Hindawi
Mobile Information Systems
Volume 2022, Article ID 7318948, 11 pages
https://doi.org/10.1155/2022/7318948

Research Article

CCMbAS: A Provably Secure CCM-Based Authentication Scheme for Mobile Internet

Yu Zhang,[1,2] **Guangmin Sun**,[1] **and Peng Zhai**[1,2]

[1]*Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China*
[2]*Institute of Mathematics and Computer Science, Jining University, Qufu 273155, China*

Correspondence should be addressed to Yu Zhang; b201602005@jnxy.edu.cn

Received 14 July 2022; Revised 17 September 2022; Accepted 21 September 2022; Published 12 October 2022

Academic Editor: Dragan Pamučar

To improve the security of authentication system and strengthen privacy protection in mobile Internet environment, this paper proposes a provably secure Chebyshev chaotic map (CCM)-based authentication scheme (CCMbAS). The proposed scheme transformed the traditional public key of Chebyshev chaotic map into a private key and combined two private keys to compute a one-time key used to encrypt authentication information. The scheme is verified using security review of BAN logic and ProVerif simulation tool. The verification results confirm that the scheme is well secured against all existing security threats. Compared with similar schemes, the proposed scheme is more efficient and secure. The security analysis shows that the proposed scheme can fulfil secure demands and ensure the security of user's information in mobile Internet environment.

1. Introduction

Mobile Internet is the Internet and service that takes mobile network as access network. It has the characteristics of openness and complexity. With the rapid upgradation of mobile communication and the wide application of intelligent terminal, the application services provided by mobile Internet are more and more widespread. However, the network environment is more and more complex. Identity authentication is the first defender of information system, which can guarantee the security of system data and user information in complex network environment. It plays a key role in application system.

Aiming at solving the security threat of identity authentication system and protecting user's privacy information, Zhu et al. [1] proposed a biometrics-based multiserver key agreement scheme (BbKAS) on chaotic map cryptosystem. The encryption key of the scheme is not secure enough because the attacker can obtain encrypting key and crack encrypted information with dictionary attack of the intercepting information. Jiang et al. [2] proposed a new three-factor scheme. Ali et al. [3] proposed a three-factor identity authentication scheme based on RSA encryption algorithm.

To reduce the computational cost, Dong et al. [4] proposed a biometric verification-based authentication scheme (BVbAS) using Chebyshev chaotic mapping. The design of the scheme is unreasonable because the registry centre must provide all concerned information about all users and servers to each other before they request authentication. The design may result in a sharp increase in the communication cost of system. Otherwise, the authentication cannot be performed.

In general, the schemes can be classified into five groups in terms of the underlying intractability problem: based on discrete-logarithm problem [5–8], based on pairing [9, 10], based on chaotic map [11–13], based on integer-factorization problem [14], and based on hash function [15–18]. Among them, schemes based on elliptic curve bilinear pairings, such as a robust provable-secure privacy-preserving authentication protocol (PpAP) for Industrial Internet of Things [10], usually require large computation cost. Chaotic cryptography has become increasingly popular due to its lower computational complexity and higher asymmetric key security [19]. In view of the computing and security advantages of chaotic cryptography, CCMbAS is proposed to solve the problems of the above schemes.

2. Related Theoretical Knowledge

2.1. Fuzzy Extractor. In order to solve the contradiction between the variability of extracted biometric feature data and the input stability of traditional cryptography, Dodis proposed an algorithm of fuzzy extractors [20]. The algorithm could keep the numerical consistency of output results in the case of slight differences in the extracted biometric features.

Fuzzy extractor includes generation function $Gen(\cdot)$ and reproduction function $ep(\cdot)$, and $Gen(\cdot)$ is a probabilistic generating function. When the user inputs a biometric feature BIO_i, the function will generate a random string b_i limited to a fixed length ($b_i \in \{0,1\}^m$) and a public reproduction parameter P_i (as an auxiliary string), namely, $Gen(BIO_i) = (b_i, P_i)$, and $Rep(\cdot)$ is a deterministic reproduction function which can reproduce the biometric key according to the input biometric feature BIO_i' and corresponding public reproduction parameter P_i. If the Hamming distance between BIO_i and BIO_i' is within the preset fault tolerance threshold, $Rep(BIO_i', P_i) = b_i$. When $Gen(\cdot)$ and $Rep(\cdot)$ run in polynomial time, fuzzy extractor is very efficient. Without the aid of the original biometric feature, the biometric key cannot be reproduced with only the public reproduction parameter through calculation [4].

The application of fuzzy extractor can be effectively combined with cryptography in the field of authentication. In recent years, fuzzy extractor is used in many multi-factor authentication schemes [21–25].

2.2. Chebyshev Map

Definition 1. Chebyshev polynomial $\mathrm{Tn}(x)$ is the polynomial of n orders about x, where n is a natural number, $x \in [-1, 1]$, and $T_n(x) = \cos(n * \arccos(x))$.

According to trigonometric transformation, Chebyshev polynomial iterative relation can be obtained as follows: $T_0(x) = 1$, $T_1(x) = x$, ..., $T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x)$.

Definition 2. The cryptosystem based on Chebyshev polynomial has the risk that session key is intercepted. In order to remedy the security defect, Zhang et al. [26] extended the domain of x from $x \in [-1, 1]$ to $x \in (-\infty, +\infty)$ in 2008, that is, extended Chebyshev polynomial:

$$T_n(x) \equiv \left(2xT_{n-1}(x) - T_{n-2}(x)\right) \bmod p, \quad (1)$$

where ≥ 2, $x \in (-\infty, +\infty)$, and p is a big prime number. It still has the semigroup property:

$$T_r\left(T_s(x)\right) \equiv T_{rs}(x) \equiv T_s\left(T_r(x)\right) (\bmod\, p). \quad (2)$$

Definition 3. It is a very hard problem of discrete logarithm to get r with the value x and y ($T_r(x) = y$). It is impossible in theory.

Definition 4. It is Diffie–Hellman problem to compute $T_{rs}(x)$ using $T_r(x)$ and $T_s(x)$. It is also impossible in theory.

Table 1: Symbol definitions.

Symbol	Definition
$(x, T_k(x))$	Public key of CA
k	Private key of CA
N	A large prime number
ID_i	The user's identity
PW_i	The user's password
ID_j	The server's identity
$Gen(\cdot)$	Generation function of fuzzy extractor
$Rep(\cdot)$	Reproduction function of fuzzy extractor
α_i	Public reproduction parameter
B_i	Biometric feature
b_i	Biometric key of the user U_i
$h(\cdot)$	Hash function
r_i, r_j	Random number
r_a, r_b, r_c	Random number
Δt	Valid time interval
$\oplus$	XOR
$\|\|$	Concatenation
t_i, t_j, t_1, t_2, t_3	Current time

3. Scheme Design

The authentication system consists of three parts: certificate authority (CA), user terminal, and server.

CA includes registration module, important data management module, and user authority management module.

User terminal includes registration module, biometric feature authentication module, password verification module, important data management module, and application interface module.

Server includes registration module, key agreement module, important data management module, and application platform interface module.

3.1. Symbol Definitions. The symbol definitions of the proposed scheme are shown in Table 1.

3.2. System Settings. CA first generates a private key k (assuming that the key is absolutely secure), then selects a random string x, and generates $T_k(x)$ through Chebyshev chaos map. The public key is published. The private key is hidden.

3.3. Registration Phase

3.3.1. Server Registration Phase. The registration process of the server is shown in Figure 1.

Step 1. The server S_j selects a unique identity ID_j and sends ID_j and the current time t_j to CA via secure channel.

Step 2. After receiving the registration request message $\{ID_j, t_j\}$ from S_j, CA first checks whether the time t_j exceeds the maximum valid time interval Δt or not. If the time interval meets the requirements, CA then checks whether the identity ID_j of the server is registered already or not. If the identity ID_j is registered

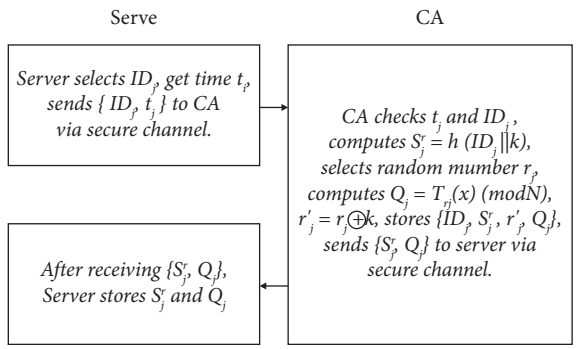

Figure 1: Server registration phase.

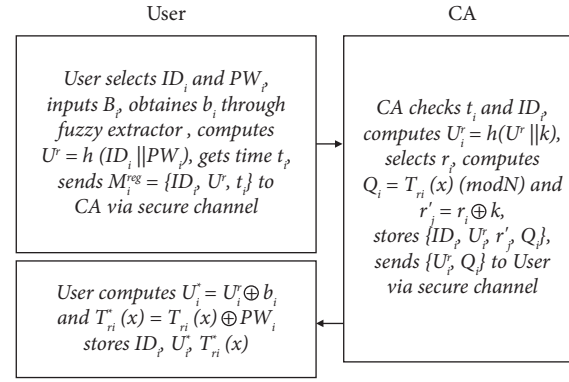

Figure 2: User registration phase.

already, CA rejects the registration request. Otherwise, CA computes $S_j^r = h(ID_j \| k)$, selects a random number r_j, and computes the key $Q_j = T_{r_j}(x)$ and $r'_j = r_j \oplus k$. The key Q_j cannot be published. Then, CA stores the data $\{ID_j, S_j^r, r'_j, Q_j\}$ in the important data management module and sends the message $\{S_j^r, Q_j\}$ to server S_j via secure channel.

Step 3. After receiving $\{S_j^r, Q_j\}$, the server S_j stores them in the important data management module.

3.3.2. User Registration Phase. The process is shown in Figure 2.

Step 1. The user U_i selects a unique identity ID_i and W_i. Then, the biometric sample B_i is input through the sensor of biometric authentication module. The biometric key b_i is obtained by using the fuzzy extractor and its public reproduction parameter α_i. That is, $(B_i) = (b_i, \alpha_i)$.

Step 2. The user U_i computes $U^r = h(ID_i \| PW_i)$, gets current time t_i, and then sends the registration information $M_i^{reg} = \{ID_i, U^r, t_i\}$ to CA via secure channel.

Step 3. After receiving $M_i^{reg} = \{ID_i, U^r, t_i\}$, CA first checks whether the time t_i exceeds the maximum time interval Δt or not. If it exceeds the maximum time interval, CA rejects the user's request. If the result is eligible, CA checks whether the identity is registered already or not. CA forbids the user to register again. If the identity ID_i is not registered, CA calculates $U_i^r = h(h(ID_i \| PW_i) \| k)$, selects a random number r_i, and calculates the key $Q_i = T_{r_i}(x)$ (the public key transformed into private key) and $r'_i = r_i \oplus k$. Then, CA stores $\{ID_i, U_i^r, r'_i, Q_i\}$ in the important data management module and sends $\{U_i^r, Q_i\}$ to user U_i via the secure channel.

Step 4. After receiving $\{U_i^r, Q_i\}$, user U_i calculates $U_i^* = U_i^r \oplus b_i$ and $T_{r_i}^*(x) = T_{r_i}(x) \oplus PW_i$ and then stores the information $ID_i, U_i^*, T_{r_i}^*(x)$ in the important data management module.

3.4. Login, Authentication, and Key Agreement Phase. If the user requests to login to the server, successfully authenticates his identity, and accesses resources, he/she must perform the steps shown in Figure 3.

Step 1. The user U_i inputs biometric feature through the sensor of biometric feature authentication module and uses fuzzy extractors and its public reproduction parameter α_i to obtain biometric key b'_i by calculating function $Rep(B'_i, \alpha_i) = b'_i$. When the Hamming distance from B'_i and B_i is only less than the default tolerance threshold value, the equation $b'_i = b_i$ can be set up and the user U_i can pass biometric feature authentication. Then, the user U_i calculates $U_i^r = U_i^* \oplus b_i$.

Step 2. The user U_i inputs the correct password W_i and calculates the equation $T_{r_i}(x) = T_{r_i}^*(x) \oplus PW_i$ so as to pass password authentication.

Step 3. The user U_i selects a random number r_a as the temporary private key, calculates $k_1 = T_{r_a}(T_{r_i}(x))$, $M_i = h(ID_i \| ID_j \| U_i^r) \oplus k_1$, and $M_1 = \{M_i, ID_i, T_{r_a}(x)\}$, obtains the current time t_1, and sends the message $\{M_1, t_1\}$ to server S_j via public network. The key k_1 is the one-time key generated by calculation after the combination of the private key r_a and Q_i,

Step 4. After receiving $\{M_1, t_1\}$, server S_j first checks whether the time t_1 exceeds the maximum time interval Δt or not. If it exceeds the maximum time interval, the server rejects the user's request. If the result is eligible, server S_j selects a random number r_b as the temporary private key, calculates $k_2 = T_{r_b} T_{r_j}(x)$, $M_j = h(ID_i \| ID_j \| S_j^r) \oplus k_2$, and $M_2 = \{M_j, ID_j, T_{r_b}(x), M_1\}$, then obtains the current time t_2, and sends $\{M_2, t_2\}$ to CA via public network. The key k_2 is the one-time key generated by calculation after the combination of the private key r_b and Q_j.

Step 5. After receiving $\{M_2, t_2\}$, CA first checks whether the time t_2 exceeds the maximum time interval or not. If it exceeds the maximum time interval, CA rejects the request. If it is eligible, CA calculates $r_j = r'_j \oplus k$, $k'_2 = T_{r_j}(T_{r_b}(x))$, $M'_j = M_j \oplus k'_2$, and $h(ID_i \| ID_j \| S_j^r)$ and verifies $h(ID_i \| ID_j \| S_j^r) = M'_j$. If the result is equal, CA authenticates the server S_j.

Step 6. Based on $M_1 = \{M_i, ID_i, T_{r_a}(x)\}$, CA calculates $r_i = r_i \oplus k$, $k'_1 = T_{r_i}(T_{r_a}(x))$, $M'_i = M_i \oplus k'_1$, and $h(ID_i \| ID_j \| U_i^r)$ and verifies $h(ID_i \| ID_j \| U_i^r) = M'_i$. If the result is not equal, CA stops authentication. If the result is equal, CA can authenticate the user U_i which applies for accessing the server S_j.

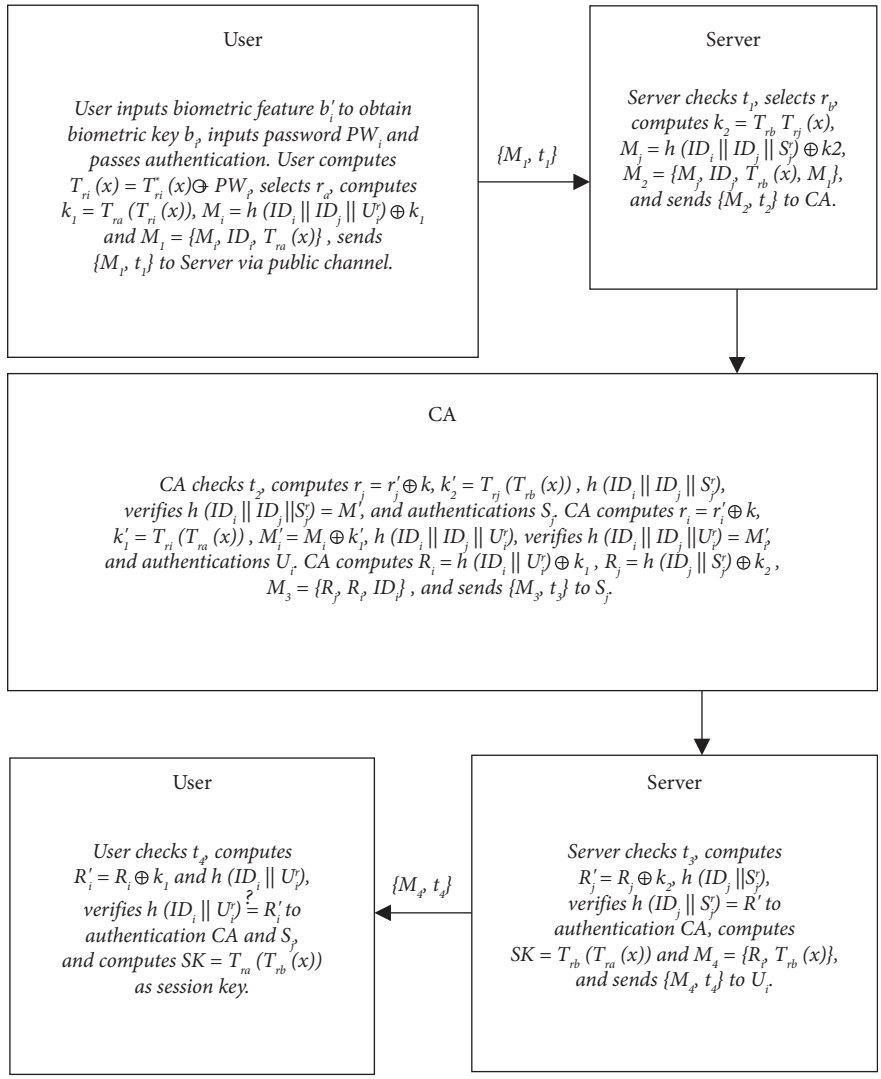

Figure 3: Login, authentication, and key agreement phase.

Step 7. CA calculates $R_i = h(ID_i \| U_i^r) \oplus k_1$, $R_j = h(ID_j \| S_j^r) \oplus k_2$, and $M_3 = \{R_j, R_i, ID_i\}$, obtains the current time t_3, and sends $\{M_3, t_3\}$ to the server S_j via public network.

Step 8. After receiving $\{M_3, t_3\}$, server S_j first checks whether the time t_3 exceeds the maximum time interval or not. If it exceeds the maximum time interval, the server will discard the received information. If the result is eligible, the server fetches R_j from M_3 and calculates $R_j' = R_j \oplus k_2$ and $h(ID_j \| S_j^r)$. Then, the server verifies $h(ID_j \| S_j^r) = R_j'$. If the result is not equal, the server stops authentication. If the result is equal, the server can authenticate CA. Then, the server calculates the session key $SK = T_{r_b}(T_{r_a}(x))$ which will be used with the user U_i, gets the current time t_4, and sends $\{M_4, t_4\} M_4 = \{R_i, T_{r_b}(x)\}$) to the user U_i via public network.

Step 9. After receiving $\{M_4, t_4\}$, user U_i first checks whether the time t_4 exceeds the maximum time interval. If it oversteps the maximum time interval, user U_i will discard the received information. If the result is eligible, the user calculates $R_i' = R_i \oplus k_1$ and $h(ID_i \| U_i^r)$. Then, the user verifies $h(ID_i \| U_i^r) \stackrel{?}{=} R_i'$. If the result is not equal, the user stops authentication. If the result is equal, the user can authenticate CA and the server S_j. Then, the user calculates the session key $SK = T_{r_a}(T_{r_b}(x))$ which will be used with the server S_j.

3.5. Password Change. If the user wants to change the password, the authentication must be completed of the user on the terminal first. Then, the user changes the password according to the steps of registration. The corresponding information stored in the user terminal and the CA can be updated.

3.6. Identity and Biometric Feature Change. If the user needs to change the identity, the identity can be changed by the similar steps of the password change. If the user needs to change the biometric feature, the biometric feature can be changed after the terminal authenticates the legitimate user.

4. Scheme Security

4.1. Security Analysis

4.1.1. Key Security. The user's biometric key b_i is generated by fuzzy extractor, so the attacker cannot get the user's biometric key through the fuzzy extractor without the user's biometric feature. In the proposed scheme, a double key combined encryption mechanism is designed. For example, the key k_1 is the one-time key generated by calculation after the combination of the private key r_a and Q_i. Because the one-time key k_1 is newly generated, the information encrypted with k_1 is difficult to crack. The user calculates $F_i^* = F_i \oplus b_i$, $U_i^* = U_i^r \oplus b_i$, and $T_{r_i}^*(x) = T_{r_i}(x) \oplus PW_i$ in order to hide U_i^r and $T_{r_i}(x)$ and then stores the information $ID_i, U_i^*, T_{r_i}^*(x)$ into the important data management module. Suppose that attacker can obtain the data stored in the user's terminal, and the encrypted information cannot be decrypted. Therefore, the information b_i, U_i^r, and $T_{r_i}(x)$ cannot be leaked or stolen.

4.1.2. Terminal Lost Attack. If the terminal device is lost, authentication requires not only the correct biometric feature information but also the correct password. The user's secret information stored in the terminal device is encrypted data. The attacker cannot provide the correct information and decrypt the stored secret information. Therefore, the system can ensure the security of the secret information in the case of terminal device loss.

4.1.3. Password Guessing Attack. In this scheme, user authentication includes two steps. If user wants to login successfully, the biometric feature and password must be correct. Without biometric feature of the legitimate user, the attacker cannot pass the initial biometric feature authentication. Therefore, the attacker cannot proceed the second step, password authentication. The shared session key generated temporarily is new and different each time. Attacker cannot guess the session key. Therefore, authentication system can effectively avoid password guessing attack.

4.1.4. Impersonation Attack. Because user authentication includes biometric feature and password, the attacker cannot pass through password authentication when he initiates impersonation attack in case of obtaining the user's biometric feature. If an attacker impersonates a legitimate user or server to transmit information, the user, server, or CA can identify the authenticity of the sender through calculation and the impersonation attack information.

4.1.5. Eavesdropping Attack. The scheme uses the randomness of hash function value to hide the authentication information transmitted in the public network and uses the one-off key randomly generated by Chebyshev chaos map to encrypt the authentication information. Under the premise of this double security, the attacker cannot get useful information by eavesdropping on the messages transmitted in the public network.

4.1.6. Denial-of-Service Attack. Within a certain time period, CA does not allow users using the same ID to apply for registration. Therefore, CA can avoid excessive consumption of server resources and effectively defend against denial-of-service attack.

4.1.7. Man-in-the-Middle Attack. Even if information of legitimate users or servers is intercepted and tampered by attacker, the attacker cannot pass the inspection and authentication of users or servers. Therefore, the attacker cannot steal the content from the information of user and server by attack.

4.1.8. Replay Attack. Time information is added to the transmitted information in the proposed scheme, which has the function of time stamp and can effectively avoid replay attack.

4.1.9. Privileged Insider Attack. In this scheme, CA uses its own private key to perform XOR operation to the key of user or server to hide the important information. The password of user is protected by one-way hash function when applying for registration and authentication, which also achieves the purpose of hiding important information. In this way, privilege attack can be effectively avoided.

4.1.10. Forward Security. The encryption key of authentication information is one-off in the process of certification. The sharing session key is also one-off after key agreement. The scheme has dual security by hiding and encryption. The attacker cannot crack the former session key.

4.1.11. Mutual Authentication. In the proposed scheme, the shared session key calculated only by the legitimate user and server will be the same. Therefore, the scheme can realize mutual authentication among CA, user, and server. Meanwhile, the scheme can ensure the communication security between legitimate user and server.

The comparison results in terms of security are shown in Table 2.

4.2. BAN Logical Proof

4.2.1. BAN Logic. Among the cryptographic protocol formal verification methods, BAN logic proposed by Burrows et al. in 1989 is the well-known one [27]. BAN logic is a kind of modal logic based on belief, which mainly includes the following three processing objects: subject, key, and formula. P, Q, and R represent the subject variable. K represents the key variable. X and Y represent the formula variable. A and B represent the two common subjects. S is the authentication

Table 2: Security comparison.

Security issues	BbKAS [1]	BVbAS [4]	PpAP [10]	Proposed scheme
Forward security	×	√	√	√
Key security	×	√	√	√
Mutual authentication	√	×	√	√
Key agreement	√	×	√	√
Privilege insider attack	√	√	√	√
Resist man-in-the-middle attack	√	√	√	√
Resist replay attack	√	√	×	√
Resist password guessing attack	×	√	√	√
Resist terminal loss attack	√	√	√	√

Table 3: The syntax and semantics of the BAN logical component.

Symbol	Definition
$P\mid\equiv X$	Subject P believes X.
$P\lhd X$	Subject P receives the message X.
$P\mid\sim X$	Subject P has sent out the message X.
$P\mid\Rightarrow X$	Subject P has jurisdiction over X.
$\#(X)$	X is fresh.
$P\overset{K}{\leftrightarrow}Q$	K is the shared key of subjects A and B, which is unknown to other subjects.
$\overset{K}{\longrightarrow}P$	K is the public key of the subject P. The other subjects do not know the corresponding private key K^{-1}.
$\overset{K}{\underset{X}{\longrightarrow}}-1\ P$	K^{-1} is the private key of the subject P.
$P\overset{X}{\rightleftharpoons}XQ$	X is the shared secret between subjects P and Q, which is unknown to other subjects.
$\{X\}_K$	The ciphertext is obtained by encrypting X with the key K.
$\langle X\rangle_Y$	A cascade between message X and secret Y can prove that the message $\langle X\rangle_Y$ is sent by a certain subject.

Table 4: BAN logic inference rules.

Sequence number	Rules
R1	$P\mid\equiv Q\overset{K}{\leftrightarrow}P, P\lhd\{X\}_K\vdash P\mid\equiv Q\mid\sim X$
R2	$P\mid\equiv\ \overset{K}{\underset{Y}{\longrightarrow}}Q, P\lhd\{X\}_{K^{-1}}\vdash P\mid\equiv Q\mid\sim X$
R3	$P\mid\equiv P\overset{Y}{\rightleftharpoons}YQ, P\lhd\{X\}_Y\vdash P\mid\equiv Q\mid\sim X$
R4	$P\mid\equiv\#(X), P\mid\equiv Q\mid\sim X\vdash P\mid\equiv Q\mid\equiv X$
R5	$P\mid\equiv Q\mid\Rightarrow X, P\mid\equiv Q\mid\equiv X\vdash P\mid\equiv X$
R6	$P\lhd(X,Y)\vdash P\lhd X$
R7	$P\lhd\langle X\rangle_Y\vdash P\lhd X$
R8	$P\mid\equiv P\overset{K}{\leftrightarrow}Q, P\lhd\{X\}_K\vdash P\lhd X$
R9	$P\mid\equiv\ \overset{K}{\longrightarrow}P, P\lhd\{X\}_K\vdash P\lhd X$
R10	$P\mid\equiv P\overset{K}{\longrightarrow}Q, P\lhd\{X\}_{K^{-1}}\vdash P\lhd X$
R11	$P\mid\equiv\#(X),\ \vdash P\mid\equiv\#(X,Y)$
R12	$P\mid\equiv X, P\mid\equiv Y,\vdash P\mid\equiv(X,Y)$
R13	$P\mid\equiv(X,Y),\ \vdash P\mid\equiv X$
R14	$P\mid\equiv Q\mid\equiv(X,Y),\ \vdash P\mid\equiv Q\mid\equiv X$
R15	$P\mid\equiv Q\mid\sim(X,Y),\ \vdash P\mid\equiv Q\mid\sim X$
R16	$P\mid\equiv R\overset{K}{\leftrightarrow}R',\ \vdash P\mid\equiv R'\overset{K}{\leftrightarrow}R$
R17	$P\mid\equiv Q\mid\equiv R\overset{K}{\leftrightarrow}R',\ \vdash P\mid\equiv Q\mid\equiv R'\overset{K}{\underset{X}{\leftrightarrow}}R$
R18	$P\ \mid\equiv R\overset{X}{\rightleftharpoons}XR',\vdash P\mid\equiv R'\overset{X}{\rightleftharpoons}XR$
R19	$P\ \mid\equiv Q\ \mid\equiv R\overset{X}{\rightleftharpoons}XR',\vdash P\mid\equiv Q\mid\equiv R'\overset{X}{\rightleftharpoons}XR$

server. K_{ab}, K_{ac}, and K_{bc} represent the specific shared key. K_a, K_b, and K_c represent the specific public key. K_a^{-1}, K_b^{-1}, and K_c^{-1} represent the specific secret key. N_a, N_b, and N_c represent the temporary value. $h(X)$ represents the irreversible hash function of X.

(1) The syntax and semantics of the BAN logical component.

The syntax and semantics of the BAN logical component are shown in Table 3.

(2) BAN logic inference rules.

Message meaning rules: R1–R3. Nonce verification rule: R4. Jurisdiction rule: R5. Seeing rules: R6–R10. Freshness rules: R11. Belief rules: R12–R15. Key and secret rules: R16–R19. BAN logic inference rules are shown in Table 4.

4.3. Scheme Security Proof

(1) Initialization $S_j\lhd T_{r_a}(x)$, $U_i\lhd T_{r_b}(x)$

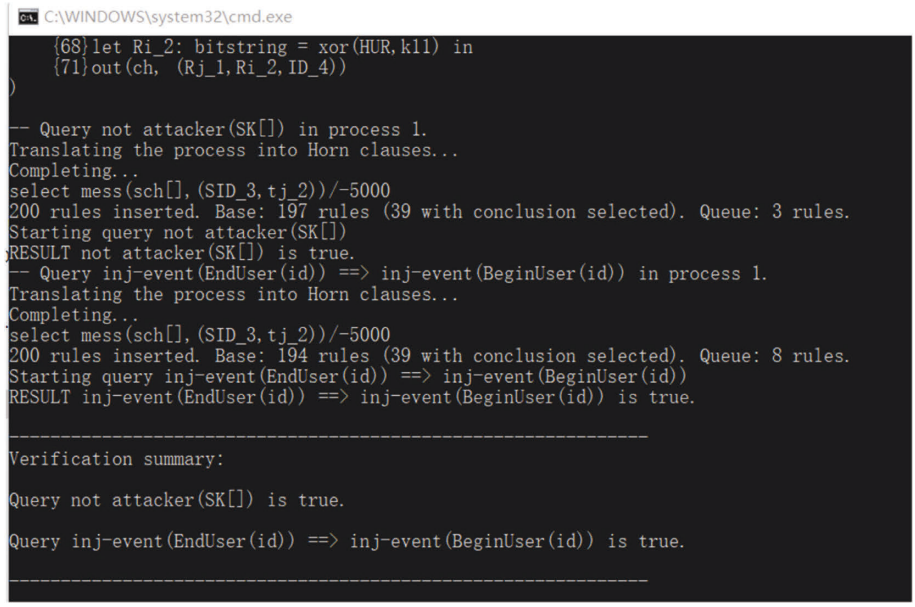

Figure 4: Performance result of ProVerif code.

(2) Establish security goals G1. $S_j| \equiv U_i| \equiv T_{r_a}(x)$, G2. $U_i| \equiv S_j| \equiv T_{r_b}(x)$, G3. $S_j| \equiv T_{r_a}(x)$, G4. $U_i| \equiv T_{r_b}(x)$, G5. $S_j| \equiv U_i| \equiv (T_{r_a}(x), T_{r_b}(x))$, G6. $U_i| \equiv S_j| \equiv (T_{r_a}(x), T_{r_b}(x))$

(3) Protocol formalization

F1. $U_i| \equiv T_{r_a}(x)$, $U_i| \Rightarrow T_{r_a}(x)$, F2. $S_j| \equiv T_{r_b}(x)$, $S_j| \Rightarrow T_{r_b}(x)$, F3. $S_j| \equiv U_i| \sim T_{r_a}(x)$, $S_j| \equiv \#(T_{r_a}(x))$, F4. $U_i| \equiv S_j| \sim T_{r_b}(x)$, $U_i| \equiv \#(T_{r_b}(x))$, F5. $S_j| \equiv U_i| \Rightarrow T_{r_a}(x)$, F6. $U_i| \equiv S_j| \Rightarrow T_{r_b}(x)$, F7. $S_j| \equiv U_i \overset{SK}{\leftrightarrow} S_j$, F8. $U_i| \equiv S_j \overset{SK}{\leftrightarrow} U_i$,

(4) Concrete proof process

V1. According to the rule R4 and formalization F3, $S_j| \equiv U_i| \sim T_{r_a}(x), S_j| \equiv \#(T_{r_a}(x)) \vdash S_j| \equiv U_i| \equiv T_{r_a}(x)$ can be got. Therefore, the goal G1 is true.

V2. In the same way of V1 above, according to the rule R4 and formalization F4, the goal G2 is true.

V3. According to the rule R5 and formalization F5, $S_j| \equiv U_i| \equiv T_{r_a}(x), S_j| \equiv U_i| \Rightarrow T_{r_a}(x) \vdash S_j| \equiv T_{r_a}(x)$ can be obtained. Therefore, the goal G3 is true.

V4. In the same way of V3 above, according to the rule R5 and formalization F6, the goal G4 is true.

V5. According to goal G3, formalization F2, and rule R12, $S_j| \equiv T_{r_a}(x), S_j| \equiv T_{r_b}(x) \vdash S_j| \equiv (T_{r_a}(x), T_{r_b}(x))$ can be obtained. Therefore, the goals G5 and G6 are true.

Basing on the BAN logic proof, the proposed authentication scheme can achieve the predetermined security goal, which proves that the scheme is secure.

4.4. ProVerif Verification

4.4.1. ProVerif Code

```
(* ----------channel---------- *)
free sch: channel [private]. (* ---secure channel---- *)
free ch: channel. (* ---unsecure channel---- *)
(* ----------variable and constants---------- *)
free ID: bitstring. (* ---User ID---- *)
const SID: bitstring. (* ---Application server ID---- *)
const x: bitstring. (* ---Seed for Chebyshev chaotic map ---- *)
const pw: bitstring [private]. (* ---password of user---- *)
free treg: bitstring. (* ---the time of registration---- *)
free s: bitstring [private]. (* ---key of application server---- *)
free u: bitstring [private]. (* ---key of user---- *)
free k: bitstring [private]. (* ---key of CA---- *)
free B: bitstring [private]. (* ---biometric of user---- *)
free w: bitstring [private]. (* ---parameter of fuzzy extraction algorithm---- *)
```

```
free SK: bitstring [private]. (* ---the session key between user and application server-- *)
(* ----------constructor---------- *)
fun H(bitstring): bitstring.
fun senc(bitstring, bitstring): bitstring.
fun T(bitstring, bitstring): bitstring. (* ---the Chebyshev chaotic map algorithm-- *)
fun xor(bitstring, bitstring): bitstring.
fun Concat(bitstring, bitstring): bitstring.
fun GEN(bitstring): bitstring. (* ---the GEN section of fuzzy extraction algorithm-- *)
fun REP(bitstring): bitstring. (* ---the REP section of fuzzy extraction algorithm-- *)
(* --------destructors&equations---------- *)
reduc forall m: bitstring, n: bitstring; sdec(senc(m,n),n) = m.
(* reduc forall a: bitstring, b: bitstring, x:bitstring; T(b,T(a,x)) = T(a,x) * T(b,x). ---the Chebyshev chaotic map algorithm-- *)
equation forall m: bitstring, n: bitstring; xor(xor(m,n),n) = m.
(* ----------events ---------- *)
event BeginUser(bitstring).
event EndUser(bitstring).
(* ----------query ---------- *)
query attacker(SK).
query id:bitstring; inj-event(EndUser(id)) ==>inj-event(BeginUser(id)).
(* ----------process---------- *)
(* ----------user process---------- *)
let user =
let (b) = GEN(B) in
let UR=H(Concat(pw, ID)) in
out(sch, (ID, UR, treg));
event BeginUser(ID);
in(sch,(URR:bitstring, Qi:bitstring)); (* -Input some data - *)
let URb = xor(URR, b) in
let Qip = xor(Qi, pw) in
new Bioaut: bitstring;
new pw: bitstring;
let b = REP(Bioaut) in
let Qi = xor(Qip, pw) in
new a: bitstring;
let tax = T(a,x) in
let k1 = T(u, tax) in
let HU=H(Concat(Concat(ID, SID), URR)) in
let MU = xor(HU, k1) in
out(ch,(MU, ID, tax));
in(ch,(Ri:bitstring, tbx:bitstring));
let Rii = xor(Ri, k1) in
let Hi = H(Concat(ID, URR)) in
if Hi = Rii then
let SK = T(a,tbx) in
0
).
(* ----------Application Server AS process---------- *)
let AS =
!
(
new tj:bitstring;
out(sch,(SID,tj));
in(sch,(SR:bitstring, Qj:bitstring));
let SR=H(Concat(SID,k)) in
let Qj = T(s,x) in
in(ch,(MU: bitstring, ID: bitstring, tax: bitstring));
new b:bitstring;
let tbx = T(b,x) in
let k2 = T(s, tbx) in
let HS=H(Concat(Concat(ID, SID), SR)) in
let MS = xor(HS, k2) in
out(ch,(MS, SID, tbx, MU, ID, tax));
in(ch,(Rj: bitstring, Ri: bitstring, ID: bitstring));
let Rjj = xor(Rj, k2) in
let Hj = H(Concat(SID, SR)) in
if Hj = Rjj then
let SK = T(b,tax) in
out(ch,(Ri, tbx));
event EndUser(ID);
0
).
(* ----------Certificate Authority CA process---------- *)
let CA =
in(sch, (ID:bitstring, UR:bitstring, treg:bitstring));

new u: bitstring;
let URR=H(Concat (UR,k)) in
let Qi = T(u,x) in
let ui = xor(u,k) in
out(sch,(URR, Qi));
in(sch, (SID:bitstring, tj: bitstring));
new s: bitstring;
let SR=H(Concat(SID,k)) in
let Qj = T(s,x) in
```

Table 5: Execution time comparison.

Scheme	Registration phase	Login, authentication, and key agreement phase	Password change phase
BbKAS [1]	$3T_H + T_{Gen} = 808T_H$	$12T_H + 6T_C + 8T_E + T_{Gen} = 845T_H$	$2T_H + T_{Gen} = 807T_H$
BVbAS [4]	$9T_H + T_{Gen} = 814T_H$	$21T_H + T_{Rep} + 4T_C = 216T_H$	$8T_H + T_{Rep} + T_{Gen} = 1000T_H$
PpAP [10]	$11T_H + 9T_P + 4T_A = 1201T_H$	$17T_H + 7T_P + 7T_A = 1277T_H$	$6T_H + 4T_P + 2T_A = 538T_H$
Proposed scheme	$3T_H + 2T_C + T_{Gen} = 814T_H$	$8T_H + T_{Rep} + 6T_C = 207T_H$	$2T_H + 2T_C + 4T_E + T_{Rep} = 201T_H$

let sj = xor(s,k) in
out(sch,(SR,Qj));

in(ch,(MS:bitstring, SID:bitstring, tbx:bitstring, MU: bitstring, ID: bitstring, tax: bitstring));
let s = xor(sj,k) in
let k22 = T(s, tbx) in
let MSS = xor(MS,k22) in
let HUS=H(Concat(Concat(ID,SID),SR)) in
if HUS = MSS then
let u = xor(ui,k) in
let k11 = T(u, tax) in
let MUU = xor(MU,k11) in
let HSU=H(Concat(Concat(ID,SID),URR)) in
if HSU = MUU then
let HUR=H(Concat(ID,URR)) in
let Ri = xor(HUR,k11) in
let HSR=H(Concat(SID,SR)) in
let Rj = xor(HSR,k22) in
out(ch,(Rj, Ri, ID));
0.
process (user| AS |CA)

4.4.2. Performance Result. The performance result is shown in Figure 4. From the result, we can see that our scheme is secure.

5. Performance

5.1. Computation Cost. According to literature [1, 10, 28–31] and the measured consumption time of the relative algorithms of the proposed scheme on our Intel Core i5-3470 platform, the details are shown as follows.

T_X: XOR. Because XOR operation time is very small, it can be ignored. T_H: hash operation. The hash operation time is 0.6 ms. T_C ($T_n(x) \bmod dP$): Chebyshev chaotic map. Its operation takes twice the time of hash operation. T_{GEN}: the time of obtaining public parameters and feature key from biometric feature by fuzzy extractor algorithm. The time is 805 times that of hash operation. T_{Rep}: the time of regenerating the biometric key from biometric feature and public parameter by fuzzy extractor algorithm. The time is 187 times that of hash operation. T_E and T_D: symmetric encryption operation and symmetric decryption operation. The operations of both of them take twice the time of hash operation. T_{ECC}: encryption or decryption of elliptic curve public key cryptography. The time is 968 times that of hash operation. T_P: the time of an elliptic curve point multiplication. The time is 126 times that of hash operation. T_A: the time of an elliptic curve point addition. The time is 14 times that of hash operation. The comparison results of execution time of the related schemes are shown in Table 5.

As can be seen from Table 5, the computation cost of two phases is the lowest respectively in our proposed scheme. The proposed scheme is superior to the similar scheme in [1, 4, 10].

Table 6: Communication cost comparison.

Scheme	Login, authentication, and key agreement phase (bits)
BbKAS [1]	$6L_M + 15L_{ID} + 8L_T = 2528$
BVbAS [4]	$7L_H + 3L_T = 1504$
PpAP [10]	$4L_H + 6L_P + 4L_T = 3072$
Proposed scheme	$6L_H + 5L_{ID} + 4L_M + 4L_T = 2144$

5.2. Communication Cost. Referring to [1, 4, 10], we set the length as follows. L_{ID}: the length of identity is 32bits; L_H: the length of hash function is 160bits; L_M: the output size of chaotic maps is 128bits; L_T: the length of time is 128bits because it can be considered as a random number; L_E: the length of symmetric encryption/decryption is 128bits; L_P: the output size of an elliptic curve point $P = (P_x, P_y)$ is 320bits; and L_r: the length of random nonce is 128bits.

Here only the often executed login and authentication phases are considered for cost calculations. The comparison results of communication cost for the protocols are presented in Table 6. It can be observed that our scheme is more efficient than the schemes [1, 10] in communication cost.

6. Conclusion

In order to improve security of authentication system and strengthen protection for sensitive information and privacy of users, a provably secure Chebyshev chaotic map (CCM)-based authentication scheme is proposed. The scheme uses hash function to hide user information and uses fuzzy extractor to authenticate user biometric feature. Especially, the proposed scheme transformed the traditional public key of Chebyshev chaotic map into a private key and combined two private keys to compute a one-time key used to encrypt authentication information. The results verified by BAN logic and ProVerif simulation tool confirm that the scheme is well secured against all existing security threats. Compared

with similar schemes, the proposed scheme is more efficient and secure. Therefore, the proposed scheme has great application value in high security demands scenarios such as mobile payment and contactless access control. In the future, we will continue to further study authentication schemes for more complex network environment.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (grant no. 41706201).

References

[1] H. Zhu, X. Hao, and Y. Zhang, "A biometrics-based multi-server key agreement scheme on chaotic maps cryptosystem," *Journal of Information Hiding and Multimedia Signal Processing*, vol. 6, no. 2, pp. 211–224, 2015.

[2] Q. Jiang, F. Wei, S. Fu, J. Ma, G. Li, and A. Alelaiwi, "Robust extended chaotic maps-based three-factor authentication scheme preserving biometric template privacy," *Nonlinear Dynamics*, vol. 83, no. 4, pp. 2085–2101, 2016.

[3] A. Rifaqat and P. A. Kumar, "A secure and robust three-factor based authentication scheme using RSA cryptosystem," *International Journal of Business Data Communications and Networking*, vol. 13, no. 1, pp. 74–84, 2017.

[4] X. Dong, M. Li, and Y. Du, "A biometric verification-based authentication scheme using Chebyshev chaotic mapping," *Journal of Beijing University of Aeronautics and Astronautics*, vol. 45, no. 05, pp. 1052–1058, 2019.

[5] S. Zhou, Q. Gan, and X. Wang, "Authentication scheme based on smart card in multi-server environment," *Wireless Networks*, vol. 26, no. 2, pp. 855–863, 2020.

[6] N. Lwamo, L. Zhu, and C. Xu, "Suaa: a secure user authentication scheme with anonymity for the single & multi-server environments," *Information Sciences*, vol. 477, pp. 369–385, 2019.

[7] N. Kumar, K. Kaur, S. C. Misra, and R. Iqbal, "An intelligent RFID-enabled authentication scheme for healthcare applications in vehicular mobile cloud," *Peer-to-Peer Networking and Applications*, vol. 9, no. 5, pp. 824–840, 2016.

[8] D. He, N. Kumar, and J. H. Lee, "Privacy-preserving data aggregation scheme against internal attackers in smart grids," *Wireless Networks*, vol. 22, no. 2, pp. 491–502, 2015.

[9] S. Roy, A. K. Das, S. Chatterjee, N. Kumar, S. Chattopadhyay, and J. J. P. C. Rodrigues, "Provably secure fine-grained data access control over multiple cloud servers in mobile cloud computing-based healthcare applications," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 457–468, 2019.

[10] D. Rangwani, D. Sadhukhan, S. Ray, M. K. Khan, and M. Dasgupta, "A robust provable-secure privacy-preserving authentication protocol for Industrial Internet of Things," *Peer-to-peer Networking and Applications*, vol. 14, no. 3, pp. 1548–1571, 2021.

[11] Y. Luo, Y. Liu, J. Liu, X. Ouyang, Y. Cao, and X. Ding, "ECM-IBS: a Chebyshev map-based broadcast authentication for wireless sensor networks," *International Journal of Bifurcation and Chaos*, vol. 29, no. 09, p. 1950118, 2019.

[12] X. Y. Guo, D. Z. Sun, and Y. Yang, "An improved three-factor session initiation protocol using Chebyshev chaotic map," *IEEE Access*, vol. 8, pp. 111265–111277, 2020.

[13] R. I. Abdelfatah, N. M. Abdal-Ghafour, and M. E. Nasr, "Secure VANET authentication protocol (SVAP) using Chebyshev chaotic maps for emergency conditions," *IEEE Access*, vol. 10, pp. 1096–1115, 2022.

[14] M. Fueyo and J. Herranz, "On the efficiency of revocation in RSA-based anonymous systems," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1771–1779, 2016.

[15] S. Barman, H. P. H. Shum, S. Chattopadhyay, and D. Samanta, "A secure authentication protocol for multi-server-based E-healthcare using a fuzzy commitment scheme," *IEEE Access*, vol. 7, pp. 12557–12574, 2019.

[16] X. Zhang and J. Liu, "Multi-factor authentication protocol based on hardware fingerprint and biometrics," *Netinfo Security*, vol. 20, no. 8, pp. 9–15, 2020.

[17] A. Aug, "Security of cryptocurrencies in blockchain technology: state-of-art, challenges and future prospects," *Journal of Network and Computer Applications*, vol. 163, p. 765, 2020.

[18] S. Roy, S. Chatterjee, and A. K. Das, "On the design of provably secure lightweight remote user authentication scheme for mobile cloud computing services," *IEEE Access*, p. 1, 2017.

[19] S. Namasudra and P. Roy, "A new secure authentication scheme for cloud computing environment," *Concurrency and Computation: Practice and Experience*, vol. 29, pp. e3864–20, 2017.

[20] Y. Dodis, R. Ostrovsky, L. Reyzin, and A. Smith, "Fuzzy extractors: how to generate strong keys from biometrics and other noisy data," *SIAM Journal on Computing*, vol. 38, no. 1, pp. 97–139, 2008.

[21] D. A. Kumar, "A secure and robust temporal credential-based three-factor authentication scheme for wireless sensor networks," *Peer-to-peer Networking and Applications*, vol. 9, no. 1, pp. 223–244, 2016.

[22] M. Wazid, A. K. Das, S. Kumari, X. Li, and F. Wu, "Design of an efficient and provably secure anonymity preserving three-factor user authentication and key agreement scheme for TMIS," *Security and Communication Networks*, vol. 9, no. 13, pp. 1983–2001, 2016.

[23] R. Zhang, Y. Xiao, S. Sun, and H. Ma, "Efficient multi-factor authenticated key exchange scheme for mobile communications," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 4, pp. 625–634, 2019.

[24] C. Herder, L. Ren, M. Dijk, M. D. Yu, and S. Devadas, "Trapdoor computational fuzzy extractors and stateless cryptographically-secure physical unclonable functions," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 1, pp. 65–82, 2017.

[25] R. Amin, R. S. Sherratt, D. Giri, S. H. Islam, and M. K. Khan, "A software agent enabled biometric security algorithm for secure file access in consumer storage devices," *IEEE Transactions on Consumer Electronics*, vol. 63, no. 1, pp. 53–61, 2017.

[26] L. Zhang, "Cryptanalysis of the public key encryption based on multiple chaotic systems," *Chaos, Solitons & Fractals*, vol. 37, no. 3, pp. 669–674, 2008.
[27] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *Proceedings of the Royal Society of London*, vol. 426, pp. 233–271, 1989.
[28] M. Zhang, J. Zhang, and Y. Zhang, "Remote three-factor authentication scheme based on Fuzzy extractors," *Security and Communication Networks*, vol. 8, no. 4, pp. 682–693, 2015.
[29] V. Odelu, A. K. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1953–1966, 2015.
[30] C. C. Chang and C. Y. Sun, "A secure and efficient authentication scheme for E-coupon systems," *Wireless Personal Communications*, vol. 77, no. 4, pp. 2981–2996, 2014.
[31] C. C. Lee, "A simple key agreement scheme based on chaotic maps for VSAT satellite communications," *International Journal of Satellite Communications and Networking*, vol. 31, no. 4, pp. 177–186, 2013.