Wireless Pers Commun (2016) 89:569–597
DOI 10.1007/s11277-016-3293-x

A Novel Chaotic Maps-Based User Authentication and Key Agreement Protocol for Multi-server Environments with Provable Security

Xiong Li[1,2] · Jianwei Niu[2] · Saru Kumari[3] · SK Hafizul Islam[4] · Fan Wu[5] · Muhammad Khurram Khan[6] · Ashok Kumar Das[7]

Published online: 16 April 2016

Abstract The widespread popularity of the computer networks has triggered concerns about information security. Password-based user authentication with key agreement protocols have drawn attentions since it provides proper authentication of a user before granting access right to services, and then ensure secure communication over insecure channels. Recently, Lee et al. pointed out different security flaws on Tsaur et al.'s multi-server user authentication protocol, and they further proposed an extended chaotic maps-based user authentication with key agreement protocol for multi-server environments. However, we observed that Lee et al.'s protocol has some functionality and security flaws, i.e., it is inefficient in detection of unauthorized login and it does not support password change mechanism. Besides, their protocol is vulnerable to registration center spoofing attack and server spoofing attack. In order to remedy the aforementioned flaws, we proposed a novel chaotic maps-based user authentication with key agreement protocol for

✉ Jianwei Niu
buaajianweiniu@163.com; niujianwei2008@gmail.com

Xiong Li
lixiongzhq@163.com

[1] School of Computer Science and Engineering, Hunan University of Science and Technology, Xiangtan 411201, China

[2] State Key Laboratory of Software Development Environment, Beihang University, Beijing 100191, China

[3] Department of Mathematics, Ch. Charan Singh University, Meerut, Uttar Pradesh 250004, India

[4] Department of Computer Science and Information Systems, Birla Institute of Technology and Science, Pilani Campus, Pilani, Rajasthan 333031, India

[5] Department of Computer Science and Engineering, Xiamen Institute of Technology, Xiamen 361021, China

[6] Center of Excellence in Information Assurance, King Saud University, Riyadh 11653, Saudi Arabia

[7] Center for Security, Theory and Algorithmic Research, International Institute of Information Technology, Hyderabad, Gachibowli, Hyderabad 500032, India

multi-server environments. The proposed protocol is provably secure in the random oracle model under the chaotic-maps based computational Diffie-Hellman assumption. In addition, we analyzed our protocol using BAN logic model. We also compared our protocol with Lee et al.'s protocol in aspects of computation cost, functionalities and securities.

Keywords User authentication · Chaotic maps · Session key agreement · Smart card · Multi-server environments

1 Introduction

Nowadays, benefit from the rapid development of the computer network and the popularization of internet, various services and remote communications can be accomplished by clicking the keyboard in front of the screen. It not only brings huge opportunities to the market, but also provides people with great convenience. However, the security of online transactions and information transmission over insecure networks have became an important issue. As a kind of basic security protocols, user authentication with key agreement protocol has been widely used in the protection of network and information security, which provides identity authentication of the user before he/she accesses to the remote services. In addition, user authentication scheme with key agreement guarantees confidential communication between user and server by negotiating a shared session key.

1.1 Related Studies

In 1981, the pioneering work of password-based user authentication for secure communication over insecure network was proposed by Lamport [1]. However, a password verifier-table is needed to authenticate the users in their scheme. Besides, the user has to reset the password periodically to ensure the security of the protocol. Since then researchers have done a lot of related works [2, 3]. Due to the silent features of smart card, including storage, computing, encryption and other excellent functions, many user authentication protocols based on password and smart card [4–12] were presented by researchers. In the early stage, most of the user authentication protocols are only applicable to the single server environments. However, the increase in the number of users and the types of services provided by the remote servers brought tremendous challenge to the single server environments. The traditional single server authentication is unable to meet the need of the applications, and then multi-server authentication become an inevitable choice for the users. Compared with the single server authentication, multi-server authentication can offer users with better and more extensive services. However, the more open environments and complex communications cause the users to more vulnerable to attacks. Generally, to get access to services provided by different servers securely, the user has to use different passwords for multiple registration. However, with the increase in the number of the servers, it is impossible for a user to remember so many passwords. Therefore, the protocols for single server environments cannot meet the requirements of multi-server applications. In 2001, Li et al. [13] first explored the multi-server authentication problem, and they designed a multi-server authentication protocol utilizing the neural networks. However, their protocol was found inefficient since huge computation and communication costs are needed to train the neural network [14].

In the year 2003, Lin et al. [14] designed an authentication protocol for multi-server environments based on the ElGamal digital signature and the intractability of the discrete logarithm problem (DLP). However, in their protocol, the user has to store large number of system parameters for each servers. In addition, this protocol does not achieve the mutual authentication. In order to eliminate these issues, Juang [15] presented a low-cost multi-server authentication protocol based on hash function and symmetric cryptographic operations. Chang and Lee [16] further analyzed the problem of Juang's protocol, and they presented an enhanced protocol. However, the protocol [16] was found insecure against insider attack and spoofing attack [17]. Later, Tsaur et al. constructed two user authentication protocols [17, 18] for multi-server environments based on the Lagrange polynomial interpolation and digital signature standard (DSS). In 2008, Tsai [19] proposed a new multi-server authentication protocol based on the hash function, and demonstrated that their protocol can resist various attacks. Contemporaneity, some of the works on multi-server user authentication [20–22] were put forward by researchers. However, these protocols have security defects [23–25].

In many cases, user does not want to leak the identity when he/she accesses services, such as in applications of network game, network television and Internet banking. Therefore, user anonymity becomes an important feature of authentication protocol. However, most of the aforementioned multi-server authentication protocols are relied on the static identity, which can be traced by attacker. Thus, the design of an authentication protocols with user anonymity and other necessary functionalities become a hot topic in information security. In 2009, Liao and Wang [26] first presented a multi-server user authentication protocol with user anonymity based on dynamic identity mechanism. In their protocol, user's identity of one session is different from those of other sessions, which provides user anonymity and keeps user's identity from being tracked. Later, Liao and Wang's protocol [26] was found vulnerable to various attacks [27]. Hsiang and Shih [27] gave an enhanced protocol against the defective protocol presented in [26]. However, the protocol in [27] was still insecure, and both Sood et al. [28] and Lee et al. [29] found some security flaws of the protocol in [27], respectively. Further, both of them independently proposed their enhanced protocols. Unfortunately, both the enhanced protocols in [28, 29] were still found vulnerable to some attacks [30, 31].

1.2 Motivation and Contribution

Due to the chaotic system's excellent properties of diffusion and confusion, chaos theory based cryptography received much attentions in recent years, and many chaotic system based user authentication protocols [33, 34] and key agreement protocols [35–39] have been constructed by researchers. In 2012, Tsaur et al. [32] constructed a user authentication protocol for multi-server environments based on self-verified timestamp. This protocol is efficient since it only uses symmetric encryption and hash function. However, Lee et al. [33] pointed out that Tsaur et al.'s protocol suffers from the insider attack, known-plaintext attack, and it does not guarantee the user anonymity. Later, Lee et al. constructed a new authentication protocol for multi-server environments using extend chaotic maps. However, we have identified some functionality and security drawbacks of the protocol in [33] as follows: (1) it is inefficient in the detection of unauthorized login; (2) it does not support password change; (3) it is vulnerable to registration center spoofing attack; (4) it vulnerable to server spoofing attack. Therefore, we have designed a novel chaotic maps-based user authentication with key agreement protocol for multi-server environments to resolve the drawbacks of the protocol presented in [33]. We demonstrate the provable security analysis

of our protocol in the random oracle model. In addition, the formal security of our protocol is validated using BAN logic model. We also analyze other functional features and security aspects of the presented protocol. Based on the functionality requirements and security aspects, we compare our protocol with the protocol in [33], which proved that our protocol is a more secure user authentication protocol for multi-server environments.

1.3 Outline of the Paper

The remaining part of this paper are arranged as follows. Section 2 described some preliminaries. The review of the protocol in [33] and its cryptanalysis are described in Sects. 3 and 4, respectively. Section 5 described the proposed chaotic maps-based user authentication protocol for multi-server environments. The provable security analysis in the random oracle model, the formal security validation using BAN logic model and informal security analysis of the proposed protocol are presented in Sect. 6. The performance analysis and comparison of our protocol with the protocol presented in [33] are described in Sect. 7. Finally, Sect. 8 summarized the paper.

2 Chebyshev Chaotic Maps

In this section, Chebyshev chaotic maps and two intractable problems are discussed.

Definition 1 Let n is an integer and x is a variable chooses from the interval $[-1, 1]$. The Chebyshev polynomial $T_n(x) : [-1, 1] \rightarrow [-1, 1]$ is defined as $T_n(x) = \cos(n \cdot \arccos(x))$.

From the Definition 1, the following recurrence relation of the Chebyshev polynomial can be derived:

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

where $T_0(x) = 1$ and $T_1(x) = x$.

Definition 2 *(Chaotic property)* When $n > 1$, the Chebyshev polynomial map $T_n(x) : [-1, 1] \rightarrow [-1, 1]$ of degree n is a chaotic map with invariant density $f^*(x) = 1/(\pi\sqrt{1 - x^2})$ for positive Lyapunov exponent $\lambda = \ln n > 0$.

Definition 3 *(Semi-group property)* This property can be described as follows:

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

where r and s are two positive integers and $x \in [-1, 1]$.

In 2008, Zhang [40] further extended the semi-group property of Chebyshev polynomials to the interval $(-\infty, +\infty)$, i.e., for $T_n(x) = (2xT_{n-1}(x) - T_{n-2}(x)) \bmod p$, where $n \geq 2$, $x \in (-\infty, +\infty)$ and p is a large prime. The equation $T_r(T_s(x)) \equiv T_{rs}(x) \equiv T_{sr}(x) \equiv T_s(T_r(x)) \bmod p$ is also hold for $x \in (-\infty, +\infty)$.

2.1 Computational Problem and Assumption

Definition 4 *(Negligible function)* A function $\epsilon(k)$ is said to be negligible if, for every $c > 0$, there exists a k_0 such that $\epsilon(k) \leq \frac{1}{k^c}$ for every $k \geq k_0$.

Definition 5 *(Chaotic maps-based Discrete Logarithm Problem (DLP))* Given $\{x, T_r(x) \bmod p\}$, it is hard for a polynomial time bounded algorithm $\mathcal{A}$ to compute r. The probability that $\mathcal{A}$ can find the solution of the DLP problem is defined as $Adv^{DLP}(\mathcal{A}) = Pr[\mathcal{A}(x, T_r(x) \bmod p) = r : r \in Z_p^*]$.

Definition 6 *(Chaotic maps-based Diffie-Hellman (CDH) problem)* Given a random tuple $\{x, T_r(x) \bmod p, T_s(x) \bmod p\}$, it is hard for a polynomial time bounded algorithm $\mathcal{A}$ to compute $T_{rs}(x) \bmod p$. The probability that $\mathcal{A}$ can find the solution of the CDH problem is defined as $Adv^{CDH}(\mathcal{A}) = Pr[\mathcal{A}(x, T_r(x) \bmod p, T_s(x) \bmod p) = T_{rs}(x) \bmod p : r, s \in Z_p^*]$.

Definition 7 *(DLP assumption)* The probability $Adv^{DLP}(\mathcal{A})$ is negligible for any $\mathcal{A}$, i.e, $Adv^{DLP}(\mathcal{A}) \leq \epsilon$, for some negligible function ϵ.

Definition 8 *(CDH assumption)* The probability $Adv^{CDH}(\mathcal{A})$ is negligible for any $\mathcal{A}$, i.e, $Adv^{CDH}(\mathcal{A} \leq \epsilon$, for some negligible function ϵ.

Definition 9 According to [41], the period of Chebyshev polynomial $T_n(x)$ is the divisor of $p^2 - 1$. In other way, we can say that $p - 1$ or $p + 1$ is the period of $T_n(x)$. Here we use $p + 1$.

Table 1 The notations used in this paper

Notation	Description
U_i	ith user
S_j	jth server
RC	Registration center
ID_i	Identity of U_i
PW_i	Password of U_i
SID_j	Identity of S_j
w	Master secret key of RC
P_i	Service period of S_j for U_i
p	Large prime number
X	Random number produced by RC
R	Public key of RC
r_i	Random number selected by U_i
r_j	Random number selected by S_j
r_k	Random number selected by RC
SK	Agreed session key between U_i and S_j
$h(\cdot)$	Secure one-way hash function with l bits
$\oplus$	XOR operator
$\|$	Message concatenation operator
$\rightrightarrows$	Secure channel
$\rightarrow$	Public (Insecure) channel

3 Review of the Protocol in [33]

In this part, we reviewed Lee et al.'s protocol [33]. In [33], there are three parties i.e., the registration center RC, user U_i and server S_j. RC first selects a random number X, two random integers (r, s), and then RC computes $w = h(r||s)$ and $R \equiv T_w(X) \bmod p$. At last, RC keeps the master keys (r, s) secret, and shares w with S_j secretly via a secure channel. The protocol given in [33] includes two phases, i.e. registration phase and login and session key agreement phase. Table 1 lists the symbols used throughout this paper. The overview of protocol in [33] are as follows:

3.1 Registration

1. $U_i \rightrightarrows RC$: $\{ID_i, h(PW_i) \oplus N\}$ U_i selects ID_i, PW_i and a nonce N. U_i computes $h(PW_i) \oplus N\}$ and then sends $\{ID_i, h(PW_i) \oplus N\}$ to RC through a secure channel.
2. $RC \rightrightarrows U_i$: Smart card RC computes $v_i = h(ID_i \| P_i \| w)$ and $\mu_i = v_i \oplus h(PW_i) \oplus N$, and injects ID_i, $\{\mu_i, P_i, R, X, h(\cdot), p\}$ into a new smart card. Then RC issues the smart card to U_i secretly.
3. U_i computes $\mu_i' = \mu_i \oplus N$ and replaces μ_i with μ_i' in the smart card.

3.2 Login and Session Key Agreement

In this phase, the smart card and S_j will perform the followings:

1. $U_i \rightarrow S_j$: $\{M_{ij}, UID_i, C_1, P_i\}$ U_i inserts the smart card and keys ID_i and PW_i. Then the smart card computes $v_i = \mu_i' \oplus h(PW_i)$ and picks a random integer r_i. Then it also calculates

$$\begin{aligned} C_1 &\equiv T_{r_i}(X) \bmod p \\ C_2 &\equiv T_{r_i}(R) \bmod p \\ UID_i &= ID_i \oplus h(C_1 \| C_2) \\ M_{ij} &= h(ID_i \| UID_i \| P_i \| v_i \| C_1 \| C_2) \end{aligned}$$

   and submits $\{M_{ij}, UID_i, C_1, P_i\}$ as the login request to S_j over a public channel.
2. $S_j \rightarrow U_i$: $\{M_{ji}, C_3, V_i\}$ Upon receiving the login request from U_i, S_j calculates $C_2' \equiv T_w(C_1) \bmod p$, $ID_i' = UID_i \oplus h(C_1 \| C_2')$, $v_i' = h(ID_i' \| P_i \| w)$, and checks $h(ID_i' \| UID_i \| P_i \| v_i' \| C_1 \| C_2') \stackrel{?}{=} M_{ij}$. If not, S_j rejects the request. Otherwise, S_j further checks whether the service period P_i is expired. If P_i is expired, S_j terminates the session. On the contrary, S_j updates P_i with $P_i^{new} = P_i - 1$. Then S_j picks a random integer r_j and calculates $v_i^{new} = h(ID_i' \| P_i^{new} \| w)$, $V_i = v_i' \oplus v_i^{new}$, $C_3 \equiv T_{r_j}(X) \bmod p$, $SK \equiv T_{r_j}(C_1) \equiv T_{r_j r_i}(X) \bmod p$, and $M_{ji} = h(ID_i' \| v_i' \| v_i^{new} \| P_i^{new} \| C_2' \| C_3 \| SK)$. At last, S_j sends $\{M_{ji}, C_3, V_i\}$ to U_i over a public channel.
3. $U_i \rightarrow S_j$: $\{M_{sk}\}$ The smart card calculates $v_i^{new\prime} = V_i \oplus v_i$, $P_i^{new} = P_i - 1$, $SK' \equiv T_{r_i}(C_3) \equiv T_{r_i r_j}(X) \bmod p$, and checks $h(ID_i \| v_i \| v_i^{new\prime} \| P_i^{new} \| C_2 \| C_3 \| SK') \stackrel{?}{=} M_{ji}$. If it is incorrect, the session will be stopped. Otherwise, the validity of S_j is authenticated by U_i. Then, the smart card calculates $\mu_i^{new} = v_i^{new\prime} \oplus h(PW_i)$ and updates $\{\mu_i', P_i\}$ with $\{\mu_i^{new}, P_i^{new}\}$. At last, the smart card calculates $M_{sk} = h(C_2 \| SK')$ and sends it to S_j over a public channel.

4. Upon receiving $\{M_{sk}\}$ from U_i, S_j checks $h(C_2'\|SK)\stackrel{?}{=}M_{sk}$. If they are not equal, the session will be terminated. Otherwise, U_i is authenticated by S_j, and U_i and S_j agree on a session key SK.

4 Cryptanalysis of the Protocol in [33]

In this part, we point out the drawbacks of protocol in [33] from the aspects of functionality and security.

4.1 No Single Registration

One of the important features of multi-server user authentication protocol is that the user can access services provided by different servers with a single registration. The single registration helps the user to access different servers with a single password and identity. However, to access multiple servers in the protocol given in [33], U_i has to repeatedly register to RC for each server S_j to get v_i and μ_i. Therefore, the protocol given in [33] is not user friendly for multi-server environments.

4.2 Inefficient Detection of Unauthorized Login

In real applications, it is suggested that the user needs to set different passwords for diverse applications to guarantee the adequate security of network-based services. Then, the user may be confused by the different passwords for diverse services, and would be likely to input wrong password in the login stage. Therefore, it is plausible and an ideal feature that any unauthorized login launched by entering a wrong password must be detected by the smart card [42]. Otherwise, the protocol may suffer from DoS (Denial of Service) attack due to the server has to spend a lot of resources to check unauthorized login messages launched by the attacker.

Due to the absence of wrong password detection mechanism of the protocol given in [33], even if U_i inputs an incorrect password by mistake in login stage, the processes of the protocol will still be performed until some wrong information is detected by S_j. The detailed descriptions of this situation are given below:

Assume that a wrong password $PW_i^*(\neq PW_i)$ is entered by U_i when he/she initiates a session, then the following procedures would be executed by smart card of U_i and S_j:

1. The smart card calculates $v_i^* = \mu_i' \oplus h(PW_i^*)(= v_i \oplus h(PW_i) \oplus h(PW_i^*) \neq v_i = h(ID_i\|P_i\|w))$ and generates a random integer r_i. Then it calculates $C_1 \equiv T_{r_i}(X) \bmod p$, $C_2 \equiv T_{r_i}(R) \bmod p$, $UID_i = ID_i \oplus h(C_1\|C_2)$, $M_{ij}^* = h(ID_i\|UID_i\|P_i\|v_i^*\|C_1\|C_2)$. At last, the message $\{M_{ij}^*, UID_i, C_1, P_i\}$ is submitted as the login request to S_j.
2. After receiving $\{M_{ij}^*, UID_i, C_1, P_i\}$ from U_i, S_j first calculates $C_2' \equiv T_w(C_1) \bmod p$, $ID_i' = UID_i \oplus h(C_1\|C_2')$, $v_i' = h(ID_i'\|P_i\|w)$, and it can be seen that $h(ID_i'\|UID_i\|P_i\|v_i'\|C_1\|C_2') \neq h(ID_i\|UID_i\|P_i\|v_i^*\|C_1\|C_2) = M_{ij}^*$ since $v_i^* \neq v_i'$. Until now, the unauthorized login caused by wrong password was rejected by S_j.

Therefore, the protocol in [33] is inefficient to detect the unauthorized login, which was generated due to wrong password. It is illogical and will increase the communication and

computational overheads on the full system. Furthermore, it may be utilized by an attacker to launch a DoS (Denial of Service) attack on the system.

4.3 No Support for Password Change

Periodically change of the password is not only convenient for the users, but also an important strategy to guarantee the security of the authentication protocol. Therefore, the password change is one of the most basic function and this facility must be incorporated in the password-based authentication protocol to provide more robustness. However, we found that the protocol in [33] does not support the update of the password. It is really not easy thing to add the password change function in [33] due to the absence of wrong password detection mechanism on smart card side. Therefore, the user has to adopt the face to face approach with the registration center to update the password, and it is undoubtedly a functional defect of the protocol in [33].

4.4 Registration Center Spoofing Attack

In [33], we clearly observed that RC shared the same secret key $w = h(r\|s)$ with all the servers, and it would introduce some security defects to the protocol. First, w becomes the actual master secret key of the system, and the securities of the system are relying on w. In order to reduce the risk of leakage, the master secret key should be held by RC only. However, in [33], w is known to all the servers except RC and the security of the system will collapse if any of the server is compromised. Therefore, sharing the secret key w with all servers is one of the most serious security flaws in the design of the protocol in [33].

Furthermore, in the registration phase of the protocol in [33], U_i's secret information v_i and μ_i are calculated by RC based on w only. Therefore, with the public information R, X, $h(\cdot)$ and the shared secret key w, any server $S_j \in \{S_1, S_2, \ldots, S_r\}$ can impersonate RC without knowing the secret information (r, s), therefore, the protocol in [33] is vulnerable to the registration center spoofing attack.

4.5 Server Spoofing Attack

Last but not least, since all the servers share the same secret key w and all the messages generated in login and session key agreement phase rely on w, any malicious server of the system can camouflage as any other servers. When U_i transmits the login message $\{M_{ij}, UID_i, C_1, P_i\}$ to S_j, it can be intercepted by a malicious server, say S_n. Then, S_n calculates C_2', ID_i', v_i' using w, and gets $h(ID_i'\|UID_i\|P_i\|v_i'\|C_1\|C_2') = M_{ij}$. Then, S_n updates the service period P_i with $P_i^{new} = P_i - 1$ and computes $v_i^{new} = h(ID_i'\|P_i^{new}\|w)$, $V_i = v_i' \oplus v_i^{new}$. Then S_n chooses a random integer r_n and computes $C_3 \equiv T_{r_n}(X) \bmod p$, $SK \equiv T_{r_n}(C_1) \equiv T_{r_n r_i}(X) \bmod p$. Finally, S_n computes $M_{ni} = h(ID_i'\|v_i'\|v_i^{new}\|P_i^{new}\|C_2'\|C_3\|SK)$ and transmits the message $\{M_{ni}, C_3, V_i\}$ to U_i. U_i's smart card then computes $v_i^{new\prime} = V_i \oplus v_i$, $P_i^{new} = P_i - 1$, $SK' \equiv T_{r_i}(C_3) \equiv T_{r_i r_n}(X) \bmod p$ and gets $h(ID_i\|v_i\|v_i^{new\prime}\|P_i^{new}\|C_2\|C_3\|SK') = M_{ni}$. Then, the smart card calculates $\mu_i^{new} = v_i^{new\prime} \oplus h(PW_i)$ and updates $\{\mu_i', P_i\}$ with $\{\mu_i^{new}, P_i^{new}\}$. Finally, the smart card calculates $M_{sk} = h(C_2\|SK')$ and transmits it to S_n. Upon receiving the message $\{M_{sk}\}$ from U_i, S_n calculates $h(C_2'\|SK)$ and gets $h(C_2'\|SK) = M_{sk}$. At last, a session key SK is shared between U_i and S_n. Therefore, any malicious server of the system can camouflage as other servers to cheat any registered user and accordingly, the protocol in [33] is suffered from the server spoofing attack.

5 The Proposed Protocol

In this part, we design a new multi-server authentication protocol utilizing the extended chaotic maps. There are also three parties in the designed protocol, i.e. registration center *RC*, server S_j and user U_i. Here, we considered *RC* is a trusted entity and in charge of the selection of the system parameters, the registration of the users and the servers, and the authentication of the users. To initialize the system, *RC* first selects a random number X and a system master key w, and computes the public key $R \equiv T_w(X) \bmod p$. In order to register to the system, S_j transmits the unique identity SID_j to *RC* in a secret manner. When receiving the registration request, *RC* computes $K_j = h(SID_j \| w)$ and transmits $\{K_j, X\}$ to S_j in a secret manner. In the proposed protocol, there are three phases, namely registration, login and session key agreement, and password change. These three phases is described below:

5.1 Registration

If U_i wants to enjoy the services provided by servers $\{S_1, S_2, \ldots, S_r\}$, he/she needs to register to *RC* in advance, and the following procedures should be executed.

1. $U_i \rightrightarrows RC$: $\{ID_i, e_i\}$ U_i first selects the identity ID_i and password PW_i. Then U_i picks a random number N_i and calculates $e_i = h(PW_i \| N_i)\}$, and transmits the registration request message $\{ID_i, e_i = h(PW_i \| N_i)\}$ to *RC* through a secure channel.
2. $RC \rightrightarrows U_i$: Smart card Upon receiving the registration request from U_i, *RC* calculates $f_i = h(ID_i \| w)$, $A_i = h(ID_i \| e_i)$, $B_i = e_i \oplus f_i$ and stores $\{A_i, B_i, X, R, h(\cdot), p\}$ into the memory of a smart card. Then the smart card is issued by *RC* to U_i through a reliable channel.
3. U_i injects the random number N_i into the smart card, and the random number N_i is no longer need to be remembered by U_i. Finally, the smart card contains the information $\{A_i, B_i, N_i, X, R, h(\cdot), p\}$.

We also illustrate the registration phase of our protocol in Fig. 1.

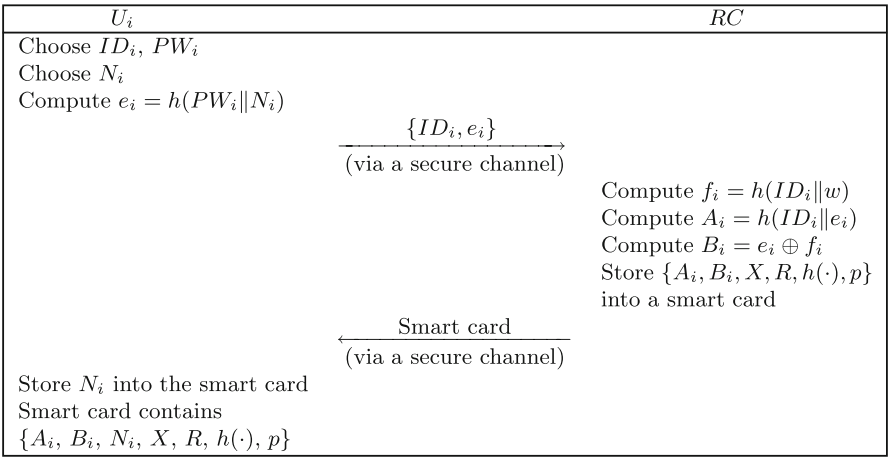

Fig. 1 The registration phase of the new protocol

5.2 Login and Session Key Agreement

In this phase, whenever U_i wants to login to S_j, the following procedures should be executed among U_i, S_j and RC.

1. $U_i \rightarrow S_j$: $\{UID_i, C_1, M_{ik}\}$ U_i first chooses the identity SID_j of S_j on which he/she wants to access. U_i inserts the smart card into a card reader and then keys ID_i and PW_i. Then, the smart card calculates $e_i = h(PW_i \| N_i)$, $A_i' = h(ID_i \| e_i)$ and checks $A_i' \stackrel{?}{=} A_i$. If the equation does not hold, it means the inputted identity or password is invalid, and thus, the connection is stopped by the smart card. Otherwise, the smart card picks a random integer r_i and calculates the followings:

$$\begin{aligned} C_1 &\equiv T_{r_i}(X) \bmod p, \\ C_2 &\equiv T_{r_i}(R) \bmod p, \\ UID_i &= ID_i \oplus h(C_1 \| C_2), \\ f_i &= B_i \oplus e_i, \\ M_{ik} &= h(ID_i \| SID_j \| f_i \| C_1 \| C_2), \end{aligned}$$

   the login message $\{UID_i, C_1, M_{ik}\}$ is submitted to S_j over an unreliable channel.
2. $S_j \rightarrow RC$: $\{UID_i, C_1, M_{ik}, SID_j, C_3, M_{jk}\}$ After receiving $\{UID_i, C_1, M_{ik}\}$, S_j generates an random number r_j and computes

$$\begin{aligned} C_3 &\equiv T_{r_j}(X) \bmod p, \\ M_{jk} &= h(K_j \| C_3), \end{aligned}$$

   S_j forwards $\{UID_i, C_1, M_{ik}, SID_j, C_3, M_{jk}\}$ to RC over an unreliable channel.
3. $RC \rightarrow S_j$: $\{D_i, M_{kj}, F_i, M_{ki}\}$ Upon receiving $\{UID_i, C_1, M_{ik}, SID_j, C_3, M_{jk}\}$, RC calculates $K_j' = h(SID_j \| w)$, $M_{jk}' = h(K_j' \| C_3)$ and checks whether M_{jk}' equals to M_{jk}. If they are not equal, the session is rejected by RC. Otherwise, S_j is authenticated by RC. Then, RC computes $C_2' \equiv T_w(C_1) \bmod p$, $ID_i' = UID_i \oplus h(C_1 \| C_2')$, $f_i' = h(ID_i' \| w)$, $M_{ik}' = h(ID_i' \| SID_j \| f_i' \| C_1 \| C_2')$ and checks $M_{ik}' \stackrel{?}{=} M_{ik}$. If they are not equal, the session is rejected by RC. Otherwise, the validity of U_i is confirmed by RC and then RC accepts the login request. For mutual authentication, RC picks a random number r_k and calculates $D_i = K_j' \oplus r_k$, $M_{kj} = h(K_j' \| C_1 \| C_3 \| r_k)$, $E_i = h(K_j' \| r_k)$, $F_i = E_i \oplus f_i' \oplus C_2'$, $M_{ki} = h(f_i' \| F_i \| SID_j \| C_2' \| C_3)$. Then, RC forwards the response message $\{D_i, M_{kj}, F_i, M_{ki}\}$ to server S_j over an unreliable channel.
4. $S_j \rightarrow U_i$: $\{F_i, M_{ki}, C_3, M_{ji}\}$ Upon receiving the response message $\{D_i, M_{kj}, F_i, M_{ki}\}$ from RC, S_j calculates $r_k' = K_j \oplus D_i$, $M_{kj}' = h(K_j \| C_1 \| C_3 \| r_k')$ and checks $M_{kj}' \stackrel{?}{=} M_{kj}$. If it does not hold, the session is rejected by S_j. Otherwise, RC is authenticated by S_j. Then, S_j computes $E_i' = h(K_j \| r_k')$, $SK \equiv T_{r_j}(C_1) \bmod p$, $M_{ji} = h(SID_j \| C_1 \| C_3 \| E_i' \| SK)$ and forwards the message $\{F_i, M_{ki}, C_3, M_{ji}\}$ to U_i over an unreliable channel.
5. $U_i \rightarrow S_j$: $\{M_{ij}\}$ Upon receiving the message $\{F_i, M_{ki}, C_3, M_{ji}\}$ from S_j, the smart card calculates $M_{ki}' = h(f_i \| F_i \| SID_j \| C_2 \| C_3)$ and checks $M_{ki}' \stackrel{?}{=} M_{ki}$. If $M_{ki}' \neq M_{ki}$, the session is rejected by U_i. Otherwise, the validity of RC is confirmed by U_i. Then, U_i computes $E_i'' = F_i \oplus f_i \oplus C_2$, $SK' \equiv T_{r_i}(C_3) \bmod p$, $M_{ji}' = h(SID_j \| C_1 \| C_3 \| E_i'' \| SK')$ and checks $M_{ji}' \stackrel{?}{=} M_{ji}$. If $M_{ji}' \neq M_{ji}$, the session is also rejected by U_i. Otherwise, the validity of S_j

is confirmed by U_i. At last, U_i calculates $M_{ij} = h(C_3 \| E_i'' \| SK')$ and forwards it to S_j over an unreliable channel.

6. On receiving the message $\{M_{ij}\}$, S_j calculates $M'_{ij} = h(C_3 \| E_i' \| SK)$ and checks $M'_{ij} \stackrel{?}{=} M_{ij}$. If $M'_{ij} \neq M_{ij}$, the connection is stopped by S_j. Otherwise, the validity of U_i is confirmed by S_j.

At last, a session key $SK \equiv T_{r_j}(C_1) \bmod p \equiv T_{r_i r_j}(X) \bmod p \equiv T_{r_i}(C_3) \bmod p = SK'$ is agreed between U_i and S_j, which can be used to securing the successive communication between U_i and S_j.

We also illustrate the login and key agreement phase of our protocol in Fig. 2.

U_i	S_j	RC
Input ID_i, PW_i and SID_j		
Compute $e_i = h(PW_i \| \| N_i)$		
Compute $A'_i = h(ID_i \| \| e_i)$		
Verify identity and password		
Verify $A'_i \stackrel{?}{=} A_i$		
Generate r_i		
Compute $C_1 \equiv T_{r_i}(X) \bmod p$		
Compute $C_2 \equiv T_{r_i}(R) \bmod p$		
Compute $UID_i = ID_i \oplus h(C_1 \| \| C_2)$		
Compute $f_i = B_i \oplus e_i$		
Compute $M_{ik} = h(ID_i \| \| SID_j \| \| f_i \| \| C_1 \| \| C_2)$		
$\{UID_i, C_1, M_{ik}\}$ $\longrightarrow$ (via an insecure channel)		
	Generate r_j	
	Compute $C_3 \equiv T_{r_j}(X) \bmod p$	
	Compute $M_{jk} = h(K_j \| \| C_3)$	
	$\{UID_i, C_1, M_{ik}, SID_j, C_3, M_{jk}\}$ $\longrightarrow$ (via an insecure channel)	
		Compute $K'_j = h(SID_j \| \| w)$
		Compute $M'_{jk} = h(K'_j \| \| C_3)$
		Compute $M'_{jk} ? = M_{jk}$ (RC authenticates S_j)
		Compute $C'_2 \equiv T_w(C_1) \bmod p$
		Compute $ID'_i = UID_i \oplus h(C_1 \| \| C'_2)$
		Compute $f'_i = h(ID'_i \| \| w)$
		Compute $M'_{ik} = h(ID'_i \| \| SID_j \| \| f'_i \| \| C_1 \| \| C'_2)$
		Verify $M'_{ik} \stackrel{?}{=} M_{ik}$ (RC authenticates U_i)
		Generate r_k
		Compute $D_i = K'_j \oplus r_k$
		Compute $M_{kj} = h(K'_j \| \| C_1 \| \| C_3 \| \| r_k)$
		Compute $E_i = h(K'_j \| \| r_k)$
		Compute $F_i = E_i \oplus f'_i \oplus C'_2$
		Compute $M_{ki} = h(f'_i \| \| F_i \| \| SID_j \| \| C'_2 \| \| C_3)$
		$\{D_i, M_{kj}, F_i, M_{kj}\}$ $\longleftarrow$ (via an insecure channel)
	Compute $r'_k = K_j \oplus D_i$	
	Compute $M'_{kj} = h(K_j \| \| C_1 \| \| C_3 \| \| r'_k)$	
	Verify $M'_{kj} \stackrel{?}{=} M_{kj}$ (S_j authenticates RC)	
	Compute $E'_i = h(K_j \| \| r'_k)$	
	Compute $SK \equiv T_{r_j}(C_1) \bmod p$	
	Compute $M_{ji} = h(SID_j \| \| C_1 \| \| C_3 \| \| E'_i \| \| SK)$	
	$\{F_i, M_{ki}, C_3, M_{ji}\}$ $\longleftarrow$ (via an insecure channel)	
Compute $M'_{ki} = h(f_i \| \| F_i \| \| SID_j \| \| C_2 \| \| C_3)$		
Verify $M'_{ki} \stackrel{?}{=} M_{ki}$ (U_i authenticates RC)		
Compute $E''_i = F_i \oplus f_i \oplus C_2$		
Compute $SK' \equiv T_{r_i}(C_3) \bmod p$		
Compute $M'_{ji} = h(SID_j \| \| C_1 \| \| C_3 \| \| E''_i \| \| SK')$		
Verify $M'_{ji} \stackrel{?}{=} M_{ji}$ (U_i authenticates S_j)		
Compute $M_{ij} = h(C_3 \| \| E''_i \| \| SK')$		
$\{M_{ij}\}$ $\longrightarrow$ (via an insecure channel)		
	Compute $M'_{ij} = h(C_3 \| \| E'_i \| \| SK)$	
	Verify $M'_{ij} \stackrel{?}{=} M_{ij}$ (S_j authenticates U_i)	

Fig. 2 The login and key agreement stage of the new protocol

5.3 Password Change

In our protocol, the password can be updated freely without any support from *RC*. In order to do it, U_i first inserts the smart card into a card reader and then keys ID_i, PW_i, and requests to renew the password. Then, the smart card calculates $e_i = h(PW_i \| N_i)$, $A_i' = h(ID_i \| e_i)$ and checks $A_i' \stackrel{?}{=} A_i$. If $A_i' \neq A_i$, it means one of the inputted user's identity or password is invalid, and the password change phase is disconnected by the smart card. Otherwise, U_i is permitted to input a new password PW_i^{new}. Then, the smart card calculates $e_i^{new} = h(PW_i^{new} \| N_i)$, $A_i^{new} = h(ID_i \| e_i^{new})$, $B_i^{new} = B_i \oplus e_i \oplus e_i^{new}$. At last, the smart card replaces A_i, B_i with A_i^{new}, B_i^{new}, respectively.

6 Security Analysis

From the description of the presented protocol, the user can acquire different servers with single registration to *RC*. In this section, the security analysis of the our protocol is given. We show that our protocol is provably secure in the random oracle model against the hardness assumption of the Chaotic maps-based Diffie-Hellman (CDH) problem. Then, BAN logic model [43] is employed for formal security validation of our protocol. This analysis illustrates that the mutual authentication and session key established between U_i and S_j are correct and secure. In addition, we show the proposed chaotic maps-based multi-server authentication with key agreement protocol can resist many known attacks and can achieve many ideal functionalities.

6.1 Provable Security Analysis

6.1.1 Adversarial Model

In this section, we proposed a formal attack model of a password-based user authentication protocol [44–46] based on the following assumptions:

- *RC* and S_j hold a common secret key K_j, *RC* holds a privatekey/public key pair $\{w, T_w(X) \bmod p\}$ and U_i holds a identity-password pair $\{ID_i, PW_i\}$, where the low-entropy password PW_i is chosen from an uniformly distributed small dictionary $\mathcal{D}$. In the registration phase, an injective transformation of $\{ID_i, w, PW_i, X, N_i\}$ is stored into the smart card of U_i.
- We assume that $\mathcal{A}$ is a probabilistic polynomial time adversary. $\mathcal{A}$ and any protocol participant P (either U_i or S_j or *RC*) interacts by executing oracle queries, which give the capability of $\mathcal{A}$ to attack the authentication protocol. We denote tth instance of a participant P by P^t.
- $\mathcal{A}$ controls the communication channel. It implies that $\mathcal{A}$ can block, intercept, remove, inject, or modify, any messages passing through public channels, i.e., all the messages between U_i^l, S_j^m and RC^n are transmitted via $\mathcal{A}$ [47].
- $\mathcal{A}$ may either (a) steal U_i's smart card and find the secret data from it by monitoring the power consumption, timing information and reverse engineering techniques as explained in [48–50] and then may apply some off-line procedure on the extracted data to successfully find the correct password PW_i of U_i; or (b) directly obtain the

password PW_i of U_i. However, $\mathcal{A}$ cannot perform both (a) and (b) at the same time [46].

$\mathcal{A}$ interacts with the protocol participants P via the following oracle queries that give the capabilities to $\mathcal{A}$ to model the real attack. To breach the security of the authentication protocol, $\mathcal{A}$ simulates the following queries:

- **Execute**(U_i^l, S_j^m, RC^n): The *Execute* query helps $\mathcal{A}$ to launch passive attacks against the authentication protocol. This query outputs the messages that are exchanged during the actual execution of the protocol by the participants.
- **Send**(P^t, x): The *Send* query gives the forgery capabilities to $\mathcal{A}$ for active attacks. $\mathcal{A}$ can send a message x through this oracle to P^t and then P^t returns some messages, that are computed by P^t based on the protocol description, to $\mathcal{A}$.
- **Reveal**(P^t): The *Reveal* query helps $\mathcal{A}$ to misuse of session keys. $\mathcal{A}$ can send a *Reveal* query to P^t and then P^t outputs the session key SK that is computed between P^t and its partner provided that the session has *accepted*, otherwise returns a *null* value.
- **Corrupt**(P^t, a): The *Corrupt* query models $\mathcal{A}$ to corrupt a protocol participant P^t, i.e. through this query $\mathcal{A}$ can get secret information about P^t. This query outputs in the following ways:
  - If $a = 1$, it returns U_i's password PW_i to $\mathcal{A}$.
  - If $a = 2$, it returns the information from U_i's smart card.
- **Test**(P^t): The semantic security of the session key SK is measured by this query. This oracle returns the session key if the session key is computed between P^t and its partner, otherwise, returns a *null value*. $\mathcal{A}$ can send only a single *Test* query to P^t. On receiving a *Test*(P^t) query, P^t flips an unbiased coin b and outputs the actual session key SK computed between the protocol participants if $b = 1$. Otherwise, a *random value* is chosen from $\{0, 1\}^*$ and returns it as answer.

Definition 10 An instance P^t is said to be *accepted* if it goes into an *accept state* after receiving the last expected protocol message.

Definition 11 The instances P^t and P^u are said to be *partnered* if (1) Both P^t and P^u is in the state *accepted*, i.e., they mutually authenticate each other and they have established the common session key; (2) Both P^t and P^u share the common *sid*; (3) P^t is P^u's partner and vice-versa. We denote the session identification (*sid*) of instance P^t as the the concatenation of all messages sent and received by P^t.

Definition 12 An instance P^t is called *fresh* if (1) P^t is in the state *accepted*, i.e., P^t and its partner mutually authenticate each other and generate the common session key (2) The *Reveal* queries are never submitted to P^t or its partner; (3) Strictly less than two *Corrupt*-queries have been submitted to P^t, if P is a mobile user instance. Otherwise, strictly less than two *Corrupt*-queries have been submitted to P^t's partner, if P is a foreign agent instance.

Definition 13 Let $Succ_{\mathcal{A}}$ denotes the event that $\mathcal{A}$ executes a single *Test* query to some fresh and terminated instance P^t, and eventually returns a guess bit b'. The adversary breaks the semantics security of the user authentication protocol if $b' = b$ holds and the advantage of $\mathcal{A}$ in violating the semantic security of authentication protocol is described as $Adv_{\mathcal{A}}(p) = 2Pr[Succ_{\mathcal{A}}] - 1$, where the bit b was chosen during the execution of the *Test* query.

Definition 14 A password-based authentication protocol is said to be semantically secured if (1) in the presence of $\mathcal{A}$, P^t and its partner are in *accepted* state and compute the common session key; (2) $Adv_{\mathcal{A}}(p) \leq \epsilon$ for some negligible function ϵ.

6.1.2 Provable Security Analysis

Theorem 1 Let $Adv_{C,\mathcal{A}}(t)$ denotes probability of success of the probabilistic polynomial time t adversary $\mathcal{A}$ in breaching the CDH problem, $\mathcal{D}$ be the small password dictionary, l denotes the security length, which is for hash values and $\mathbb{P}$ is our protocol. Assume that $\mathcal{A}$ can ask at most q_s times *Send* queries, q_e times *Execute* queries and q_h times *Hash* queries to break the sematic security of the designed user authentication protocol and $Adv_{\mathbb{P}}(\mathcal{A})$ is success probability of $\mathcal{A}$ in breaking the semantic security of the proposed protocol, then,

$$Adv_{\mathbb{P}}(\mathcal{A}) \leq \frac{q_h^2}{2^{l-1}} + \frac{(q_s + q_e)^2}{p} + \frac{q_s}{2^{l-1}} + 2 \cdot \left(q_h \cdot Adv_{C,\mathcal{A}}(t) + \frac{q_s}{|\mathcal{D}|} \right)$$

Proof Assume that $\mathcal{A}$ tries to breach the proposed protocol, then we can build an algorithm $\mathcal{C}$ to break the security of the CDH problem, i.e. $\mathcal{C}$ outputs $T_{ab}(X)$ mod p from a given random tuple $\{X, T_a(X) \bmod p, T_b(X) \bmod p\}$, where $a, b \in Z_p^*$. In this proof, we now define sequence of games G_i $(0 \leq i \leq 5)$ and for each game G_i, the event E_i $(0 \leq i \leq 5)$ is also defined that $\mathcal{A}$ successfully guesses the bit b involved in *Test* query. We also define another event F (independent of the game G_i) that may occur during $\mathcal{A}$'s execution and F is detectable by $\mathcal{C}$. It can be noted that the game G_i and G_{i+1} are identical unless F occurs. Thus, we can write

$$|Pr[E_{i+1}] - Pr[E_i]| \leq Pr[F] \tag{1}$$

Game G_0: The real attack in the random oracle model and the simulation of this game are same and thus, we can write

$$Adv_{\mathbb{P}}(\mathcal{A}) = |2Pr[E_0] - 1| \tag{2}$$

Game G_1: This game simulates the has function $h(\cdot)$ by maintaining the hash list L_h. Thus, the real execution of the protocol and the simulation of this game are perfectly indistinguishable since the oracles including *Hash*, *Execute*, *Reveal*, *Send*, *Corrupt* and *Test* are also simulated as done in the real attack. The simulation of the different polynomial number of queries asked by $\mathcal{A}$ are given in Figs. 3, 4, 5, 6, 7 and 8. Thus, we have

$$Pr[E_1] = Pr[E_0] \tag{3}$$

- $\mathcal{C}$ maintains an initial-empty hash list L_h, which maintains the tuples of the form $\{x, y\}$, where $y = h(x)$ is output for the input x. $\mathcal{C}$ returns the answer to $\mathcal{A}$ for each $h(\cdot)$ query as follows:
  - $\mathcal{C}$ searches the list L_h for a tuple $\{x, y\}$ and returns y if such a tuple is found in L_h.
  - Else, $\mathcal{C}$ selects a number $y \in_R Z_p^*$ such that L_h has no tuple of the form $\{\cdot, y\}$ and returns y to $\mathcal{A}$. Finally, $\mathcal{C}$ inserts the new tuple $\{x, y\}$ into the list L_h.

Fig. 3 Simulation of hash oracle queries

Game G_2: The previous game and the simulation of this game are identical except that this game will be halted if a collision occurs during the the simulation of $\{UID_i, C_1, M_{ik}\}$, $\{UID_i, C_1, M_{ik}, SID_j, C_3, M_{jk}\}$, $\{D_i, M_{kj}, F_i, M_{ki}\}$, $\{F_i, M_{ki}, C_3, M_{ji}\}$ and $\{M_{ij}\}$. Based on the birthday attack, probability of collisions of the simulation of $h(\cdot)$ oracle is at most $\frac{q_h^2}{2^{l+1}}$ [46, 51]. It can be noted that $r_i, r_j \in [1, p+1]$, $C_1 \equiv T_{r_i}(X) \mod p$ and $C_3 \equiv T_{r_j}(X) \mod p$ are chosen randomly from Z_p^* and thus the probability of collisions in the transcripts simulation is at most $\frac{(q_s+q_e)^2}{2p}$. Therefore, we have

$$|Pr[E_2] - Pr[E_1]| \leq \frac{q_h^2}{2^{l+1}} + \frac{(q_s+q_e)^2}{2p} \tag{4}$$

Game G_3: The simulations of this game and previous game are identical. The only exception is that this game will be aborted if $\mathcal{A}$ correctly guess the authentication values M_{ik}, M_{jk}, M_{kj}, M_{ki}, M_{ji} and M_{ij} without asking the oracle $h(\cdot)$. Thus, we can say that the simulations of this game and the previous game are identical unless any protocol instance rejects a valid authentication value. Thus, we have

- Assume that U_i^l is in expected state, then for a $Send(U_i^l, \mathbf{start})$ query, U_i^l responds as follows:
  - Choose $r_i \in_R Z_p^*$ and compute $C_1 = T_{r_i}(X) \mod p$, $C_2 = T_{r_i}(R) \mod p$, $UID_i = ID_i \oplus h(C_1 \| C_2)$, $f_i = B_i \oplus e_i$, $M_{ik} = h(ID_i \| SID_j \| f_i \| C_1 \| C_2)$.
  - Output the message $\{UID_i, C_1, M_{ik}\}$.
- Assume that S_j^m is in expected state, then for a $Send(S_j^m, \{UID_i, C_1, M_{ik}\})$ query, S_j^m answers as follows:
  - Choose $r_j \in_R Z_p^*$ and compute $C_3 = T_{r_j}(X) \mod p$, $M_{jk} = h(K_j \| C_3)$.
  - Output the message $\{UID_i, C_1, M_{ik}, SID_j, C_3, M_{jk}\}$.
- Assume that RC^n is in expected state, then for a $Send(RC^n, \{UID_i, C_1, M_{ik}, SID_j, C_3, M_{jk}\})$ query, RC^n responds as follows:
  - Calculate $K_j' = h(SID_j \| w)$, $M_{jk}' = h(K_j' \| C_3)$, and check whether $M_{jk}' \stackrel{?}{=} M_{jk}$.
    - If $M_{jk}' \neq M_{jk}$, reject the message $\{UID_i, C_1, M_{ik}, SID_j, C_3, M_{jk}\}$.
    - Else, compute $C_2' = T_w(C_1) \mod p$, $ID_i' = UID_i \oplus h(C_1 \| C_2')$, $f_i' = h(ID_i' \| w)$, $M_{ik}' = h(ID_i' \| SID_j \| f_i' \| C_1 \| C_2')$, and check $M_{ik}' \stackrel{?}{=} M_{ik}$.
      - If $M_{ik}' \neq M_{ik}$, reject the message $\{UID_i, C_1, M_{ik}, SID_j, C_3, M_{jk}\}$.
      - Else, choose $r_k \in_R Z_p^*$ and compute $D_i = K_j' \oplus r_k$, $M_{kj} = h(K_j' \| C_1 \| C_3 \| r_k)$, $E_i = h(K_j' \| r_k)$, $F_i = E_i \oplus f_i' \oplus C_2'$, $M_{ki} = h(f_i' \| F_i \| SID_j \| C_2' \| C_3)$.
      - Output the message $\{D_i, M_{kj}, F_i, M_{ki}\}$.
- Assume that S_j^m is in expected state, then for a $Send(S_j^m, \{D_i, M_{kj}, F_i, M_{ki}\})$, S_j^m responds as follows:
  - Calculate $r_k' = K_j \oplus D_i$, $M_{kj}' = h(K_j \| C_1 \| C_3 \| r_k')$ and check $M_{kj}' \stackrel{?}{=} M_{kj}$.
    - If $M_{kj}' \neq M_{kj}$i, reject the message $\{D_i, M_{kj}, F_i, M_{ki}\}$).
    - Else, compute $E_i' = h(K_j \| r_k')$, $SK = T_{r_j}(C_1) \mod p$, $M_{ji} = h(SID_j \| C_1 \| C_3 \| E_i' \| SK)$.
    - Output the message $\{F_i, M_{ki}, C_3, M_{ji}\}$.
- Assume that U_i^l is in expected state, then for a $Send(U_i^l, \{F_i, M_{ki}, C_3, M_{ji}\})$, U_i^l responds as follows:
  - Calculate $M_{ki}' = h(f_i \| F_i \| SID_j \| C_2 \| C_3)$, and check $M_{ki}' \stackrel{?}{=} M_{ki}$.
    - If $M_{ki}' \neq M_{ki}$, the message $\{F_i, M_{ki}, C_3, M_{ji}\}$ is rejected.
    - Else, compute $E_i'' = F_i \oplus f_i \oplus C_2$, $SK' = T_{r_i}(C_3) \mod p$, $M_{ji}' = h(SID_j \| C_1 \| C_3 \| E_i'' \| SK')$, and check $M_{ji}' \stackrel{?}{=} M_{ji}$.
      - If $M_{ji}' \neq M_{ji}$, the message $\{F_i, M_{ki}, C_3, M_{ji}\}$ is rejected.
      - Else, calculate $M_{ij} = h(C_3 \| E_i'' \| SK')$.
      - Output the message $\{M_{ij}\}$.
- Assume that S_j^m is in expected state, then for a $Send(S_j^m, \{M_{ij}\})$, S_j^m responds as follows::
  - Calculate $M_{ij}' = h(C_3 \| E_i' \| SK)$, and check $M_{ij}' \stackrel{?}{=} M_{ij}$.
    - If $M_{ij}' \neq M_{ij}$, the message $\{M_{ij}\}$ is rejected.
    - Else, the message $\{M_{ij}\}$ is accepted.

Finally, both U_i^l and S_j^m accept and then successfully terminate the session.

Fig. 4 Simulation of send oracle queries

- The $Execute(U_i^l, S_j^m, RC^n)$ query is simulated with the $Send$ query in the following ways:
  - $\{UID_i, C_1, M_{ik}\} \leftarrow Send(U_i^l, \textbf{start})$
  - $\{UID_i, C_1, M_{ik}, SID_j, C_3, M_{jk}\} \leftarrow Send(S_j^m, \{UID_i, C_1, M_{ik}\})$
  - $\{D_i, M_{kj}, F_i, M_{ki}\} \leftarrow Send(RC^k, \{UID_i, C_1, M_{ik}, SID_j, C_3, M_{jk}\})$
  - $\{F_i, M_{ki}, C_3, M_{ji}\} \leftarrow Send(S_j^m, \{D_i, M_{kj}, F_i, M_{ki}\})$
  - $\{M_{ij}\} \leftarrow Send(U_i^l, \{F_i, M_{ki}, C_3, M_{ji}\})$

  Finally, this query outputs $\{UID_i, C_1, M_{ik}\}$, $\{UID_i, C_1, M_{ik}, SID_j, C_3, M_{jk}\}$, $\{D_i, M_{kj}, F_i, M_{ki}\}$, $\{F_i, M_{ki}, C_3, M_{ji}\}$ and $\{M_{ij}\}$.

Fig. 5 Simulation of execute oracle queries

- The $Corrupt(P^t, a)$ query can be simulated as follows:
  - If $a = 1$, it outputs PW_i of U_i.
  - Else, it returns the secret information stored on the smart card.

Fig. 6 Simulation of corrupt oracle queries

- The $Reveal(P^t)$ is simulated as follows:
  - If P^t has accepted, it outputs the session key SK accepted between P^t and its partner.
  - Else, it returns a *null* value.

Fig. 7 Simulation of reveal oracle queries

- The $Test(P^t)$ query can be simulated as follows:
  - It obtains SK from the $Reveal(P^t)$ query and flip an unbiased coin b.
    - If $b = 1$, return SK.
    - Else, return a random value from $\{0, 1\}^*$.

Fig. 8 Simulation of test oracle queries

$$|Pr[E_3] - Pr[E_2]| \leq \frac{q_s}{2^l} \tag{5}$$

Game G_4: In this game, CDH problem is simulated. For this purpose, $\mathcal{A}$ queries *CDH* oracle with $\{X, \ T_a(X) \bmod p, \ T_b(X) \bmod p\}$ to compute $CDH(T_a(X) \bmod p, T_b(X) \bmod p) = T_{ab}(X) \bmod p$, where $a, b \in Z_p^*$. The session key is guessed without asking the corresponding oracle $h(\cdot)$, i.e. SK is completely independent from $h(\cdot)$ and $T_{r_j}(C_1) \bmod p$ or $T_{r_i}(C_3) \bmod p$. Therefore, this game and the previous game are indistinguishable unless $\mathcal{A}$ queries $h(\cdot)$ on the common value $T_{r_i r_j}(X) \bmod p$. Hence, $Adv_{C,\mathcal{A}}(p) \geq \frac{1}{q_h} \cdot |Pr[E_4] - Pr[E_3]|$ i.e., we have

$$|Pr[E_4] - Pr[E_3]| \leq q_h \cdot Adv_{C,\mathcal{A}}(t) \tag{6}$$

Game G_5: The simulation of this game is identical with the simulation of the game G_4. However, the *Test* query of this game is terminated if $\mathcal{A}$ asks a $h(\cdot)$ query with $T_{r_i r_j}(X) \bmod p$. $\mathcal{A}$ can get the session key SK by $h(\cdot)$ query with probability at most $\frac{q_h^2}{2p}$. Thus we have,

$$|Pr[E_5] - Pr[E_4]| \leq \frac{q_h^2}{2^{l+1}} \tag{7}$$

The adversary $\mathcal{A}$ will not have any advantage in distinguishing the real session key from a random one if he doesn't make any $h(\cdot)$ query with the correct input. Therefore, $Pr[E_5] = \frac{1}{2}$. Note that two *Corrupt* cannot be executed by $\mathcal{A}$. It means that if $\mathcal{A}$ executes the $Corrup(U_i, 2)$ query, however, he is not allowed to ask the password-corrupt query $(Corrup(U_i, 1))$. The success probability of $\mathcal{A}$ against the off-line password guessing attack is at most $\frac{q_s}{|\mathcal{D}|}$. Adding the Eqs. (2–7), we have

$$Adv_{\mathbb{P}}(\mathcal{A}) \leq \frac{q_h^2}{2^{l-1}} + \frac{(q_s + q_e)^2}{p} + \frac{q_s}{2^{l-1}} + 2 \cdot (q_h \cdot Adv_{C,\mathcal{A}}(t) + \frac{q_s}{|\mathcal{D}|})$$

□

6.2 Authentication Proof Based on the BAN Logic Model

The BAN logic model [43], a well-known formal security validation tool, plays a very important role in formal analysis to validate the security features of cryptographic protocols. In the following, we first show some basic notations and rules about BAN logic, where A and B are two entities involved in the protocol, and M, N are two messages, and K is a secret key.

- $A \mid\equiv M$: A believes M, i.e. A believes M is true.
- $A \triangleleft M$: A see M, i.e., a message containing M is sent to A by someone and M can be read by A.
- $A \mid \sim M$: A once said M, i.e., a message containing M once sent by A in some time.
- $A \Rightarrow M$: A control M, i.e., M is subject to the jurisdiction of A, and A is trusted for M.
- $\#(M)$: M is fresh, i.e., there is no any entity sent a message containing M at any time before the current session of protocol.
- $A \overset{K}{\longleftrightarrow} B$: K is shared between A and B, and can be used to secure the communication between them. K is called secure if any other entity cannot get K except A, B and the entity who is trusted by A or B.
- (M, N): M or N is a portion of message (M, N).
- $\{M, N\}_K$: M and N are encrypted with the key K.
- $(M, N)_K$: M and N are hashed by the key K.
- **Rule 1**. Message meaning rule: $\frac{A \mid\equiv A \overset{K}{\longleftrightarrow} B, A \triangleleft \{M\}_K}{A \mid\equiv B \mid\sim M}$, if A believes that he/she shared the key K with B, and A saw the message $\{M\}_K$, A trust that B once said X.
- **Rule 2**. Nonce verification rule: $\frac{A \mid\equiv \#(M), A \mid\equiv B \mid\sim M}{A \mid\equiv B \mid\equiv M}$, if A believes M is fresh and A believes B once said M, A believes B believes M.
- **Rule 3**. Jurisdiction rule: $\frac{A \mid\equiv B \Rightarrow M, A \mid\equiv B \mid\equiv M}{A \mid\equiv M}$, if A believes that B had jurisdiction right to message M and A believes B believes M, A believes M.
- **Rule 4**. Freshness conjuncatenation rule: $\frac{A \mid\equiv \#(M)}{A \mid\equiv \#(M, N)}$, if M as a portion of message (M, N) is fresh, message (M, N) is also fresh.
- **Rule 5**. Belief rule: $\frac{A \mid\equiv B \mid\equiv (M, N)}{A \mid\equiv B \mid\equiv (M)}$, if A believes B believes the message set (M, N), A also believes B believes the message M.

BAN logic [43] had widely be used to analyze the correctness of authentication with key agreement protocol. Protocol correctness means that both of the communication parties confirm that they shared a fresh session key with each other after mutual authentication.

Thus, the user authentication with key agreement protocol should achieve the following goals:

- G 1: $U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$.
- G 2: $U_i \mid\equiv S_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$.
- G 3: $S_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$.
- G 4: $S_j \mid\equiv U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j)$.

First, the idealized form of the messages in our protocol are transformed according to the BAN logic notations:

- Message 1: $U_i \rightarrow RC$: $(ID_i, SID_j, U_i \overset{\{X\}_{w \cdot r_i}}{\longleftrightarrow} RC)_{h(ID_i \| w)}$
- Message 2: $S_j \rightarrow RC$: $(S_j \overset{h(SID_j \| w)}{\longleftrightarrow} RC)_{h(SID_j \| w)}$
- Message 3: $RC \rightarrow S_j$: $(C_1, C_3, S_j \overset{r_k}{\longleftrightarrow} RC)_{h(SID_j \| w)}$
- Message 4: $RC \rightarrow U_i$: $(SID_j, C_2, C_3, U_i \overset{\{X\}_{w \cdot r_i}}{\longleftrightarrow} RC)_{h(ID_i \| w)}$
- Message 5: $S_j \rightarrow U_i$: $(SID_j, C_1, U_i \overset{SK}{\longleftrightarrow} S_j)_{h(K_j \| r_k)}$
- Message 6: $U_i \rightarrow S_j$: $(C_3, U_i \overset{SK}{\longleftrightarrow} S_j)_{h(K_j \| r_k)}$

Second, some initial assumptions of our protocol are given below:

$$
\begin{aligned}
&A_1.\ U_i \mid\equiv \#(r_i) && A_8.\ S_j \mid\equiv S_j \overset{h(SID_j \| w)}{\longleftrightarrow} RC \\
&A_2.\ S_j \mid\equiv \#(r_j) && A_9.\ RC \mid\equiv S_j \overset{h(SID_j \| w)}{\longleftrightarrow} RC \\
&A_3.\ RC \mid\equiv \#(r_k) && A_{10}.\ S_j \mid\equiv U_i \overset{h(K_j \| r_k)}{\longleftrightarrow} S_j \\
&A_4.\ U_i \mid\equiv \#(C_1) && A_{11}.\ U_i \mid\equiv U_i \overset{h(K_j \| r_k)}{\longleftrightarrow} S_j \\
&A_5.\ S_j \mid\equiv \#(C_3) && A_{12}.\ U_i \mid\equiv S_j \Rightarrow (U_i \overset{SK}{\longleftrightarrow} S_j) \\
&A_6.\ U_i \mid\equiv U_i \overset{h(ID_i \| w)}{\longleftrightarrow} RC && A_{13}.\ S_j \mid\equiv U_i \Rightarrow (U_i \overset{SK}{\longleftrightarrow} S_j) \\
&A_7.\ RC \mid\equiv U_i \overset{h(ID_i \| w)}{\longleftrightarrow} RC
\end{aligned}
$$

A_1, A_2 and A_3 mean that U_i, S_j and RC generate fresh random numbers r_i, r_j and r_k, respectively. Therefore, they assure their freshness, respectively. Since $C_1 \equiv T_{r_i}(X) \bmod p$, $C_3 \equiv T_{r_j}(X) \bmod p$, A_4 and A_5 are valid according to A_1 and A_2. $A_6 - A_9$ are hold because the secret key is shared by one party and can be generate by the other party. With the assumptions A_{10} and A_{11}, the secret $h(K_j \| r_k)$ can be generated by U_i and S_j from message 4 and message 3, respectively, and thus RC is trusted by both U_i and S_j. Therefore, any other party cannot get the secret $h(K_j \| r_k)$ and the assumptions A_{10} and A_{11} are hold. The assumption A_{12} (A_{13}) is also hold because once U_i (S_j) generates r_i (r_j) and submits C_1 (C_3) to S_j (U_i), the final session key SK is determined by the random number r_j (r_i), which is generated by S_j (U_i) from the viewpoint of U_i (S_j).

Then, we prove the idealized form of the proposed protocol using the initial assumptions and the rules of the BAN logic. The detailed descriptions of this are as follows:

In accordance with the message 5, we can get

S_1: $U_i \triangleleft (SID_j, C_1, U_i \overset{SK}{\longleftrightarrow} S_j)_{h(K_j \| r_k)}$.

In accordance with S_1 and A_{11}, the following judgement can be obtained by applying the message meaning rule

S_2: $U_i \mid\equiv S_j \mid \sim (SID_j, C_1, U_i \overset{SK}{\longleftrightarrow} S_j)$.

In accordance with S_2 and A_4, the following judgement can be derived by applying the freshness conjuncatenation rule

S_3: $U_i \mid\equiv \#(SID_j, C_1, U_i \overset{SK}{\longleftrightarrow} S_j)$.

In accordance with S_2 andS_3, the following judgement can be obtained by applying the nonce verification rule

S_4: $U_i \mid\equiv S_j \mid\equiv (SID_j, C_1, U_i \overset{SK}{\longleftrightarrow} S_j)$.

In accordance with S_4, the following judgement can be derived by applying belief rule

$$S_5 : U_i \mid\equiv S_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j). \tag{G2}$$

In accordance with S_5 and A_{12}, the following judgement can be obtained by applying jurisdiction rule

$$S_6 : U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j). \tag{G1}$$

In accordance with message 6, we can get

S_7: $S_j \triangleleft (C_3, U_i \overset{SK}{\longleftrightarrow} S_j)_{h(K_j \| r_k)}$.

In accordance with S_7 and A_{10}, the following judgement can be obtained by applying the message meaning rule

S_8: $S_j \mid\equiv U_i \sim (C_3, U_i \overset{SK}{\longleftrightarrow} S_j)$.

In accordance with S_8 and A_5, the following judgement can be derived by applying the freshness conjuncatenation rule

S_9: $S_j \mid\equiv \#(C_3, U_i \overset{SK}{\longleftrightarrow} S_j)$.

In accordance with S_8 andS_9, the following judgement can be obtained by applying the nonce verification rule

S_{10}: $S_j \mid\equiv U_i \mid\equiv (C_3, U_i \overset{SK}{\longleftrightarrow} S_j)$.

In accordance with S_{10}, the following judgement can be obtained by applying belief rule

$$S_{11} : S_j \mid\equiv U_i \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j). \tag{G4}$$

In accordance with S_{11} and A_{13}, the following judgement can be obtained by applying jurisdiction rule

$$S_{12} : S_j \mid\equiv (U_i \overset{SK}{\longleftrightarrow} S_j). \tag{G3}$$

From the judgements S_6, S_5, S_{12} and S_{11}, we can see that our protocol can achieve all the goals 1, 2, 3 and 4, and both of U_i and S_j believe that they shared a session key $SK \equiv T_{r_i r_j}(X) \bmod p$ with each other.

6.3 Further Discussions on Various Attacks

In this part, we demonstrated that our protocol meets all known functional properties and withstands various attacks.

6.3.1 Resist Insider Attack

In the registration stage of our protocol, U_i submits the registration request $\{ID_i, e_i = h(PW_i \| N_i)\}$ to RC for registration, where the actual password PW_i is masked by the random number N_i. Therefore, the privileged insider of RC cannot reveal PW_i of U_i without knowing N_i. Thus, the proposed protocol is free from insider attack.

6.3.2 Quick Detection of Unauthorized Login

In the login and session key agreement stage of the proposed protocol, U_i keys ID_i, PW_i, and chooses the identity SID_j of S_j. Then the smart card calculates $e_i = h(PW_i \| N_i)$, $A_i' = h(ID_i \| e_i)$ and checks $A_i' \stackrel{?}{=} A_i$. If $A_i' \neq A_i$, it means one of the inputted user's identity or password is invalid, and thus the login request is aborted by the smart card. Otherwise, it is asserted that the inputted identity and password are correct. Therefore, in our protocol, any unauthorized login due to wrong password or identity can be quickly detected in the smart card side, which is helpful to avoid network congestion and DoS attack.

6.3.3 Mutual Authentication Among Three Parties

In our protocol, RC is not only responsible for the initialization of the (i) system's parameters, (ii) user registration and (iii) server registration, but also participates in the authentication process, which is needed in the multi-server architecture to enhanced the security of the system. In login and session key agreement stage of our protocol, U_i, S_j and RC can authenticate each other.

- *Mutual authentication between S_j and RC*: In step 1 of login and session key agreement stage, U_i's login request $\{UID_i, C_1, M_{ik}\}$ was transmitted to the target server S_j. Then S_j calculates some information and forwards the login request message $\{UID_i, C_1, M_{ik}, SID_j, C_3, M_{jk}\}$ to RC. Then RC can compute $K_j' = h(SID_j \| w)$, $M_{jk}' = h(K_j' \| C_3)$ and authenticates S_j by checking whether M_{jk}' equals to M_{jk}. On the contrary, when receiving RC's response message $\{ID_i, M_{kj}, F_i, M_{ki}\}$ in step 4 of login and session key agreement stage, S_j derives $r_k' = K_j \oplus D_i$, $M_{kj}' = h(K_j \| C_1 \| C_3 \| r_k')$ and checks whether $M_{kj}' = M_{kj}$. If they are equal, the validity of RC is authenticated by S_j. Since the shared secret key $K_j = h(SID_j \| w)$ between S_j and RC is contained in their exchanged messages, any other entity cannot forged the valid messages without the secret information $h(SID_j \| w)$. Therefore, mutual authentication between S_j and R_C is achieved.
- *Mutual authentication between U_i and RC* In our protocol, the mutual authentication between U_i and RC relies on the shared secret information $f_i = h(ID_i \| w)$. In step 1 of login and session key agreement stage, U_i's login request $\{UID_i, C_1, M_{ik}\}$ was transmitted to S_j. Then S_j calculates some information and forwards the login request message $\{UID_i, C_1, M_{ik}, SID_j, C_3, M_{jk}\}$ to RC. RC derives $C_2' \equiv T_w(C_1) \mod p$, $ID_i' = UID_i \oplus h(C_1 \| C_2')$, $f_i' = h(ID_i' \| w)$, $M_{ik}' = h(ID_i' \| SID_j \| f_i' \| C_1 \| C_2')$ and checks

$M'_{ik} \stackrel{?}{=} M_{ik}$. The validity of U_i is authenticated by RC if M'_{ik} equals to M_{ik}. On the contrary, RC sends a response message $\{D_i, M_{kj}, F_i, M_{ki}\}$ to S_j, and S_j forwards the mutual authentication message $\{F_i, M_{ki}, C_3, M_{ji}\}$ to U_i in step 4. U_i computes $M'_{ki} = h(f_i \| F_i \| SID_j \| C_2 \| C_3)$ and checks $M'_{ki} \stackrel{?}{=} M_{ki}$. If $M'_{ki} = M_{ki}$, the validity of RC is authenticated by U_i since $f_i = h(ID_i \| w)$ is only known to U_i and RC and any other entity cannot forged the valid messages exchanged between U_i and RC. Therefore, U_i and RC mutually authenticate each other in the proposed protocol.

- *Mutual authentication between U_i and S_j*: In step 2 of login and session key agreement stage of our protocol, S_j gets C_1 from U_i's login request message $\{UID_i, C_1, M_{ik}\}$. In step 4, S_j computes $r'_k = K_j \oplus D_i$, $E'_i = h(K_j \| r'_k)$, $SK \equiv T_{r_j}(C_1) \mod p$, $M_{ji} = h(SID_j \| C_1 \| C_3 \| E'_i \| SK)$ and forwards the response message $\{F_i, M_{ki}, C_3, M_{ji}\}$ to U_i. Then in step 5, U_i computes $E''_i = F_i \oplus f_i \oplus C_2$, $SK' \equiv T_{r_i}(C_3) \mod p$, $M'_{ji} = h(SID_j \| C_1 \| C_3 \| E''_i \| SK')$ and authenticates S_j by checking $M'_{ji} \stackrel{?}{=} M_{ji}$. If they are equal, S_j is authenticated by U_i, then U_i computes $M_{ij} = h(C_3 \| E''_i \| SK')$ and submits the mutual authentication message $\{M_{ij}\}$ to S_j. When receiving $\{M_{ij}\}$, S_j computes $M'_{ij} = h(C_3 \| E'_i \| SK)$ and U_i is authenticated by S_j if $M'_{ij} = M_{ij}$. Since $SK \equiv T_{r_i r_j}(X) \mod p$ is shared between U_i and S_j, and any other entity cannot forged the valid messages exchanged between U_i and S_j without knowing SK. Thus, our protocol achieved mutual authentication between U_i and S_j.

6.3.4 Perfect Forward Secrecy

It is an valuable security feature for a password-based authentication with the session key agreement protocol. This property includes that the adversary cannot derive the previously established session keys even if he/she gets the system master secret key w. In the proposed protocol, U_i and S_j can share a session key $SK \equiv T_{r_i r_j}(X) \mod p$ after mutual authentication, where SK has no relationship with the master secret key w. However, SK is only related to the random number r_i and r_j chosen by U_i and S_j, respectively. Besides, the attacker cannot compute r_i, r_j and $T_{r_i r_j}(X)$ from C_1, C_2 due to the discrete logarithm problem (DLP) or Diffie-Hellman problem (DHP). Therefore, the proposed protocol can provide perfect forward secrecy.

6.3.5 Known-Key Security

Known-key security is another important security feature for a password-based authentication with the session key agreement protocol. It states that the compromise of one session key should not be helpful to do harm for other session keys. In the proposed protocol, the session key $SK \equiv T_{r_i r_j}(X) \mod p$ relies on the random numbers r_i and r_j. Since the random numbers r_i and r_j are dynamically changing in each session, one session key is independent of the other session keys. Therefore, even if one session key $SK \equiv T_{r_i r_j}(X) \mod p$ is compromised, the attacker cannot get the other session key $SK' \equiv T_{r'_i r'_j}(X) \mod p$ without knowing r'_i and r'_j. Thus, the proposed protocol can provide known-key security.

6.3.6 Resist Registration Center and Server Spoofing Attacks

In [33], *RC* shares the master secret key w with all the servers, which not only increases the security risk of the authentication system, but also leads to the vulnerability of registration center and sever spoofing attacks. In the proposed protocol, the master secret key w is only held by *RC*. Since w is long enough and safeguarded by the secure hash function $h(\cdot)$, the adversary could not drive w from the communication messages transmitted over the public channel. Therefore, the attacker cannot impersonate *RC*, and accordingly, the registration center spoofing attack can be avoided in our protocol. In addition, each S_j holds a secret key $h(SID_j \| w)$, which is different from those of other server's. Therefore, without knowing $h(SID_j \| w)$, any attacker or even the legal but malicious server cannot impersonate other servers, and therefore, the proposed protocol can resist server spoofing attack.

6.3.7 Resist Replay Attack

In our protocol, U_i, S_j and *RC* generate different random numbers r_i, r_j and r_k, respectively, for each session, which ensures that the messages belonging to current session are valid only for this session, and different with the messages of other session's. Therefore, the attacker has no chance to launch a replay attack to the proposed protocol, and our protocol is free from this kind of attack.

6.3.8 User Anonymity

In the proposed protocol, the masked identity $UID_i (= ID_i \oplus h(C_1 \| C_2))$ replaced the real identity ID_i to be contained in the login request message $\{UID_i, C_1, M_{ik}\}$. Note that *RC* holds the master secret key w and only he/she can derive U_i's real identity ID_i by computing $C_2' \equiv T_w(C_1) \mod p$, and $ID_i' = UID_i \oplus h(C_1 \| C_2')$. Any attacker who wants to reveal the real identity of U_i has to face the DLP problem. Therefore, user anonymity is achieved in our protocol.

6.3.9 Resist Stolen Smart Card Attack

Smart card is widely in two-factor password-based user authentication. However, some studies have shown that the information stored in the smart card can be derived by some side channel attacks. Therefore, the stolen smart card attack should be considered in designing the authentication protocol. In this attack, the adversary captured a lost/stolen smart card of an authorized and then he/she may try to changed or guessed the password, or may try to utilized the smart card to login to the system to impersonate the user. In our protocol, if U_i's lost/stolen smart card was captured by an adversary, then he/she can derive the information $\{A_i, B_i, X, R, h(\cdot), N_i\}$ from the smart card, where $A_i = e_i \oplus f_i$, $e_i = h(PW_i \| N_i)$, $f_i = h(ID_i \| w)$, and $B_i = h(ID_i \| e_i)$. We can see that A_i and B_i are both related to ID_i and PW_i, therefore, it is impossible for the adversary to guess ID_i and PW_i from the smart card's information. Without knowing ID_i and PW_i, the adversary cannot guess or change the password, and thus, he/she is unable to impersonate U_i. Therefore, our protocol is free from stolen smart card attack.

7 Comparisons on Functionality and Performance

In this subsection, the evaluations of the proposed protocol are given by comparing it with the protocol in [33] on functionality and performance aspects. The functionality comparison between our protocol and the protocol in [33] is listed in Table 2, from which we can assert that our protocol is more ideal and secure than the protocol in [33], because the proposed protocol can resist most of attacks and has additionally functionalities.

For performance evaluation, we have considered the followings:

- T_H : Time required to compute hash function.
- T_C : Time required to compute extended chaotic function.

Since the login and session key agreement phases should be executed in every time when the user access different servers, we only compare the computation costs of these phases in the comparison. The comparison results of our protocol and the protocol in [33] are summarized in Table 3. Table 3 shows that the presented protocol needs 8 additional hash operations than the protocol in [33]. Based on the system configuration (1) 3GB RAM and (2) Pentium IV 3.2 GHz processor, the experimental results executed in [52] and we have $t_H \approx 0.20$ ms and $t_C \approx 32.40$ ms, respectively. We compared our protocol and the protocol in [33] against the execution time (millisecond) in Table 4.

In the proposed protocol, the registration center RC is involved in the authentication process, which helps avoid server and registration center spoofing attacks. Besides, our protocol owns a mechanism to verify the validity of the password in the beginning of login and session key agreement stage, which not only makes our protocol efficient to quickly detect the unauthorized login, but also allows users to freely change the password. Therefore, the addition of 8 hash operations is worthy to make our protocol more secure and achieve more ideal functions.

Table 2 Functionality comparisons

Attribute	The protocol in [33]	The presented protocol
Single registration	No	Yes
Resist server spoofing attack	No	Yes
Resist registration center spoofing attack	No	Yes
Quickly detection of unauthorized login	No	Yes
Support password change	No	Yes
Resist insider attack	Yes	Yes
Resist replay attack	Yes	Yes
Perfect forward secrecy	Yes	Yes
Known-key security	Yes	Yes
User anonymity	Yes	Yes
BAN logic analysis	Yes	Yes
Provable security analysis	No	Yes

Table 3 Performance comparisons in login and session key agreement stage

Computation cost	The protocol in [33]	The presented protocol
User costs	$3T_C + 5T_H$	$3T_C + 7T_H$
Server costs	$3T_C + 6T_H$	$2T_C + 4T_H$
RC costs	–	$1T_C + 8T_H$
Total costs	$6T_C + 11T_H$	$6T_C + 19T_H$

Table 4 Execution time (ms) comparisons in login and session key agreement stage

Computation cost	The protocol in [33] (ms)	The presented protocol (ms)
User costs	108.20	108.60
Server costs	108.40	65.60
RC costs	–	34.00
Total costs	197.60	198.00

8 Conclusion and Future Work

In this paper, we reviewed and analyzed the security drawbacks of an extended chaotic maps-based authentication multi-server protocol in [33]. We find the protocol in [33] is vulnerable to registration center and server spoofing attacks. These two attacks are possible due to the master secret key w is shared among *RC* and all servers. Additionally, the protocol in [33] does not support password change for users. Besides, there is no detection mechanism for unauthorized login in the protocol in [33], which causes unnecessary computations and communications, and induces DOS attack. To remedy the aforementioned drawbacks, a new extend chaotic map based multi-server authentication with key agreement protocol has been presented in this paper, where the user can enjoy different services provided by multiple servers with one time registration. The provable security analysis of the presented protocol is provided in the random oracle model. In addition, the formal security of our protocol is also validated using BAN logic model. Besides, the other functional features and security of presented protocol are discussed, and it only increase 8 hash operations to remedy the aforementioned flaws. Compared with the protocol in [33], the presented protocol is a more secure and ideal chaotic maps-based multi-server authentication protocol for practical use. In the future, we would like to establish a security model for chaotic based authentication and key agreement protocol, and design the authentication schemes with key agreement under this model. Furthermore, we will consider the privacy protection for the design of security protocols, and explore the effective method for user authentication and key agreement with privacy protection.

Acknowledgments This work was supported by the National Natural Science Foundation of China under Grant Nos. 61300220, 61572013 and 61572188, the General and Special Financial Grant from China Postdoctoral Science Foundation under Grant Nos. 2014M550590 and 2015T80035, respectively. SK Hafizul Islam is partially supported by OPERA Award, BITS Pilani, India. Fan Wu is supported by Fujian Education and Scientific Research Program for Young and Middle-aged Teachers under Grant No. JA14369. Besides, the authors extend their sincere appreciations to the Deanship of Scientific Research at King Saud University for its funding this Prolific Research Group (PRG-1436-16).

References

1. Lamport, L. (1981). Password authentication with insecure communication. *Communications of the ACM*, *24*(11), 770–772.
2. Horng, G. (1995). Password authentication without using a password table. *Information Processing Letters*, *550*(5), 247–250.
3. Jan, J. K., & Chen, Y. Y. (1998). Paramita wisdom password authentication scheme without verification tables. *Journal of Systems and Software*, *42*(1), 45–57.
4. He, D. B., Kumar, N., & Naveen, C. (2015). A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks. *Information Sciences*, *321*, 263–277.
5. He, D. B., & Wang, D. (2015). Robust biometrics-based authentication scheme for multi-server environment. *IEEE Systems Journal*, *9*(3), 816–823.
6. He, D. B., Kumar, N., Chen, J. H., Lee, C. C., Chilamkurti, N., & Yeo, S. S. (2015). Robust anonymous authentication protocol for healthcare applications using wireless medical sensor networks. *Multimedia Systems*, *21*(1), 49–60.
7. Li, X., Niu, J. W., Khan, M. K., & Liao, J. G. (2013). An enhanced smart card based remote user password authentication scheme. *Journal of Network and Computer Applications*, *36*(5), 1365–C1371.
8. Li, X., Niu, J. W., Ma, J., Wang, W. D., & Liu, C. L. (2011). Cryptanalysis and improvement of a biometrics-based remote user authentication scheme using smart cards. *Journal of Network and Computer Applications*, *34*(1), 73–79.
9. Wang, R. C., Juang, W. S., & Lei, C. L. (2011). Robust authentication and key agreement scheme preserving the privacy of secret key. *Computer Communications*, *34*(3), 274–280.
10. Chang, Y. F., Tai, W. L., & Chang, H. C. (2014). Untraceable dynamic-identity-based remote user authentication scheme with verifiable password update. *International Journal of Communication Systems*, *27*(11), 3430–3440.
11. Li, X., Niu, J. W., Liao, G. J., & Liang, W. (2015). Cryptanalysis of a dynamic identity-based remote user authentication scheme with verifiable password update. *International Journal of Communication Systems*, *28*(2), 374–382.
12. Liu, Y. C., Gong, P., Yan, X. P., & Li, P. (2015). On the security of a dynamic identity-based remote user authentication scheme with verifiable password update. *International Journal of Communication Systems*, *28*(5), 842–847.
13. Li, L. H., Lin, I. C., & Hwang, M. S. (2001). A remote password authentication scheme for multi-server architecture using neural networks. *IEEE Transactions on Neural Network*, *12*(6), 1498–1504.
14. Lin, I. C., Hwang, M. S., & Li, L. H. (2003). A new remote user authentication scheme for multi-server architecture. *Future Generation Computer Systems*, *19*(1), 13–22.
15. Juang, W. S. (2004). Efficient multi-server password authenticated key agreement using smart cards. *IEEE Transactions on Consumer Electronics*, *50*(1), 251–255.
16. Chang, C. C., & Lee, J. S. (November 2004). An efficient and secure multi-server password authentication scheme using smart cards. In *Proceedings of the third international conference on cyberworlds*, (pp. 417–422).
17. Tsaur, W. J., Wu, C. C., & Lee, W. B. (2004). A smart card-based remote scheme for password authentication in multi-server Internet services. *Computer Standards & Interfaces*, *27*(1), 39–51.
18. Tsaur, W. J., Wu, C. C., & Lee, W. B. (2005). An enhanced user authentication scheme for multi-server Internet services. *Applied Mathematics and Computation*, *170*(1), 258–266.
19. Tsai, J. L. (2008). Efficient multi-server authentication scheme based on one-way hash function without verification table. *Computers and Security*, *27*(3–4), 115–121.
20. Yoon, E. J., & Young, Y. K. (2013). Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem. *Journal of Supercomputing*, *63*(1), 235–255.
21. Liao, Y. P., & Hsiao, C. M. (2013). A novel multi-server remote user authentication scheme using self-certified public keys for mobile clients. *Future Generation Computer Systems*, *29*(3), 886–900.
22. Lee, Y. S., Kim, E., Seok, S. J., & Jung, M. S. (2012). A smart card-based user authentication scheme to ensure the PFS in multi-server environments. *IEICE Transactions on Communications*, *E95–B*(2), 619–622.
23. He, D. B. (2011). Security flaws in a biometrics-based multi-server authentication with key agreement scheme. *IACR Cryptology ePrint Archive*, 2011/365.
24. Chou, J. S., Chen, Y., Huang, C. H., & Huang, Y. S. (2012). Comments on four multi-server authentication protocols using smart card, *IACR Cryptology ePrint Archive*, 2012/406.

25. He, D. B., & Hu, H. (2012). Cryptanalysis of a smart card-based user authentication scheme for multi-server environments. *IEICE Transactions on Communications*, *E95–B*(9), 3052–3054.
26. Liao, Y. P., & Wang, S. S. (2009). A secure dynamic ID based remote user authentication scheme for multi-server environment. *Computer Standard & Interfaces*, *31*(1), 24–29.
27. Hsiang, H. C., & Shih, W. K. (2009). Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment. *Computer Standard & Interfaces*, *31*(6), 1118–1123.
28. Sood, S. K., Sarje, A. K., & Singh, K. (2011). A secure dynamic identity based authentication protocol for multi-server architecture. *Journal of Network and Computer Applications*, *34*(2), 609–618.
29. Lee, C. C., Lin, T. H., & Chang, R. X. (2011). A secure dynamic ID based remote user authentication scheme for multi-server environment using smart cards. *Expert Systems with Applications*, *38*(11), 13863–13870.
30. Li, X., Xiong, Y. P., Ma, J., & Wang, W. D. (2012). An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards. *Journal of Network and Computer Applications*, *35*(2), 763–769.
31. Li, X., Ma, J., Wang, W. D., Xiong, Y. P., & Zhang, J. S. (2013). A novel smart card and dynamic ID based remote user authentication scheme for multi-server environment. *Mathematical and Computer Modelling*, *58*(1–2), 85–95.
32. Tsaur, W. J., Li, J. H., & Lee, W. B. (2012). An efficient and secure multi-server authentication scheme with key agreement. *Journal of Systems and Software*, *85*(4), 876–882.
33. Lee, C. C., Lou, D. C., Li, C. T., & Hsu, C. W. (2014). An extended chaotic-maps-based protocol with key agreement for multiserver environments. *Nonlinear Dynamics*, *76*(1), 853–866.
34. Li, C. T., Lee, C. C., & Weng, C. Y. (2013). An extended chaotic maps based user authentication and privacy preserving scheme against DoS attacks in pervasive and ubiquitous computing environments. *Nonlinear Dynamics*, *74*(4), 1133–1143.
35. Lee, C. C., Chen, C. L., Wu, C. Y., & Huang, S. Y. (2012). An extended chaotic maps-based key agreement protocol with user anonymity. *Nonlinear Dynamics*, *69*(1–2), 79–87.
36. He, D. B., Chen, Y. T., & Chen, J. H. (2012). Cryptanalysis and improvement of an extended chaotic maps-based key agreement protocol. *Nonlinear Dynamics*, *69*(3), 1149–1157.
37. Lai, H., Xiao, J., Li, L., et al. (2012). Applying semigroup property of enhanced chebyshev polynomials to anonymous authentication protocol. *Mathematical Problems in Engineering*; vol. 2012, Article ID 454823, 17 pp, 2012. doi:10.1155/2012/454823
38. Zhao, F. J., Gong, P., Li, S., Li, M. G., & Li, P. (2013). Cryptanalysis and improvement of a three-party key agreement protocol using enhanced Chebyshev polynomials. *Nonlinear Dynamics*, *74*(1–2), 419–427.
39. Xie, Q., Zhao, J. M., & Yu, X. Y. (2013). Chaotic maps-based three-party password-authenticated key agreement scheme. *Nonlinear Dynamics*, *74*(4), 1021–1027.
40. Zhang, L. H. (2008). Cryptanalysis of the public key encryption based on multiple chaotic systems. *Chaos, Solitons & Fractals*, *37*(3), 669–674.
41. Kocarev, L., & Lian, S. (2011). *Chaos-based cryptography: Theory, algorithms and applications* (Vol. 354). Berlin: Springer.
42. Tsai, C. S., Lee, C. C., & Hwang, M. S. (2006). Password authentication schemes: Current status and key issues. *International Journal Network Security*, *3*(2), 101–115.
43. Burrows, M., Abadi, M., & Needham, R. M. (1871). A logic of authentication. *Proceedings of the Royal Society of London A—Mathematical and Physical Sciences*, *1989*(426), 233–271.
44. Ballare, M., & Rogaway, P. (1993). Random oracles are practical: A paradigm for designing efficient protocols. In*Proceedings of the 1st ACM conference on computer and communications security (CCS'93)*, (pp. 62–73).
45. Shoup, V. (2004). Sequences of games: A tool for taming complexity in security proofs. Cryptology ePrint Archieve, Report 2004/332. http://eprint.iacr.org/2004/332.
46. Xu, J., Zhu, W. T., & Feng, D. G. (2009). An improved smart card based password authentication scheme with provable security. *Computer Standards and Interfaces*, *31*(4), 723–728.
47. Dolev, D., & Yao, A. (1983). On the security of public key protocols. *IEEE Transactions on Information Theory*, *29*(2), 198–208.
48. Kocher, P., Jaffe, J., & Jun, B. (1999). Differential power analysis. In *Proceedings of advances in cryptology (Crypto'99)* (pp. 388–397).
49. Messerges, T. S., Dabbish, E. A., & Sloan, R. H. (2002). Examining smart-card security under the threat of power analysis attacks. *IEEE Transactions on Computers*, *51*(5), 541–552.
50. Joye, M., & Olivier, F. (2005). *Side-channel analysis, encyclopedia of cryptography and security*. Amsterdam: Kluwer.

51. Zhou, T., & Xu, J. (2011). Provable secure authentication protocol with anonymity for roaming service in global mobility networks. *Computer Networks*, *55*, 205–213.
52. Xue, K., & Hong, P. (2012). Security improvement on an anonymous key agreement protocol based on chaotic maps. *Communications in Nonlinear Science and Numerical Simulation*, *17*, 2969–2977.

Xiong Li received his master's degree in mathematics and cryptography from Shaanxi Normal University (SNNU) in 2009 and Ph.D. degree in computer science and technology from Beijing University of Posts and Telecommunications (BUPT) in 2012. Dr. Li now is a lecturer of Hunan University of Science and Technology (HNUST). He has published more than 20 referred journal papers. His research interests include cryptography and information security, etc.

Jianwei Niu received his Ph.D. degree in 2002 in computer science from Beijing University of Aeronautics and Astronautics (BUAA). He is a professor in the School of Computer Science and Engineering, BUAA. He has published more than 80 referred papers on such as IEEE INFOCOM, ACM Multimedia, ACM CHI, Sensys, TECS, TII, JPDC, and etc., and filed more than 30 patents in mobile and pervasive computing. He served as the Program Co-Chair of IEEE SEC 2008, Vice Chiar of CPScom 2013, TPC members of InfoCom, Percom, ICC, WCNC, Globecom, LCN, and etc. He has served as editor of International Journal of Ad Hoc and Ubiquitous Computing, Journal of Internet Technology and Journal of Network and Computer Applications (Elsevier). His research work is sponsored by NSFC, National 863 Plan of China, Nokia and other funds. He received New Century Excellent Researcher Award from Ministry of Education of China 2009, the first prize of technical invention of the Ministry of Education of China 2012, and won the best paper award in IEEE ICC 2013, IEEE WCNC 2013, IEEE ICACT 2013, CWSN 2012 and IEEE GreenCom 2010. His current research interests include mobile and pervasive computing, mobile video analysis.

Saru Kumari is currently an Assistant Professor with the Department of Mathematics, Ch. Charan Singh University, Uttar Pradesh, India. She was awarded her Ph.D. degree in Mathematics in 2012 from CCS University, Meerut, Uttar Pradesh, India. Her current research interests include information security, digital authentication, security of wireless sensor networks, and applied mathematics.

SK Hafizul Islam has received his Ph.D in Computer Science and Engineering from Indian School of Mines (ISM), Dhanbad, India, under the INSPIRE Fellowship Ph.D Program, funded by DST, Govt. of India and Information Security Education and Awareness (ISEA) program, funded by Department of Information Technology, Ministry of Communication and Information Technology, Govt. of India. He did M.Tech in Computer Application from ISM Dhanbad in 2009. He received his B.Sc. (Hons.) in Mathematics and M.Sc. in Applied Mathematics from Vidyasagar University, West Bengal, India in 2004 and 2006, respectively. Presently, he has been working as an Assistant Professor in the Department of Computer Science and Information Systems, BITS Pilani, Rajasthan 333031, India. He served as reviewer in many reputed International Journals and Conferences. His research interest includes Cryptography and Information Security.

Fan Wu received the Bachelor of Engineering degree in Computer Science from Shandong University, Jinan, China, in 2003 and received Master of Engineering degree in Computer Software and Theory from Xiamen University, Xiamen, China, in 2008. Now, he is a lecturer in Xiamen Institute of Technology. His current research interests include information security, internet protocols, and network management.

Muhammad Khurram Khan is currently an associate professor at Center of Excellence in Information Assurance (CoEIA), King Saud University, Kingdom of Saudi Arabia. He is the Editor-in-Chief of Telecommunication Systems (Springer). He is the full-time Editor/Associate Editor of several international journals, including Journal of Network and Computer Applications (Elsevier), IEEE Access Journal, Security and Communication Networks (Wiley), Journal of Medical Systems (Springer), PLOS ONE (USA), Computers and Electrical Engineering (Elsevier), Electronic Commerce Research (Springer), Scientific World Journal (Hindawi), etc. He has also played role of the guest editor of several international ISI-indexed journals. Moreover, he is one of the organizing chairs of more than 5 dozen international conferences and member of technical committees of more than 6 dozen international conferences. He has published around 200 research papers in the journals and conferences of international repute. In addition, he is an inventor of 7 US/PCT patents in different areas of Cybersecurity. He has edited 7 books/proceedings published by Springer-Verlag and IEEE. His research areas of interest are Cybersecurity, digital authentication, biometrics, multimedia security, and technological innovation management. He is a senior member of the IEEE (USA) and a member of the IEEE Technical Committee on Security and Privacy.

Ashok Kumar Das is currently working as an Assistant Professor in the Center for Security, Theory and Algorithmic Research of the International Institute of Information Technology (IIIT), Hyderabad 500 032, India. He received his Ph.D. degree in Computer Science and Engineering, M.Tech. degree in Computer Science and Data Processing, and M.Sc. degree in Mathematics, all from the Indian Institute of Technology, Kharagpur, India. He received the Institute Silver Medal from the Indian Institute of Technology, Kharagpur, India. His current research interests include cryptography, wireless sensor network security, hierarchical access control, data mining, security in vehicular ad hoc networks, smart grid and cloud computing, and remote user authentication. He has published more than 95 papers in international journals and conferences in these areas. For more details, please visit https://sites.google.com/site/iitkgpakdas/, http://www.iiit.ac.in/people/faculty/ashokkdas.