Hindawi
Security and Communication Networks
Volume 2018, Article ID 5432960, 13 pages
https://doi.org/10.1155/2018/5432960

Research Article

A Novel Multiserver Authentication Protocol with Multifactors for Cloud Service

Jian Song,[1] **Guang-song Li**,[2] **Bo-ru Xu**,[3] **and Chuan-gui Ma**[4]

[1]*Force 61660 of P.L.A., Beijing 100089, China*
[2]*State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China*
[3]*University of Space Engineering, Beijing 101400, China*
[4]*Aviation Institute, Beijing 101116, China*

Correspondence should be addressed to Guang-song Li; lgsok@163.com

Received 23 February 2018; Accepted 9 August 2018; Published 21 November 2018

Academic Editor: Pino Caballero-Gil

Secure and efficient authentication protocols are necessary for cloud service. Multifactor authentication protocols taking advantage of smart card, user's password, and biometric, are more secure than password-based single-factor authentication protocols which are widely used in practice. However, most of the multiserver authentication protocols may have weak points, such as smart card loss attack, man-in-the-middle attack, anonymity, and high computation cost of authentication center. In order to overcome the above weaknesses, we propose a novel multiserver multifactor authentication protocol based on the Kerberos protocol using the extended Chebyshev chaotic mapping as a cryptographic algorithm. The proposed protocol achieves anonymity without sharing secret keys in advance and needs the user to register with the authentication center only once. Finally, we prove the security of the new protocol with BAN logic and compare it with other multifactor authentication protocols for multiserver environment. The results show that our proposed protocol is more secure and efficient and better for practical application.

1. Introduction

With the rapid developing of cloud computing [1, 2], now a variety of cloud servers have stored massive user sensitive data. When users want to access the data, they need to log on to the server through the public channel. What is more, users may have a variety of service requirements and may need to access multiple application servers in a short time. Figure 1 depicts a typical scenario for cloud service. However, in this process, an adversary could intercept, tamper, and forge the information between the user and the server through some technical means. When users access some privacy services, they do not even want other people to know their identity. In order to provide secure and efficient services for a valid user, authentication protocols were proposed [3].

In practice, there are three basic methods to verify the identity of users: (1) what the user knows, such as user password; (2) what the user has, such as smart card; (3) the user's unique biological information, such as fingerprint and iris. As single-factor authentication protocols are based on password which are easy to operate, scalable, and cheap, most people prefer to use this authentication scheme. Therefore, the most commonly used authentication scheme in the current network is still single-factor authentication protocols based on password [4]. However the single-factor authentication protocol has the following inherent defects: (1) the limitation of human memory capacity leads to low entropy of password selection; (2) the development of password cracking hardware and algorithm makes the efficiency of offline dictionary attack greatly improved. Moreover in a single-factor authentication protocol the server needs to store the user's identity and the corresponding password information, even if the password information is hashed; once the server data is stolen, the user will face serious security threat [5].

To solve the problem, Chang et al. [6] firstly introduced the smart card as another factor besides password into authentication schemes, which contributes to the two-factor authentication scheme. In such scheme, the users are required to know not only the correct password but also the

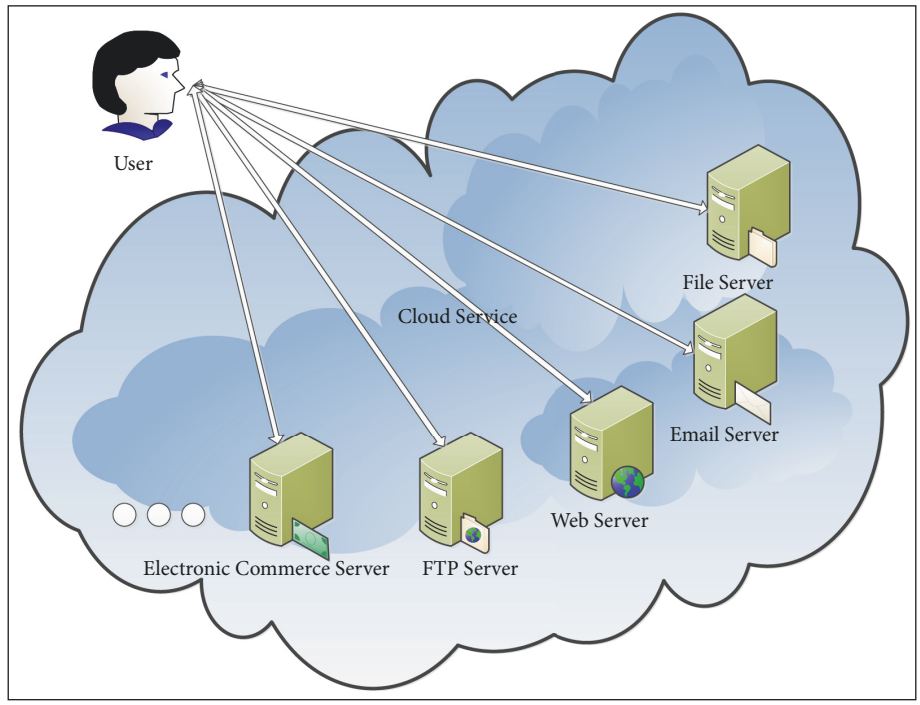

Figure 1: A typical scenario for cloud service.

corresponding smart card; then he/she can access to resource by interacting with the server. However, passwords might be forgotten, and smart cards might be lost or stolen. In contrast, biometric methods, such as fingerprints or iris scans, have no such drawbacks. Therefore, these years, many multifactor authentication protocols using biometric characteristic as an additional factor were proposed [7–13]. Many of these protocols are only for the single server environment. That is, when users want to access multiple servers, they have to register many times and maintain a lot of username/password pairs with the corresponding smart card, which is inefficient if each login should be unique for each server or insecure if the same login is used for multiple servers.

In 2016, Amin et al. proposed a new multifactor authentication scheme for multiserver environment and claimed that it was secure for all known attacks [14]. However, in 2017, Jiang et al. found that Amin et al.'s scheme has the following security issues. (1) If the smart card was stolen, the attacker could recover the user's ID and password. (2) If the temporary parameters of either of the two parties were leaked, the attacker could obtain the session key [15]. Then, Jiang et al. improved Amin et al.'s scheme with Rabin cryptosystem, fuzzy validation, and timestamp and verified the security of the improved scheme with ProVerif. In article [15], Jiang et al. also pointed out that Wu et al.'s scheme [16] is vulnerable to smart card loss attacks. Nevertheless, we found that, in Jiang et al.'s scheme [15], the user's identity is hidden in a message only with the timestamp as variable. If the user's timestamp is the same as the adversary's, then the adversary could obtain the user's real identity by simple XOR operations. So the scheme does not achieve anonymity. Recently, several multifactor authentication schemes have been proposed to the study of authentication and key agreement in the multiserver environment [16–19]. However, most of these schemes' computational cost is high due to the modulus exponentiation operation, the point addition operation of elliptic curve, and so on. Thus, those schemes may not be suitable for some cloud scenarios, in which the user may access multiple servers in a short time, the user terminal only has limited computing power, the server needs to handle a large number of requests at the same time, and so on.

Though multifactor authentication protocols are widely studied by many scholars, few of them are specifically for cloud service. We have taken into account the needs of cloud services and applied new technologies to design multifactor authentication for the above environment. In order to design more efficient and secure authentication protocols, the extended Chebyshev chaotic mapping [20–22] is introduced in this paper. The computational cost of extended Chebyshev polynomials is lower, compared to the traditional modular exponentiation operation and the point addition operation of elliptic curve [20–24]. Moreover, with the idea of Kerberos protocol, we propose a novel multifactor authentication protocol for the multiserver environment. In our scheme, the frequency of user accessing the authentication center is reduced, which greatly relieves the burden of the authentication center. In addition, the new protocol accomplishes security and usability features necessary for all the participants, while maintaining high efficiency.

The remainder of this paper is organized as follows. The preliminaries of enhanced Chebyshev chaotic maps and fuzzy extraction are given in Section 2. In Section 3, we propose a novel multifactor authentication protocol for multiserver environment. Section 4 and Section 5 present security and

efficiency analyses of the new protocol. Section 6 concludes the paper.

2. Preliminaries

2.1. Enhanced Chebyshev Chaotic Maps [20–22]. The enhanced Chebyshev polynomial $T_n(x)$ is a polynomial in x of degree n and is defined by the following relation:

$$T_n(x) \equiv \begin{cases} 1, & n = 0 \\ x \bmod p, & n = 1 \\ \left(2x \cdot T_{n-1}(x) - T_{n-2}(x)\right) \bmod p, & n \geq 2 \end{cases} \quad (1)$$

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

The enhanced Chebyshev polynomial satisfies the semigroup property and satisfies

$$T_r\left(T_s(x)\right) = T_{rs}(x) = T_s\left(T_r(x)\right) \quad (2)$$

for $s, r \in \mathbb{Z}^+$.

2.2. Difficulty Assumptions. Enhanced Chebyshev polynomials are associated with three hard problems, which are the extended chaotic-map-based discrete logarithm problem (DLP), the computational Diffie–Hellman problem (CDHP), and the decisional Diffie–Hellman problem (DDHP), described as follows.

(1) Extended Chaotic-Map-Based DLP: given x,y,$T(\cdot)$ and p, where p is a large prime number, finding the integer r satisfying

$$y = T_r(x) \bmod p \quad (3)$$

is computationally infeasible.

(2) Extended Chaotic-Map-Based CDHP: given $T_r(x)$, $T_s(x)$, x, $T(\cdot)$, and p, where $r, s \geq 2$, $x \in (-\infty, +\infty)$ and p is a large prime number, calculating

$$T_{rs}(x) \equiv T_r\left(T_s(x)\right) \equiv T_s\left(T_r(x)\right) \bmod p \quad (4)$$

is computationally infeasible.

(3) Extended Chaotic-Map-Based DDHP: given $T_r(x)$, $T_s(x)$, $T_z(x)$, x, $T(\cdot)$, and p, p is a large prime number, deciding whether

$$T_{rs}(x) \equiv T_z(x) \bmod p \quad (5)$$

holds or is not computationally infeasible.

2.3. Fuzzy Extractor. Traditional hash functions return different outputs if their inputs are not completely the same. Thus we need some other technology to extract biometrics. According to [25], the biometrics of all persons can be retrieved as nearly uniform random bit strings R by an auxiliary string P from biometric input B with a fuzzy extractor. The extractor can recover R with the auxiliary string P even if the biometric input is B', as long as it is very close to the original B. Thus, R can be utilized as a key stand for biometrics in a security application. Fuzzy extractor consists of two procedures (*Gen, Rep*).

(1) $Gen(B) \longrightarrow \{R, P\}$

Gen is a probabilistic algorithm, which takes a biometric input B as input and outputs a random string R with length l and a public string P.

(2) $Rep(B', P) \longrightarrow R$

Rep is a deterministic reproduction procedure which is able to recover R from a slightly different biometric B' and the auxiliary parameter P. That is, $Rep(B', P) = R$ for all B' satisfying $dis(B, B') \leq \varepsilon$, where ε is an error-tolerance.

Table 1: Notations.

Notation	Description
ID_X	The identity of the entity X
sk_X	The secret key of the entity X
AC	Authentication center
SC	Smart card
s_a/s_b	The master key of authentication center
$k_0/T_{k_0}(x)$	The private/public key of authentication center
$k_j/T_{k_j}(x)$	The private/public key of sever S_j
PW_i	The password of user U_i
FP_i	The fingerprint information of user U_i
$TimeSetup$	The generation time of Y_{ij}
l	The security length parameter
$h(\cdot)$	Cryptographic one-way hash function, satisfying $h : \{0,1\}^* \longrightarrow \{0,1\}^l$
$\|$	String concatenation
$\oplus$	Exclusive-or operation

2.4. Adversary's Capability. In this paper, we assume the following about a probabilistic, polynomial-time adversary to properly capture the security requirements of a multifactor biometric authentication scheme that uses smart cards during the registration phase, authentication phase, and password change phase.

(1) The adversary is able to have complete control over all message exchanges between the protocol participants. That is, the adversary can intercept, insert, modify, delete, and eavesdrop on messages exchanged among the two parties at will.

(2) The adversary can extract sensitive information from the smart card of a user through a power analysis attack.

2.5. Notations. Table 1 lists the notation that is used throughout this paper.

3. Our Proposed Authentication Protocol

For cloud service, we proposed a multifactor authentication protocol in which there are three kinds of entities: the user, the server, and the authentication center (the trusted third party), as described in Figure 2.

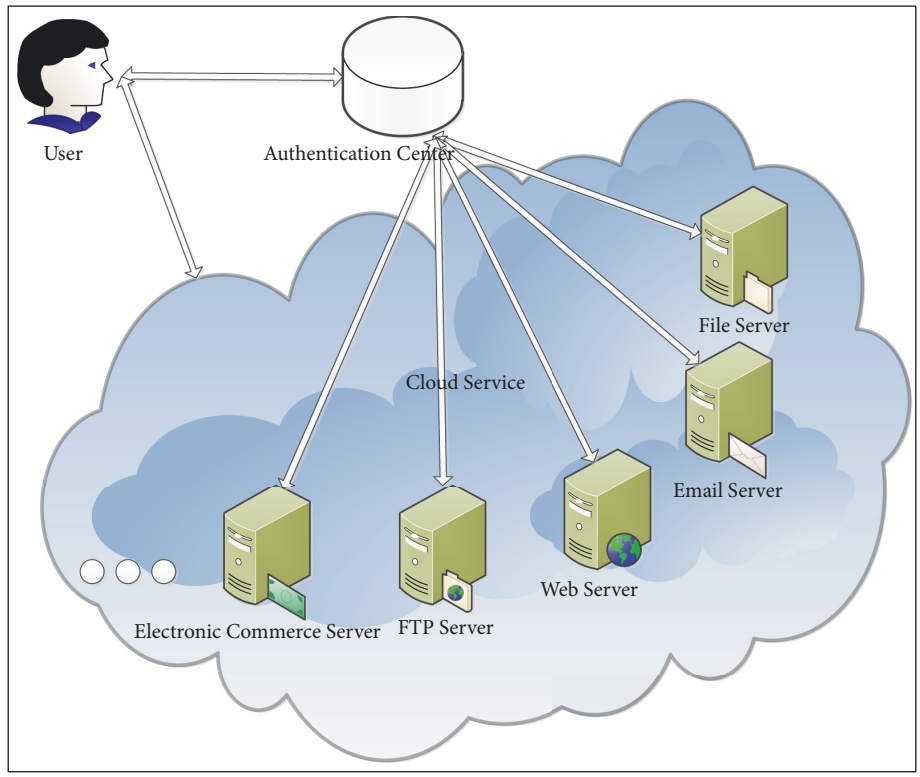

Figure 2: Network model of our protocol.

The characteristics of each participant are different. In our scheme, the actual needs of all the participating entities are considered under the guarantee of security. (1) For user, the user's anonymity is first achieved. Secondly, in multiserver environment users can access all servers only by registering one time. At the same time, considering the limitation of the user's computing power, the user's computational cost is low in our scheme. In addition, the user can change his/her password offline. (2) For the authentication center, taking into account the fact that authentication center needs to participate in each user's access in existing authentication protocols for multiserver environment, our scheme designs a ticket. When the ticket is not expired, there is no need for authentication center to participate in the authentication process, which greatly reduces the burden of the authentication center. (3) For the server, in our scheme the authentication center and the server do not need to share a key in advance. Moreover, considering the different actual requirements of each application server, the expiry time of the ticket in our scheme is determined by the server.

Our scheme contains four phases, namely, system setup phase (Figure 3), user registration phase (Figure 4), authentication phase (Figure 5), and password change phase (Figure 6).

3.1. System Setup. AC selects and computes the system parameters in offline mode. And sever S_j registers with AC through a secure channel.

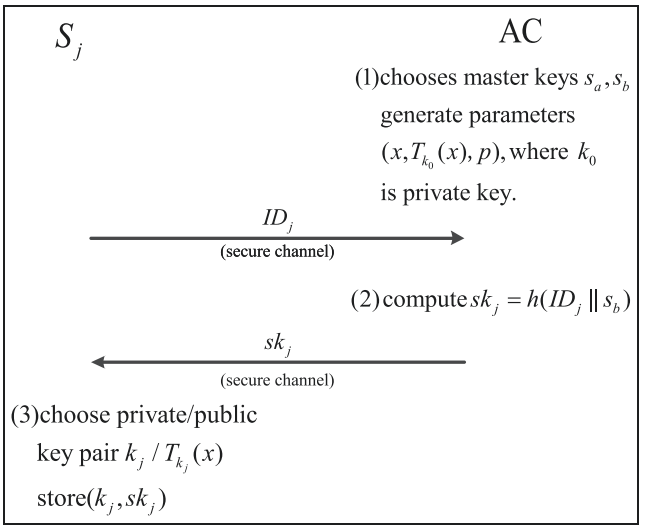

Figure 3: System setup phase.

Step 1. AC chooses master secret keys s_a and s_b. Then AC generates a random number x and a large prime number p and chooses a random number k_0 as private key. Next, AC computes $T_{k_0}(x)$ as public key and makes the parameters $(x, T_{k_0}(x), p)$ known to the public.

Step 2. Sever S_j selects an identity ID_j and sends it to AC through secure channel. AC checks whether ID_j exists in the

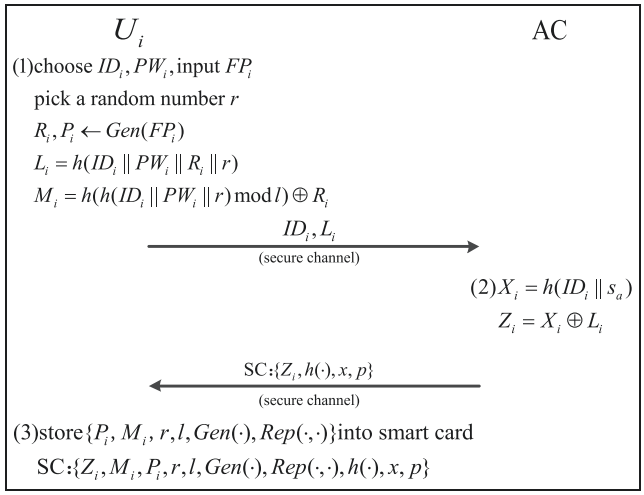

Figure 4: User registration phase.

database. If it does, AC indicates S_j to select a new identity; otherwise, AC compute $sk_j = h(ID_j \| s_b)$ and send it to S_j through a secure channel.

Step 3. Sever S_j chooses a random number k_j as private key and computes $T_{k_j}(x)$ as public key. Finally, S_j stores (sk_j, k_j) in its memory and makes $T_{k_j}(x)$ known to the public.

3.2. User Registration. In this phase, U_i registers with AC through secure channel and gets back a smart card SC.

Step 1. User U_i selects an identity ID_i and a password PW_i; then U_i inputs fingerprint information FP_i through fingerprint extractor. Next, U_i gets a random bit strings R_i and an auxiliary string P_i from biometric input FP_i with algorithm Gen. Then, U_i chooses a high-entropy random number r and computes $L_i = h(ID_i \| PW_i \| R_i \| r)$ and $M_i = h(h(ID_i \| PW_i \| r) \bmod l) \oplus R_i$. Finally, U_i sends ID_i and L_i to AC through a secure channel.

Step 2. After receiving ID_i and L_i from U_i, AC checks whether ID_i exists in the database. If it does, AC indicates U_i to select a new identity; otherwise, AC computes $X_i = h(ID_i \| s_a)$ and $Z_i = X_i \oplus L_i$; then AC stores $\{Z_i, h(\cdot), x, p\}$ into SC. Finally, AC sends SC to U_i through a secure channel.

Step 3. After receiving SC from AC, U_i stores $\{P_i, M_i, r, l, Gen(\cdot), Rep(\cdot,\cdot)\}$ into SC. Finally, SC contains $\{Z_i, M_i, P_i, r, l, Gen(\cdot), Rep(\cdot,\cdot), h(\cdot), x, p\}$.

3.3. Authentication. U_i first logins to the SC; then U_i starts a mutual authentication process with AC to get a ticket Y_{ij} for accessing sever S_j. Next, U_i implements mutual authentication with S_j by Y_{ij} and establishes a session key $k_{session}$ with S_j, where the ticket Y_{ij} has an expiry time which is determined by S_j.

Step 1. U_i attaches the smart card SC and enters the identity ID_i, password PW_i, and fingerprint FP_i. Then, SC computes $R_i' = Req(FP_i', P_i')$ and $R_i = h(h(ID_i' \| PW_i' \| r) \bmod l) \oplus M_i$. The smart card SC rejects U_i's login request if $R_i \neq R_i'$; otherwise, SC chooses a random number u and computes $T_{U_i-\mathrm{SA}} = T_u(T_{k_0}(x))$, $NID_{ij} = (ID_i \| ID_j) \cdot T_{U_i-\mathrm{SA}}$, $L_i = h(ID_i \| PW_i \| R_i \| r)$, $X_i = Z_i \oplus L_i$, and $Aut_1 = h(ID_i \| ID_j \| X_i \| T_{U_i-\mathrm{SA}} \| TimeStamp_1)$. Finally, U_i sends $NID_{ij}, T_u(x), Aut_1, TimeStamp_1$ to AC.

Step 2. After receiving the message $NID_{ij}, T_u(x), Aut_1, TimeStamp_1$ from U_i, AC verifies whether $TimeStamp_1$ is valid. If not, AC rejects U_i's request; otherwise, AC computes $T'_{U_i-\mathrm{SA}} = T_{k_0}(T_u(x))$, $ID_i' \| ID_j = NID_{ij}/T'_{U_i-\mathrm{SA}}$, $X_i' = h(ID_i' \| s_a)$, and $h(ID_i' \| ID_j \| X_i' \| T'_{U_i-\mathrm{SA}} \| TimeStamp_1)$. Then AC terminates the session, if $Aut_1 \neq h(ID_i' \| ID_j \| X_i' \| T'_{U_i-\mathrm{SA}} \| TimeStamp_1)$; otherwise, AC computes a ticket $Y_{ij} = h(ID_i' \| h(ID_j \| s_b) \| TimeSetup)$, $NY_{ij} = (Y_{ij} \| TimeSetup) \oplus h(X_i')$, and $Aut_2 = h(ID_i' \| ID_j \| X_i' \| Y_{ij} \| TimeSetup \| TimeStamp_2)$, where $TimeSetup$ is the generation time of Y_{ij}. Finally, AC sends $NY_{ij}, Aut_2, TimeStamp_2$ to U_i.

Step 3. After receiving the message $Y_{ij}, Aut_2, TimeStamp_2, U_i$ verifies whether $TimeStamp_2$ is valid. If not, U_i terminates the session; otherwise, U_i computes $Y_{ij}' \| TimeSetup' = NY_{ij} \oplus h(X_i)$ and $h(ID_i \| ID_j \| X_i \| Y_{ij}' \| TimeSetup \| TimeStamp_2)$. Then U_i terminates the session, if $Aut_2 \neq h(ID_i \| ID_j \| X_i \| Y_{ij}' \| TimeSetup \| TimeStamp_2)$; otherwise, U_i chooses a random number u_2 and then computes $T_{U_i-S_j} = T_{u_2}(T_{k_j}(x))$, $NID_{ij}* = (ID_i \| ID_j \| TimeSetup) \cdot T_{U_i-S_j}$, and $Aut_3 = h(ID_i \| ID_j \| Y_{ij}' \| T_{U_i-S_j} \| TimeSetup \| TimeStamp_3)$. Finally, U_i sends $NID_{ij}*, T_{u_2}(x), Aut_3, TimeStamp_3$ to S_j.

Step 4. After receiving the message $NID_{ij}*, T_{u_2}(x), Aut_3, TimeStamp_3, S_j$ verifies whether $TimeStamp_3$ is valid. If not, S_j terminates the session; otherwise, S_j computes $T'_{U_i-S_j} = T_{k_j}(T_{u_2}(x))$ and $ID_i' \| ID_j' \| TimeSetup' = NID_{ij} * /T'_{U_i-S_j}$. Then S_j verifies whether $ID_j' = ID_j$ and $TimeSetup'$ is valid. If not, S_j terminates the session; otherwise, S_j computes $Y_{ij}' = h(ID_i' \| sk_j \| TimeSetup')$ and $h(ID_i' \| ID_j \| Y_{ij} \| T'_{U_i-S_j} \| TimeSetup' \| TimeStamp_3)$. Then S_j terminates the session, if $Aut_3 \neq h(ID_i' \| ID_j \| Y_{ij} \| T'_{U_i-S_j} \| TimeSetup' \| TimeStamp_3)$; otherwise, S_j chooses a random number v and then computes $k_{session} = T_v(T_{u_2}(x))$ and $Aut_4 = h(ID_i' \| ID_j \| Y_{ij}' \| k_{session} \| TimeStamp_4)$. Finally, S_j sends $T_v(x), Aut_4, TimeStamp_4$ to U_i.

Step 5. After receiving the message $T_v(x), Aut_4, TimeStamp_4$, U_i verifies whether $TimeStamp_4$ is valid. If not, U_i terminates the session; otherwise, U_i computes $k_{session} = T_{u_2}(T_v(x))$ and $h(ID_i \| ID_j \| Y_{ij} \| k_{session} \| TimeStamp_4)$. Then U_i terminates the session, if $Aut_4 \neq h(ID_i \| ID_j \| Y_{ij} \| k_{session} \| TimeStamp_4)$; otherwise, U_i and S_j complete mutual authentication successfully. At this point,

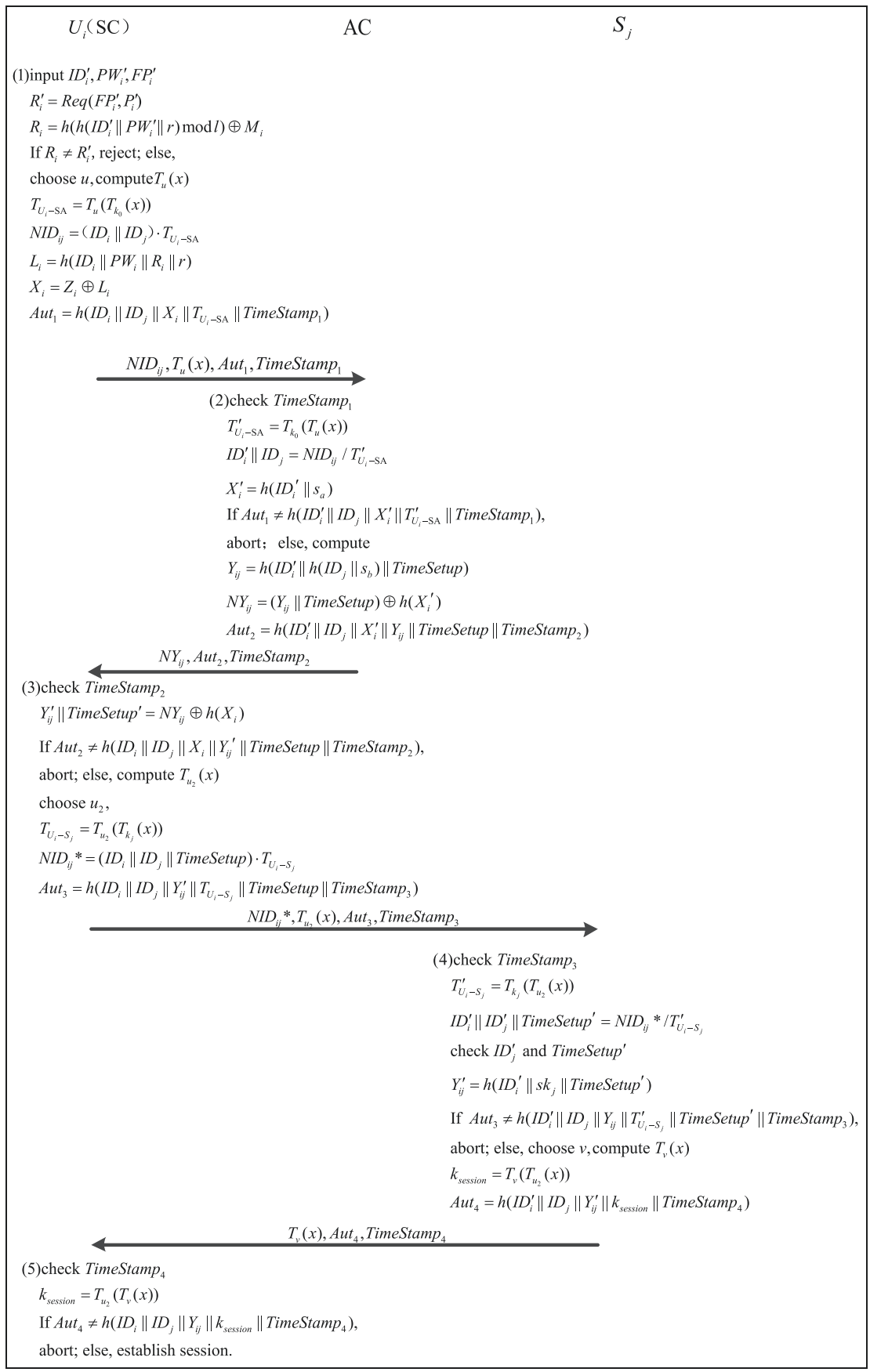

Figure 5: Authentication phase.

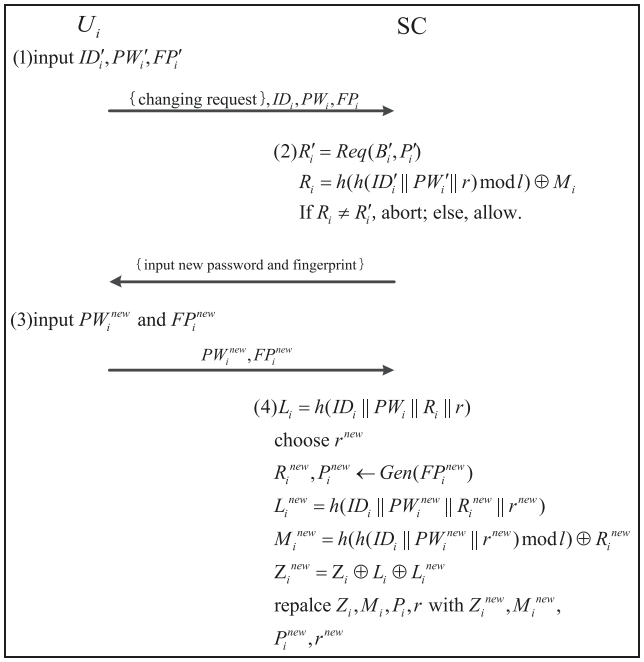

Figure 6: Password change phase.

a session key $k_{session} = T_{u_2 \cdot v}(x) = T_{u_2}(T_v(x)) = T_v(T_{u_2}(x))$ has been established among U_i and S_j.

After all these steps are completed, if U_i wants to access S_j again, it can be executed directly from Step 3 without AC's participating, where the ticket Y_{ij} must be not out of date.

3.4. Password Change. In this phase, U_i only needs to log into SC successfully and then inputs new password PW_i^{new} and fingerprint information FP_i^{new}, without involvement of AC and S_j.

Step 1. U_i inserts the smart card into a card reader and enters the identity ID_i, password PW_i, and fingerprint FP_i.

Step 2. SC computes $R_i' = Req(FP_i', P_i')$ and $R_i = h(h(ID_i' \| PW_i' \| r) \bmod l) \oplus M_i$. The smart card SC rejects U_i's login request if $R_i \neq R_i'$; otherwise, SC indicates U_i to input new password and fingerprint information.

Step 3. U_i inputs new password PW_i^{new} and fingerprint information FP_i^{new}.

Step 4. SC computes $L_i = h(ID_i \| PW_i \| R_i \| r)$, then chooses a random number r^{new}, and computes R_i^{new}, $P_i^{new} \longleftarrow Gen(FP_i^{new})$, $L_i^{new} = h(ID_i \| PW_i^{new} \| R_i^{new} \| r^{new})$, $M_i^{new} = h(h(ID_i \| PW_i^{new} \| r^{new}) \bmod l) \oplus R_i^{new}$, and $Z_i^{new} = Z_i \oplus L_i \oplus L_i^{new}$. Finally, SC updates Z_i, M_i, P_i, r with $Z_i^{new}, M_i^{new}, P_i^{new}, r^{new}$.

4. Security Analysis

In this section, we first use the BAN logic [26] to prove that a ticket will be agreed between the user and the authentication center; moreover, a session key will be agreed between the user and the sever after performing our new protocol. Then we demonstrate that the proposed protocol can withstand various known attacks and satisfy security requirements in cloud service.

Table 2: Notations in BAN logic.

Notation	Description
P, Q	entity
X, Y	statements
K	key
$\{X, Y\}$	X or Y is one part of $\{X, Y\}$
$P \mid\equiv X$	P believes X
$P \lhd X$	P sees X
$P \mid\sim X$	P once said X
$P \mid\Longrightarrow X$	P has jurisdiction over X
$\#(X)$	X is fresh
$P \overset{K}{\longleftrightarrow} Q$	P and Q share a key K
$P \overset{X}{\rightleftharpoons} Q$	P and Q share a secret X
$\{X\}_K$	X is encrypted with the key K
$\langle X \rangle_Y$	X combined with Y

4.1. Notations and Logic Rules. Table 2 lists the notations used in the BAN logic.

There are 19 logical rules in BAN logic. The nth logical rule denotes $\mathrm{R}_n (n = 1, 2, \cdots, 19)$. Some main logical rules of the BAN logic, which will be used in our analysis, are described as follows, where Γ / A means conclusion A can be deduced by precondition sets Γ.

The message-meaning rule is

$$\mathrm{R}_1: \frac{P \mid\equiv P \overset{K}{\longleftrightarrow} Q, P \lhd \{X\}_K}{P \mid\equiv Q \mid\sim X}$$
$$\mathrm{R}_3: \frac{P \mid\equiv P \overset{Y}{\rightleftharpoons} Q, P \lhd \{X\}_Y}{P \mid\equiv Q \mid\sim X} \quad (6)$$

The nonce-verification rule is

$$\mathrm{R}_4: \frac{P \mid\equiv \#(X), P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\equiv X} \quad (7)$$

The jurisdiction rule is

$$\mathrm{R}_5: \frac{P \mid\equiv Q \mid\Longrightarrow X, P \mid\equiv Q \mid\equiv X}{P \mid\equiv X} \quad (8)$$

The seeing rule is

$$\mathrm{R}_8: \frac{P \mid\equiv P \overset{K}{\longleftrightarrow} Q, P \lhd \{X\}_K}{P \lhd X} \quad (9)$$

The freshness rule is

$$\mathrm{R}_{11}: \frac{P \mid\equiv \#(X)}{P \mid\equiv \#(X, Y)} \quad (10)$$

The belief rule is

$$\mathrm{R}_{13}\colon \frac{P \mid\equiv (X,Y)}{P \mid\equiv (X)}$$
$$\mathrm{R}_{14}\colon \frac{P \mid\equiv Q \mid\equiv (X,Y)}{P \mid\equiv Q \mid\equiv (X)} \tag{11}$$

4.2. Formal Proof. First, our proposed protocol is transformed to the idealized form.

Message 1:

$$U_i \longrightarrow \mathrm{AC}: \left\{U_i \overset{ID_i}{\rightleftharpoons} \mathrm{AC}, U_i \overset{ID_j}{\rightleftharpoons} \mathrm{AC}, T_u(x), TimeStamp_1\right\}_{X_i} \tag{12}$$

Message 2:

$$\mathrm{AC} \longrightarrow U_i: \left\{U_i \overset{ID_i}{\rightleftharpoons} \mathrm{AC}, U_i \overset{ID_j}{\rightleftharpoons} \mathrm{AC}, U_i \overset{Y_{ij}}{\rightleftharpoons} \mathrm{AC}, U_i \overset{TimeSetup}{\rightleftharpoons} \mathrm{AC}, TimeStamp_2\right\}_{X_i} \tag{13}$$

Message 3:

$$U_i \longrightarrow S_j: \left\{U_i \overset{ID_i}{\rightleftharpoons} S_j, U_i \overset{ID_j}{\rightleftharpoons} S_j, U_i \overset{TimeSetup}{\rightleftharpoons} S_j, T_{u_2}(x), TimeStamp_3\right\}_{Y_{ij}} \tag{14}$$

Message 4:

$$U_i \longrightarrow S_j: \left\{U_i \overset{ID_i}{\rightleftharpoons} S_j, U_i \overset{ID_j}{\rightleftharpoons} S_j, T_v(x), TimeStamp_4\right\}_{Y_{ij}} \tag{15}$$

We need to prove that our proposed protocol could achieve the following goals.

Goal 1:

$$U_i \mid\equiv U_i \overset{ID_i}{\rightleftharpoons} \mathrm{AC} \tag{16}$$

Goal 2:

$$\mathrm{SA} \mid\equiv U_i \overset{ID_i}{\rightleftharpoons} \mathrm{AC} \tag{17}$$

Goal 3:

$$U_i \mid\equiv U_i \overset{ID_j}{\rightleftharpoons} \mathrm{AC} \tag{18}$$

Goal 4:

$$\mathrm{SA} \mid\equiv U_i \overset{ID_j}{\rightleftharpoons} \mathrm{AC} \tag{19}$$

Goal 5:

$$U_i \mid\equiv U_i \overset{Y_{ij}}{\rightleftharpoons} S_j \tag{20}$$

Goal 6:

$$S_j \mid\equiv U_i \overset{Y_{ij}}{\rightleftharpoons} S_j \tag{21}$$

Goal 7:

$$U_i \mid\equiv U_i \overset{ID_i}{\rightleftharpoons} S_j \tag{22}$$

Goal 8:

$$S_j \mid\equiv U_i \overset{ID_i}{\rightleftharpoons} S_j \tag{23}$$

Goal 9:

$$U_i \mid\equiv U_i \overset{k_{session}}{\longleftrightarrow} S_j \tag{24}$$

Goal 10:

$$S_j \mid\equiv U_i \overset{k_{session}}{\longleftrightarrow} S_j \tag{25}$$

Then, the following assumptions are made about the initial status of our proposed protocol.

$$\begin{aligned}
&A_1\colon U_i \mid\equiv \#TimeStamp_2\\
&A_2\colon U_i \mid\equiv \#T_{u_2}(x)\\
&A_3\colon U_i \mid\equiv \#TimeStamp_4\\
&A_4\colon U_i \mid\equiv U_i \overset{X_i}{\rightleftharpoons} \mathrm{AC}\\
&A_5\colon U_i \mid\equiv \mathrm{AC} \mid\Longrightarrow U_i \overset{TimeSetup}{\rightleftharpoons} \mathrm{AC}\\
&A_6\colon U_i \mid\equiv \mathrm{AC} \mid\Longrightarrow U_i \overset{ID_i}{\rightleftharpoons} \mathrm{AC}\\
&A_7\colon U_i \mid\equiv \mathrm{AC} \mid\Longrightarrow U_i \overset{ID_j}{\rightleftharpoons} \mathrm{AC}\\
&A_8\colon U_i \mid\equiv S_j \mid\Longrightarrow U_i \overset{ID_i}{\rightleftharpoons} S_j\\
&A_9\colon U_i \mid\equiv S_j \mid\Longrightarrow U_i \overset{k_{session}}{\longleftrightarrow} S_j\\
&A_{10}\colon \mathrm{AC} \mid\equiv \#TimeStamp_1\\
&A_{11}\colon \mathrm{AC} \mid\equiv U_i \overset{X_i}{\rightleftharpoons} \mathrm{AC}\\
&A_{12}\colon \mathrm{AC} \mid\equiv S_j \overset{sk_j}{\rightleftharpoons} \mathrm{AC}\\
&A_{13}\colon \mathrm{AC} \mid\equiv U_i \mid\Longrightarrow U_i \overset{ID_i}{\rightleftharpoons} \mathrm{AC}\\
&A_{14}\colon \mathrm{AC} \mid\equiv U_i \mid\Longrightarrow U_i \overset{ID_j}{\rightleftharpoons} \mathrm{AC}\\
&A_{15}\colon S_j \mid\equiv \#TimeStamp_3\\
&A_{16}\colon S_j \mid\equiv S_j \overset{sk_j}{\rightleftharpoons} \mathrm{AC}\\
&A_{17}\colon S_j \mid\equiv U_i \mid\Longrightarrow U_i \overset{ID_i}{\rightleftharpoons} S_j\\
&A_{18}\colon S_j \mid\equiv U_i \mid\Longrightarrow U_i \overset{k_{session}}{\longleftrightarrow} S_j
\end{aligned} \tag{26}$$

The detailed steps are presented as follows.

(1) From message 1, it is easy to have the following statement:

$$S_1 : \mathrm{AC} \lhd \left\{ U_i \overset{ID_i}{\rightleftharpoons} \mathrm{AC}, U_i \overset{ID_j}{\rightleftharpoons} \mathrm{AC}, T_u(x), TimeStamp_1 \right\}_{X_i} \tag{27}$$

(2) By S_1, A_{11}, and R_3, it is easy to obtain

$$S_2 : \mathrm{AC} \mid\equiv U_i \mid\sim \left(U_i \overset{ID_i}{\rightleftharpoons} \mathrm{AC}, U_i \overset{ID_j}{\rightleftharpoons} \mathrm{AC}, T_u(x), TimeStamp_1 \right) \tag{28}$$

(3) By A_{10} and R_{11}, it is easy to obtain

$$S_3 : \mathrm{AC} \mid\equiv \# \left(U_i \overset{ID_i}{\rightleftharpoons} \mathrm{AC}, U_i \overset{ID_j}{\rightleftharpoons} \mathrm{AC}, T_u(x), TimeStamp_1 \right) \tag{29}$$

(4) By S_2, S_3, and R_4, it is easy to obtain

$$S_4 : \mathrm{AC} \mid\equiv U_i \mid\equiv \left(U_i \overset{ID_i}{\rightleftharpoons} \mathrm{AC}, U_i \overset{ID_j}{\rightleftharpoons} \mathrm{AC}, T_u(x), TimeStamp_1 \right) \tag{30}$$

(5) By $S_4$and R_{14}, it is easy to obtain

$$\begin{gathered} S_5 : \mathrm{AC} \mid\equiv U_i \mid\equiv U_i \overset{ID_i}{\rightleftharpoons} \mathrm{AC} \\ S_6 : \mathrm{AC} \mid\equiv U_i \mid\equiv U_i \overset{ID_j}{\rightleftharpoons} \mathrm{AC} \\ S_7 : \mathrm{AC} \mid\equiv U_i \mid\equiv T_u(x) \end{gathered} \tag{31}$$

(6) By S_5, S_6, A_{13}, A_{14}, and R_5, it is easy to obtain

$$\begin{gathered} S_8 : \mathrm{AC} \mid\equiv U_i \overset{ID_i}{\rightleftharpoons} \mathrm{AC} \ \text{(Goal 2)} \\ S_9 : \mathrm{AC} \mid\equiv U_i \overset{ID_j}{\rightleftharpoons} \mathrm{AC} \ \text{(Goal 4)} \end{gathered} \tag{32}$$

(7) From message 2, it is easy to have the following statement:

$$S_{10} : U_i \lhd \left\{ U_i \overset{ID_i}{\rightleftharpoons} \mathrm{AC}, U_i \overset{ID_j}{\rightleftharpoons} \mathrm{AC}, U_i \overset{Y_{ij}}{\rightleftharpoons} \mathrm{AC}, U_i \overset{TimeSetup}{\rightleftharpoons} \mathrm{AC}, TimeStamp_2 \right\}_{X_i} \tag{33}$$

(8) By S_{10}, A_4, and R_3, it is easy to obtain

$$S_{11} : U_i \mid\equiv \mathrm{AC} \mid\sim \left(U_i \overset{ID_i}{\rightleftharpoons} \mathrm{AC}, U_i \overset{ID_j}{\rightleftharpoons} \mathrm{AC}, U_i \overset{Y_{ij}}{\rightleftharpoons} \mathrm{AC}, U_i \overset{TimeSetup}{\rightleftharpoons} \mathrm{AC}, TimeStamp_2 \right) \tag{34}$$

(9) By A_1 and R_{11}, it is easy to obtain

$$S_{12} : U_i \mid\equiv \# \left(U_i \overset{ID_i}{\rightleftharpoons} \mathrm{AC}, U_i \overset{ID_j}{\rightleftharpoons} \mathrm{AC}, U_i \overset{Y_{ij}}{\rightleftharpoons} \mathrm{AC}, U_i \overset{TimeSetup}{\rightleftharpoons} \mathrm{AC}, TimeStamp_2 \right) \tag{35}$$

(10) By S_{11}, S_{12}, and R_4, it is easy to obtain

$$S_{13} : U_i \mid\equiv \mathrm{AC} \mid\equiv \left(U_i \overset{ID_i}{\rightleftharpoons} \mathrm{AC}, U_i \overset{ID_j}{\rightleftharpoons} \mathrm{AC}, U_i \overset{Y_{ij}}{\rightleftharpoons} \mathrm{AC}, U_i \overset{TimeSetup}{\rightleftharpoons} \mathrm{AC}, TimeStamp_2 \right) \tag{36}$$

(11) By S_{13} and R_{14}, it is easy to obtain

$$\begin{gathered} S_{14} : U_i \mid\equiv \mathrm{AC} \mid\equiv U_i \overset{ID_i}{\rightleftharpoons} \mathrm{AC} \\ S_{15} : U_i \mid\equiv \mathrm{AC} \mid\equiv U_i \overset{ID_j}{\rightleftharpoons} \mathrm{AC} \\ S_{16} : U_i \mid\equiv \mathrm{AC} \mid\equiv U_i \overset{Y_{ij}}{\rightleftharpoons} \mathrm{AC} \\ S_{17} : U_i \mid\equiv \mathrm{AC} \mid\equiv U_i \overset{TimeSetup}{\rightleftharpoons} \mathrm{AC} \end{gathered} \tag{37}$$

(12) By S_{14}, S_{15}, S_{17}, A_5, A_6, A_7, and R_5, it is easy to obtain

$$\begin{gathered} S_{18} : U_i \mid\equiv U_i \overset{ID_i}{\rightleftharpoons} \mathrm{AC} \ \text{(Goal 1)} \\ S_{19} : U_i \mid\equiv U_i \overset{ID_j}{\rightleftharpoons} \mathrm{AC} \ \text{(Goal 3)} \\ S_{20} : U_i \mid\equiv U_i \overset{TimeSetup}{\rightleftharpoons} \mathrm{AC} \end{gathered} \tag{38}$$

(13) By $Y_{ij} = h(ID_i \parallel sk_j \parallel TimeSetup)$ and A_5, we can deduce that

$$S_{21} : U_i \mid\equiv \mathrm{AC} \mid\Longrightarrow U_i \overset{Y_{ij}}{\rightleftharpoons} \mathrm{AC} \tag{39}$$

(14) By S_{16}, S_{21}, and R_5, it is easy to obtain

$$S_{22} : U_i \mid\equiv U_i \overset{Y_{ij}}{\rightleftharpoons} \mathrm{AC} \tag{40}$$

(15) By S_{22}, A_{12}, and $Y_{ij} = h(ID_i \parallel sk_j \parallel TimeSetup)$, it is easy to obtain

$$S_{23} : U_i \mid\equiv U_i \overset{Y_{ij}}{\rightleftharpoons} S_j \ \text{(Goal 5)} \tag{41}$$

(16) In message 3, U_i sends ID_i and $TimeSetup$ encrypted by S_j's public key to the sever S_j. As $Y_{ij} = h(ID_i \parallel sk_j \parallel TimeSetup)$ and ID_i and $TimeSetup$ are integrity protected by secure hash function, combining A_{16}, we can deduce that

$$S_{24} : S_j \mid\equiv U_i \overset{Y_{ij}}{\rightleftharpoons} S_j \ \text{(Goal 6)} \tag{42}$$

(17) From message 3, it is easy to have the following statement:

$$S_{25} : S_j \lhd \left\{ U_i \overset{ID_i}{\rightleftharpoons} S_j, U_i \overset{ID_j}{\rightleftharpoons} S_j, U_i \overset{TimeSetup}{\rightleftharpoons} S_j, T_{u_2}(x), TimeStamp_3 \right\}_{Y_{ij}} \tag{43}$$

(18) By S_{24}, S_{25}, and R_5, it is easy to obtain

$$S_{26} : S_j \mid\equiv U_i \mid\sim \left(U_i \overset{ID_i}{\rightleftharpoons} S_j, U_i \overset{ID_j}{\rightleftharpoons} S_j, U_i \overset{TimeSetup}{\rightleftharpoons} S_j, T_{u_2}(x), TimeStamp_3\right) \tag{44}$$

(19) By A_{15} and R_{11}, it is easy to obtain

$$S_{27} : S_j \mid\equiv \#\left(U_i \overset{ID_i}{\rightleftharpoons} S_j, U_i \overset{ID_j}{\rightleftharpoons} S_j, U_i \overset{TimeSetup}{\rightleftharpoons} S_j, T_{u_2}(x), TimeStamp_3\right) \tag{45}$$

(20) By S_{26}, S_{27}, and R_4, it is easy to obtain

$$S_{28} : S_j \mid\equiv U_i \mid\equiv \left(U_i \overset{ID_i}{\rightleftharpoons} S_j, U_i \overset{ID_j}{\rightleftharpoons} S_j, U_i \overset{TimeSetup}{\rightleftharpoons} S_j, T_{u_2}(x), TimeStamp_3\right) \tag{46}$$

(21) By S_{28} and R_{14}, it is easy to obtain

$$\begin{aligned} S_{29} &: S_j \mid\equiv U_i \mid\equiv U_i \overset{ID_i}{\rightleftharpoons} S_j \\ S_{30} &: S_j \mid\equiv U_i \mid\equiv U_i \overset{ID_j}{\rightleftharpoons} S_j \\ S_{31} &: S_j \mid\equiv U_i \mid\equiv U_i \overset{TimeSetup}{\rightleftharpoons} S_j \\ S_{32} &: S_j \mid\equiv U_i \mid\equiv T_{u_2}(x) \end{aligned} \tag{47}$$

(22) By S_{29}, A_{17}, and R_5, it is easy to obtain

$$S_{33} : S_j \mid\equiv U_i \overset{ID_i}{\rightleftharpoons} S_j \ \text{(Goal 8)} \tag{48}$$

(23) By S_{32}, $k_{\text{session}} = T_v(T_{u_2}(x))$, and difficulty assumptions, we can deduce that

$$S_{34} : S_j \mid\equiv U_i \mid\equiv U_i \overset{k_{session}}{\longleftrightarrow} S_j \tag{49}$$

(24) By S_{34}, A_{18}, and R_5, it is easy to obtain

$$S_{35} : S_j \mid\equiv U_i \overset{k_{session}}{\longleftrightarrow} S_j \ \text{(Goal 10)} \tag{50}$$

(25) From message 4, it is easy to have the following statement:

$$S_{36} : U_i \lhd \left\{U_i \overset{ID_i}{\rightleftharpoons} S_j, U_i \overset{ID_j}{\rightleftharpoons} S_j, T_v(x), TimeStamp_4\right\}_{Y_{ij}} \tag{51}$$

(26) By S_{23}, S_{36}, and R_5, it is easy to obtain

$$S_{37} : U_i \mid\equiv S_j \mid\sim \left(U_i \overset{ID_i}{\rightleftharpoons} S_j, U_i \overset{ID_j}{\rightleftharpoons} S_j, T_v(x), TimeStamp_4\right) \tag{52}$$

(27) By A_3 and R_{11}, it is easy to obtain

$$S_{38} : U_i \mid\equiv \#\left(U_i \overset{ID_i}{\rightleftharpoons} S_j, U_i \overset{ID_j}{\rightleftharpoons} S_j, T_v(x), TimeStamp_4\right) \tag{53}$$

(28) By S_{37}, S_{38}, and R_4, it is easy to obtain

$$S_{39} : U_i \mid\equiv S_j \mid\equiv \left(U_i \overset{ID_i}{\rightleftharpoons} S_j, U_i \overset{ID_j}{\rightleftharpoons} S_j, T_v(x), TimeStamp_4\right) \tag{54}$$

(29) By S_{39} and R_{14}, it is easy to obtain

$$\begin{aligned} S_{40} &: U_i \mid\equiv S_j \mid\equiv U_i \overset{ID_i}{\rightleftharpoons} S_j \\ S_{41} &: U_i \mid\equiv S_j \mid\equiv U_i \overset{ID_j}{\rightleftharpoons} S_j \\ S_{42} &: U_i \mid\equiv S_j \mid\equiv T_v(x) \end{aligned} \tag{55}$$

(30) By S_{40}, A_8, and R_5, it is easy to obtain

$$S_{43} : U_i \mid\equiv U_i \overset{ID_i}{\rightleftharpoons} S_j \ \text{(Goal 7)} \tag{56}$$

(31) By S_{42}, $k_{session} = T_{u_2}(T_v(x))$, and difficulty assumptions, we can deduce that

$$S_{44} : U_i \mid\equiv S_j \mid\equiv U_i \overset{k_{session}}{\longleftrightarrow} S_j \tag{57}$$

(32) By S_{44}, A_9, and R_5, it is easy to obtain

$$S_{45} : U_i \mid\equiv U_i \overset{k_{session}}{\longleftrightarrow} S_j \ \text{(Goal 9)} \tag{58}$$

Through (Goal 1)…(Goal 10), we have proved that the user and the authentication center believe that they share a ticket, and the user and the sever believe that they share a session key.

4.3. Resisting Stolen/Lost Smart Card Attack. If the smart card is stolen/lost by the adversary, the adversary can extract the information $\{Z_i, M_i, P_i, r, l, Gen(\cdot), Rep(\cdot,\cdot), h(\cdot), x, p\}$ stored in the smart card, where $Z_i = h(ID_i \parallel s_a) \oplus h(ID_i \parallel PW_i \parallel R_i \parallel r)$ and $M_i = h(h(ID_i \parallel PW_i \parallel r) \bmod l) \oplus R_i$. But, the adversary only knows the value of r. Obviously, he or she cannot obtain U_i's identification or password. So our proposed protocol could withstand the stolen/lost smart card attack.

4.4. Resisting Replay Attack. In our protocol, the mechanism of timestamp is included in each message. Then U_i, S_j, and AC could detect the replay of some message by checking the freshness of the timestamp. Therefore, our new protocol can withstand the replay attack.

4.5. Resisting Man-in-the-Middle Attack. If the adversary carries out the man-in-the-middle attack, he or she needs to choose a $T_{u_2}(x)$ and compute a valid Aut_3. However, the adversary cannot get Y'_{ij} and $TimeSetup$ included in Aut_3. Thus, the adversary cannot compute a valid Aut_3. Similarly, the adversary cannot also compute a valid Aut_4. Therefore, our new protocol can withstand the man-in-the-middle attack.

4.6. Mutual Authentication. Our new protocol achieves mutual authentication both between U_i and AC and between U_i and S_j.

Mutual authentication between U_i and AC: in Step 2 of authentication phase, AC computes $X_i' = h(ID_i' \| s_a)$ and checks the legitimacy of U_i by checking whether Aut_1 is equal to $h(ID_i' \| ID_j \| X_i' \| T'_{U_i-\mathrm{SA}} \| TimeStamp_1)$, because only U_i with the correct password and smart card has the knowledge of the secret $X_i = h(ID_i \| s_a)$ and the capability of generating the valid value $Aut_1 = h(ID_i \| ID_j \| X_i \| T_{U_i-\mathrm{SA}} \| TimeStamp_1)$. AC can ensure that U_i is really who he or she claims. In Step 3 of authentication phase, U_i checks the legitimacy of AC by checking whether Aut_2 is equal to $h(ID_i \| ID_j \| X_i \| Y_{ij}' \| TimeSetup \| TimeStamp_2)$, because only AC with the master key s_a can compute secret $X_i = h(ID_i \| s_a)$ and $Aut_2 = h(ID_i' \| ID_j \| X_i' \| Y_{ij} \| TimeSetup \| TimeStamp_2)$. U_i can ensure that he or she is communicating with the real AC.

Mutual authentication between U_i and S_j: in Step 4 of authentication phase, S_j computes $Y_{ij}' = h(ID_i' \| sk_j \| TimeSetup')$and checks the legitimacy of U_i by checking whether Aut_3 is equal to $h(ID_i' \| ID_j \| Y_{ij} \| T'_{U_i-S_j} \| TimeSetup' \| TimeStamp_3)$, because only U_i verified by AC has the knowledge of the ticket Y_{ij} and the capability of generating the valid value $Aut_3 = h(ID_i \| ID_j \| Y_{ij}' \| T_{U_i-S_j} \| TimeSetup \| TimeStamp_3)$. S_j ensured that U_i is really who he or she claims. In Step 5 of authentication phase, U_i checks the legitimacy of S_j by checking whether Aut_4 is equal to $h(ID_i \| ID_j \| Y_{ij} \| k_{session} \| TimeStamp_4)$, because only S_j with the private key sk_j can compute the ticket $Y_{ij}' = h(ID_i' \| sk_j \| TimeSetup')$ and $Aut_4 = h(ID_i' \| ID_j \| Y_{ij}' \| k_{session} \| TimeStamp_4)$. U_i ensured that he or she is communicating with a legitimate S_j.

4.7. Anonymity. In our protocol, the user's identity ID_i is involved in NID_{ij}, NY_{ij}, and $NID_{ij}{}^*$, which is encrypted with $T_{U_i-\mathrm{SA}}$ and $T_{U_i-\mathrm{S_j}}$. The adversary cannot get ID_i without knowing the random number u, v, and the AC's private key k_0, because $T_{U_i-\mathrm{SA}}$ and $T_{U_i-S_j}$ are computationally infeasible because of the hardness of the extended chaotic-map-based CDHP. Thus the adversary cannot extract the user's real identity ID_i. Therefore, our protocol achieves user anonymity.

4.8. Ticket Security. If the adversary wants to get the ticket Y_{ij}, he or she can only retrieve it from $NY_{ij} = (Y_{ij} \| TimeSetup) \oplus h(X_i{}')$. However, through Section 4.3 we know that the attacker could not get secrecy X_i even if the smart card was lost or stolen. Thus the adversary cannot compute $h(X_i{}')$ and Y_{ij}.

Moreover, the server's identity ID_j is involved in the NID_{ij}, NY_{ij}, and $NID_{ij}{}^*$, which is encrypted with $T_{U_i-\mathrm{SA}}$ and $T_{U_i-S_j}$. The adversary cannot get ID_j, because $T_{U_i-\mathrm{SA}}$ and $T_{U_i-S_j}$ are computationally infeasible because of the hardness of the extended chaotic-map-based CDHP. So though the adversary get the ticket Y_{ij}, he or she does not know which sever to access with Y_{ij}.

4.9. Perfect Forward Secrecy. In our protocol, the established session key is $k_{session} = T_{u_2 \cdot v}(x) = T_{u_2}(T_v(x)) = T_v(T_{u_2}(x))$, where u_2 and v are random numbers selected by the user and the sever, respectively. Previously established session keys remain secure even when the long-term keys of the server and the user are disclosed, because the adversary is computationally infeasible to calculate the session key with $T_{u_2}(x)$ and $T_v(x)$ because of the hardness of the extended chaotic-map-based CDHP.

4.10. Security Features Comparisons. We compare the security features of the proposed protocol with those of the previous multifactor authentication protocols for multiserver environment, including Jiang et al.'s [19], Wu et al.'s [20], and Das's [27].

Table 3 shows the results of the security features comparisons. From Table 3, we note that Jiang et al.'s protocol does not achieve user anonymity. Wu et al.'s and Das et al.'s protocol cannot resist stolen/lost smart card attack. Table 3 shows that our new protocol is the only one that is free from security attacks and provides anonymity and perfect forward secrecy.

5. Efficiency Analysis

This section compares the efficiency of the proposed protocol with that of the previous multifactor authentication protocols for multiserver environment, including Jiang et al.'s [19], Wu et al.'s [20], and Das's [27]. Table 4 shows separately the results of the security features comparisons and the efficiency comparisons.

To simplify the presentation, the following symbols are defined. $T_{Che}, T_{EcM}, T_H, T_S, T_M, T_{QR}$ denote the time for executing $T_n(x) \bmod p$ in Chebyshev polynomial using the algorithm in the literature [28], ECC point multiplication, the hash, the symmetric encryption/decryption, the modular squaring, and the computation of a square root modulo N, respectively. To be more precise, on an Intel Pentium4 2600 MHz processor with 1024 MB RAM, where n and p are 1024 bits long, $T_{Che}, T_{EcM}, T_H, T_S$ is 21 ms, 63.1ms, 0.5ms, and 8.7ms, respectively [28, 29]. The computational time of the bit XOR operation and multiplication operation is ignored compared with the above operations.

Table 4 shows that our proposed protocol has better efficiency than the protocols of Wu et al. and Das et al. Although the protocol of Jiang et al. has slightly better efficiency than our proposed protocol, it cannot accomplish anonymity. Besides, as any user accesses any server, the participation of the authentication center is required. When the number of users is huge, the computational cost of the authentication center may be very high, which could cause the authentication center crashing. In our protocol, when the ticket is valid, there is no need for the authentication center to participate. What is more, the total computational cost is greatly reduced. Overall, compared with other schemes, our

Table 3: Comparison of security features.

	Jiang et al.[19]	Wu et al.[20]	Das[27]	Our protocol
Resisting stolen/lost smart card attack	✔	✖	✖	✔
Resisting replay attack	✔	✔	✔	✔
Resisting man-in-the-middle attack	✔	✔	✔	✔
Mutual authentication	✔	✔	✔	✔
Anonymity	✖	✔	✖	✔
Perfect forward secrecy	✔	✔	✖	✔

Table 4: Efficiency comparisons.

	User	Authentication center	Server	Computational cost
Jiang et al.[19]	$8T_H + T_M$	$12T_H + T_{QR}$	$5T_H$	$25T_H + T_M + T_{QR}$
Wu et al.[20]	$11T_H + 2T_{EcM}$	$10T_H$	$3T_H + 2T_{EcM}$	$24T_H + 4T_{EcM}$
Das[27]	$7T_H + T_{EcM}$	$2T_H + T_S$	$2T_H + T_S$	$11T_H + T_{EcM} + 2T_S$
Our protocol	$6T_H + T_{Che}\,(2T_H + T_{Che})^*$	$4T_H + T_{Che}$	$3T_H + 2T_{Che}$	$13T_H + 4T_{Che}\,(5T_H + 3T_{Che})^*$

∗ The computational cost when the ticket is not expired.

scheme is more in accordance with the actual application requirements while ensuring the security and efficiency.

6. Conclusion

In this paper, we propose a novel multiserver authentication protocol based on the extended Chebyshev chaotic map with multifactors for cloud service. In our protocol, we designed a ticket for achieving mutual authentication between the user and the server which is innovative. When the ticket is valid, there is no need for authentication center to participate in the authentication process, which further reduces the burden of the authentication center. The ticket has an expiry time which is determined by the server according to specific requirement. Compared with the Kerberos protocol, there is no need to share a secret key in advance between the authentication center and the server.

Efficiency analysis shows that our protocol can resist a variety of attacks and provide the desirable security features. Compared with the existing schemes, the new protocol accomplishes various security and usability features necessary for all the participants, while maintaining relative high efficiency. Therefore, our scheme is more suitable for practical application.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper is supported by National Key Research and Development Program (nos. 2016YFB0800101 and 2016YFB0800100), Innovative Research Groups of the National Natural Science Foundation of China (Grant no. 61521003), and National Natural Science Foundation of China (Grants nos. 61379150 and 61309016).

References

[1] B. Hayes, "Cloud Computing," *Communications of the ACM*, vol. 51, no. 7, pp. 9–11, 2008.

[2] M. Armbrust, A. Fox, R. Griffith et al., "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[3] H. Takabi, J. B. D. Joshi, and G.-J. Ahn, "Security and privacy challenges in cloud computing environments," *IEEE Security & Privacy*, vol. 8, no. 6, pp. 24–31, 2010.

[4] J. Bonneau, C. Herley, P. C. Van Oorschot, and F. Stajano, "The quest to replace passwords: a framework for comparative evaluation of web authentication schemes," in *Proceedings of the 33rd IEEE Symposium on Security and Privacy*, pp. 553–567, San Francisco, Calif, USA, May 2012.

[5] M. Adeptus, *Hashdumps and Passwords*, 2014, http://www.adeptus-mechanicus.com/codex/hashpass/hashpass.php.

[6] C.-C. Chang and T.-C. Wu, "Remote password authentication with smart cards," *IEE Proceedings Part E Computers and Digital Techniques*, vol. 138, no. 3, pp. 165–168, 1991.

[7] Q. Xie, D. S. Wong, G. Wang, X. Tan, K. Chen, and L. Fang, "Provably secure dynamic ID-based anonymous two-factor authenticated key exchange protocol with extended security model," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1382–1392, 2017.

[8] F. Wang, Y. Xu, H. Zhang, Y. Zhang, and L. Zhu, "2FLIP: A two-factor lightweight privacy-preserving authentication scheme for VANET," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 2, pp. 896–911, 2016.

[9] Q. Jiang, N. Kumar, J. Ma et al., "A privacy-aware two-factor authentication protocol based on elliptic curve cryptography for wireless sensor networks," *International Journal of Network Management*, vol. 27, no. 3, 2016.

[10] D. Wang, D. He, P. Wang, and C.-H. Chu, "Anonymous Two-Factor Authentication in Distributed Systems: Certain Goals

Are Beyond Attainment," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 4, pp. 428–442, 2015.

[11] D. Wang and P. Wang, "Two Birds with One Stone: Two-Factor Authentication with Security Beyond Conventional Bound," *IEEE Transactions on Dependable & Secure Computing*, vol. 99, 2016.

[12] Q. Jiang, Z. Chen, and B. Li, "Security analysis and improvement of bio-hashing based three-factor authentication scheme for telecare medical information systems," *Journal of Ambient Intelligence Humanized Computing*, vol. 5, pp. 1–13, 2017.

[13] Z. Tan, "An efficient biometrics-based authentication scheme for telecare medicine information systems," *Przegląd Elektrotechniczny*, vol. 89, no. 5, pp. 200–204, 2013.

[14] X. Yan, W. Li, P. Li, J. Wang, X. Hao, and P. Gong, "A secure biometrics-based authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 37, no. 5, article no. 9972, 2013.

[15] Z. Tan, "A user anonymity preserving three-factor authentication scheme for telecare medicine information systems," *Journal of Medical Systems*, vol. 38, article 16, 2014.

[16] H. Arshad and M. Nikooghadam, "Three-Factor Anonymous Authentication and Key Agreement Scheme for Telecare Medicine Information Systems," *Journal of Medical Systems*, vol. 38, no. 12, 2014.

[17] D. He, S. Zeadally, N. Kumar et al., "Anonymous Authentication for Wireless Body Area Networks With Provable Security," *IEEE Systems Journal*, vol. 99, pp. 1–12, 2016.

[18] R. Amin, S. H. Islam, G. P. Biswas, M. K. Khan, L. Leng, and N. Kumar, "Design of an anonymity-preserving three-factor authenticated key exchange protocol for wireless sensor networks," *Computer Networks*, vol. 101, pp. 42–62, 2016.

[19] Q. Jiang, S. Zeadally, J. Ma, and D. He, "Lightweight three-factor authentication and key agreement protocol for internet-integrated wireless sensor networks," *IEEE Access*, vol. 5, pp. 3376–3392, 2017.

[20] F. Wu, L. Xu, S. Kumari, and X. Li, "An improved and provably secure three-factor user authentication scheme for wireless sensor networks," *Peer-to-Peer Networking and Applications*, pp. 1–20, 2016.

[21] M. Zhang, J. Zhang, and W. Tan, "Remote three-factor authentication protocol with strong robustness for multi-server environment," *China Communications*, vol. 14, no. 6, pp. 126–136, 2017.

[22] P. Bergamo, P. D'Arco, A. De Santis, and L. Kocarev, "Security of public-key cryptosystems based on Chebyshev polynomials," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 52, no. 7, pp. 1382–1393, 2005.

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

[24] X. Wang and J. Zhao, "An improved key agreement protocol based on chaos," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 12, pp. 4052–4057, 2010.

[25] Y. Dodis, L. Reyzin, and A. Smith, "Fuzzy extractors: how to generate strong keys from biometrics and other noisy data," in *Advances in Cryptology—EUROCRYPT 2004*, pp. 523–540, Springer, Berlin, Germany, 2004.

[26] C. Boyd and W. Mao, "On a limitation of BAN logic," in *Proceedings of the The Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology*, pp. 240–247, Springer-Verlag, New York, NY, USA, 1994.

[27] A. K. Das, "A Secure and Efficient User Anonymity-Preserving Three-Factor Authentication Protocol for Large-Scale Distributed Wireless Sensor Networks," *Wireless Personal Communications*, vol. 82, no. 3, pp. 1377–1404, 2015.

[28] *Chaos-based cryptography: Theory, algorithms and applications*, Springer, 2011.

[29] Y. Sun, H. Zhu, and X. Feng, "A novel and concise multi-receiver protocol based on chaotic maps with privacy protection," *International Journal of Network Security*, vol. 19, no. 3, pp. 371–382, 2017.