end if**

Output: d

(3) If two uncorrupted parties have completed matching sessions with predistributed parameter, these sessions produce the same key as output; (4) is negligible. Theorem B.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 [51].

Proof. The proof's process is similar to **Theorem B.1**. The protocol Π_2 is the composable instance of protocol multiple Π_1. Since Theorem B.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 extinguisher 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_2.

Algorithm 2 Proposed protocol simulator

Input: $H, E_K() / D_K(), (x, T_{k_A}(x)), (x, T_{k_B}(x), , (x, T_{k_C}(x))$

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 run the protocol Π_1 with a server S_i, and let a server run the protocol Π_1 with a trust server S_j. After the trust server S_j authenticate the user and the S_i, the trust server runs the protocol Π_1 with Alice and Bob continuously.
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

(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 B.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 [51].*

Proof. The proofs process is similar to **Theorem B.1.** The protocol Π_2 is the composable instance of protocol multiple Π_1. Since **Theorem B.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.