Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 3805058, 15 pages
https://doi.org/10.1155/2020/3805058

Research Article

A Robust IoT-Based Three-Factor Authentication Scheme for Cloud Computing Resistant to Session Key Exposure

Feifei Wang[1], **Guosheng Xu**[1,2], **Guoai Xu**[1,2], **Yuejie Wang**[3] and **Junhao Peng**[4]

[1]*Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*National Engineering Laboratory of Mobile Network Security, Beijing 100876, China*
[3]*Peking University, Beijing 100871, China*
[4]*Guangzhou University, Guangzhou 510006, China*

Correspondence should be addressed to Guosheng Xu; guoshengxu@bupt.edu.cn

Received 1 November 2019; Revised 5 January 2020; Accepted 20 January 2020; Published 18 February 2020

Academic Editor: Ghufran Ahmed

With the development of Internet of Things (IoT) technologies, Internet-enabled devices have been widely used in our daily lives. As a new service paradigm, cloud computing aims at solving the resource-constrained problem of Internet-enabled devices. It is playing an increasingly important role in resource sharing. Due to the complexity and openness of wireless networks, the authentication protocol is crucial for secure communication and user privacy protection. In this paper, we discuss the limitations of a recently introduced IoT-based authentication scheme for cloud computing. Furthermore, we present an enhanced three-factor authentication scheme using chaotic maps. The session key is established based on Chebyshev chaotic-based Diffie–Hellman key exchange. In addition, the session key involves a long-term secret. It ensures that our scheme is secure against all the possible session key exposure attacks. Besides, our scheme can effectively update user password locally. Burrows–Abadi–Needham logic proof confirms that our scheme provides mutual authentication and session key agreement. The formal analysis under random oracle model proves the semantic security of our scheme. The informal analysis shows that our scheme is immune to diverse attacks and has desired features such as three-factor secrecy. Finally, the performance comparisons demonstrate that our scheme provides optimal security features with an acceptable computation and communication overheads.

1. Introduction

With the rapid growth of Internet of Things (IoT) technologies, Internet-enabled devices have had a tremendous impact on people's works and lives [1–3]. However, the Internet-enabled devices have limited storage, computing power, and communication ability. To solve this limitation, cloud computing emerged as a new service paradigm [4]. It provides a new method with high efficiency and convenience to realize information and resource sharing. The users are able to access the resources, services, or applications that are deployed in distributed cloud servers by utilizing a handheld device anywhere and anytime. And the control server is in charge of authorizing the users and distributed servers.

As the communication channel is open and unprotected, there are diverse and severe security threats for stealing sensitive data and resource in cloud computing environment [5, 6]. An authentication protocol is indispensable to prevent unauthorized access and protect the sensitive data and user privacy. From the first smart card-based authentication protocol [7] introduced by Yang and Shieh in 1999, there have been a large number of enhanced schemes proposed [2, 8–13]. Based on the authentication factors the user employs, the authentication schemes are divided into two-factor authentication schemes and three-factor authentication schemes. Based on the cryptosystem the authentication scheme adopts, the authentication schemes are divided into hash-based schemes, symmetric cryptosystem-based schemes, and public key cryptosystem-based schemes.

1.1. Related Works. In terms of authentication schemes for cloud computing, some proposals have been presented one after another to improve the security and efficiency [14–17]. In 2015, Tsai and Lo [18] put forward an anonymous authentication protocol using bilinear pairing, in which the user can directly login the distributed server without the help of control server. Afterwards, He et al. [19] revealed that their scheme is not resistant to server impersonation attack and put forward an enhanced scheme. In 2017, Kumari et al. [20] presented a biometric-based authentication protocol employing elliptic curve cryptosystem (ECC). However, their scheme cannot withstand known session-specific temporary information attack and fails to preserve three-factor secrecy. In 2018, Amin et al. [21] pointed out that two anonymous authentication schemes [22, 23] have weaknesses like forgery attack and session key disclosure attack and introduced a hash-based two-factor authentication scheme. Unfortunately, Wang et al. [24] revealed that their scheme still cannot resist session key disclosure attack. In 2019, Mo et al. [25] introduced an ECC-based single-server two-factor authentication protocol. But this protocol is not resistant to stolen-verifier attack. In the same year, Zhou et al. [26] put forward a two-factor authentication scheme employing hash function. But we observe that their scheme suffers from forgery attack and replay attack and does not preserve forward secrecy. For better understanding, we summarize these schemes in Table 1.

Among these schemes, the hash-based schemes [21–23, 26] are highly efficient, but they have diverse vulnerabilities, such as desynchronization attack, forgery attack, and failure to achieve forward secrecy and user anonymity. Wang et al. [27, 28] have demonstrated that public key technique is essential for achieving some security attributes such as user anonymity. However, the existing public key cryptosystem-based schemes [18, 20, 25] still have more or less security vulnerabilities due to design deficiencies. Besides, they have high computation overhead as time-consuming operations such as bilinear pairing and scalar multiplication are involved.

In addition, the security of the session key is a noteworthy issue. The existing schemes are not secure against various session key exposure attacks. A great many schemes such as the schemes in [20, 21, 25, 26] cannot withstand known session-specific temporary information attack. And many schemes such as the schemes in [21–23, 26] cannot provide forward secrecy. Besides, in some schemes like the scheme of Amin et al. [21], the attacker can even reveal the session key when he obtains the smart card [29].

1.2. Motivation and Contributions. The existing schemes suffer from various security defects or involve high computation overhead. The security attributes or the efficiency needs to be improved. In particular, the great majority of schemes fail to guarantee the security of session key, as they are subjected to various session key exposure attacks. It motivates us to present an enhanced authentication scheme that can meet the security requirements at minimum cost and be secure against all the possible session key exposure attacks. We sum up the contributions of the paper as below:

(1) We reveal that Zhou et al.'s scheme [26] does not consider impersonation attack, known session-specific temporary information attack, and forward secrecy.

(2) We put forward an enhanced three-factor authentication scheme using chaotic maps. The session key comprises of a long-term secret value and the secret key generated by Chebyshev chaotic-based Diffie–Hellman key exchange. It can prevent all kinds of session key exposure attacks. The use of Chebyshev chaotic maps contributes to the establishment of secure session key and simultaneously reduces the computation cost.

(3) The Burrows–Abadi–Needham logic proof confirms the completeness of our scheme. The formal analysis under the random oracle model proves the semantic security of session key. And the informal analysis shows that our scheme can resist all kinds of potential attacks and provide desired properties like three-factor secrecy. The performance comparisons demonstrate that our scheme has high security, and its computation and communication overheads are acceptable.

1.3. Organization of the Paper. The rest of the paper is formed as below. We give some background materials in Section 2. We reveal the security defects of Zhou et al.'s scheme in Section 3. We present the enhanced three-factor authentication scheme in Section 4. We discuss the security of our scheme using several widely accepted security analysis methods in Section 5. We present the performance comparisons of our scheme and related schemes in Section 6. The conclusion is given in Section 7.

2. Preliminaries

2.1. Chebyshev Chaotic Maps. According to Zhang [30], the enhanced Chebyshev polynomial is defined as $T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \bmod p$, where $n \geq 2$, $x \in [-\infty, +\infty]$, and p is a big prime. The Chebyshev polynomials satisfy commutative law, i.e., $T_a(T_b(x)) \equiv T_b(T_a(x)) \bmod p$.

There is a hard mathematical problem on Chebyshev polynomials:

(i) Chebyshev chaotic Diffie–Hellman problem (CHDHP): for given $T_a(x)$, $T_b(x)$, and x, the computation of $T_{ab}(x) \equiv T_a(T_b(x)) \equiv T_b(T_a(x)) \bmod p$ is infeasible.

2.2. Adversary Model. Based on [31], the abilities of adversary are summarized as below:

(i) The adversary can eavesdrop, replay, block, or alter the transmitted messages in open channel

Table 1: Related authentication schemes for cloud computing.

Category	Schemes	Authentication factors	Cryptographic primitive	Security limitations
Public key cryptosystem-based schemes	Tsai and Lo [18]	Three-factor	Bilinear pairing	Server impersonation attack
	He et al. [19]	Two-factor	Bilinear pairing	Inefficient typo detection
	Kumari et al. [20]	Three-factor	Elliptic curve cryptosystem	Known session-specific temporary information attack Fails to preserve three-factor secrecy Stolen-verifier attack
	Mo et al. [25]	Two-factor	Elliptic curve cryptosystem	Forgery attack Replay attack Known session-specific temporary information attack
Hash-based schemes	Amin et al. [21]	Two-factor	Hash function	Off-line guessing attack Session key disclosure attack Fails to preserve forward secrecy Known session-specific temporary information attack
	Xue et al. [22]	Two-factor	Hash function	User anonymity Privileged insider attack Off-line password guessing attack Fails to preserve forward secrecy
	Chuang and Chen [23]	Three-factor	Hash function	User impersonation attack Session key discloser attack Fails to preserve forward secrecy
	Zhou et al. [26]	Two-factor	Hash function	Forgery attack Replay attack Fails to preserve forward secrecy Known session-specific temporary information attack

(ii) When testing forward secrecy, the adversary can obtain control server's master key or cloud server's secret key

(iii) The adversary can disclose the password or the parameters of smart card

(iv) When testing three-factor secrecy, the adversary is capable of obtaining any two kinds of authentication factors

2.3. Notations. The notations of the paper are presented in Table 2.

Table 2: Notations.

Symbols	Description
CS	Control server
ID_{CS}	Identity of control server
s	Master key of CS
y	Secret value of CS
U_i	User
ID_i, PW_i, b_i	Identity, password, biometric of user
PID_i	Pseudoidentity of user
S_j	Cloud server
SID_j	Identity of cloud server
sm_j	Secret key of cloud server
T_1, T_2, T_3, T_4	Timestamps
SK	Session key
E_k/D_k	Symmetric encryption/decryption algorithm with key k
$\|$	The string concatenation operation
$\oplus$	The bitwise XOR operation
H_1	Hash function
H_2	Biohash function, on the basis of user's biometric B_i, along with a tokenized random number, it outputs a random string

3. Cryptanalysis of Zhou et al.'s Scheme

We briefly review Zhou et al.'s scheme [26] and point out its limitations in this section. In their scheme, the attacker can perform impersonation attack by replaying the intercepted message. Besides, their scheme is vulnerable to two kinds of session key exposure attacks.

3.1. Review of Zhou et al.'s Scheme

3.1.1. User Registration Phase. U_i delivers an enrollment request to CS in this phase.

Step 1: U_i picks his identity ID_i, pseudoidentity PID_i, and password PW_i. Then U_i selects a random number r_i and calculates $P_i = H_1(PW_i \| r_i)$. Afterwards, U_i delivers the registration request $\{ID_i, PID_i\}$ to CS via the secure channel.

Step 2: after getting $\{ID_i, PID_i\}$, CS checks if ID_i is valid. If it holds, CS computes $A_i = H_1(PID_i \| ID_{CS} \| s)$ and $B_i = H_1(ID_i \| s)$. CS saves ID_i in its database and returns $\{A_i, B_i, ID_{CS}\}$ to U_i via the secure channel.

Step 3: after receiving $\{A_i, B_i, \mathrm{ID}_{\mathrm{CS}}\}$, U_i computes $C_i = A_i \oplus P_i$, $D_i = B_i \oplus H_1(\mathrm{ID}_i \| P_i)$, and $E_i = r_i \oplus H_1(\mathrm{ID}_i \| \mathrm{PW}_i)$. Then, U_i stores $\langle C_i, D_i, E_i, \mathrm{PID}_i, \mathrm{ID}_{\mathrm{CS}}\rangle$ in the memory of a smart card.

3.1.2. Cloud Server Registration Phase. CS distributes the secret key to S_j in this phase.

Step 1: S_j chooses its identity SID_j and pseudoidentity PSID_j and delivers $\{\mathrm{SID}_j, \mathrm{PSID}_j\}$ to CS via the secure channel.

Step 2: after getting $\{\mathrm{SID}_j, \mathrm{PSID}_j\}$, CS computes $sm_1 = H_1(\mathrm{PSID}_j \| \mathrm{ID}_{\mathrm{CS}} \| s)$ and $sm_2 = H_1(\mathrm{SID}_j \| s)$. CS delivers $\{sm_1, sm_2, \mathrm{ID}_{\mathrm{CS}}\}$ to S_j via the secure channel.

Step 3: S_j keeps $\{sm_1, sm_2, \mathrm{ID}_{\mathrm{CS}}\}$ as secret.

3.1.3. Login and Authentication Phase. U_i and S_j authenticate each other in the assistance of CS as shown in Figure 1.

Step 1: U_i inputs ID_i^* and PW_i^*. The smart card picks a new pseudoidentity $\mathrm{PID}_i^{\mathrm{new}}$, computes $r_i^* = E_i \oplus H_1(\mathrm{ID}_i^* \| \mathrm{PW}_i^*)$, $P_i^* = H_1(\mathrm{PW}_i^* \| r_i^*)$, $A_i^* = C_i \oplus P_i^*$, $B_i^* = D_i \oplus H_1(\mathrm{ID}_i \| P_i^*)$, $f_1 = A_i^* \oplus \alpha$, $f_2 = H_1(\alpha \| \mathrm{PID}_i \| \mathrm{ID}_{\mathrm{CS}}) \oplus \mathrm{ID}_i$, $f_3 = B_i^* \oplus \mathrm{PID}_i^{\mathrm{new}} \oplus H_1(\alpha \| \mathrm{ID}_i)$, and $f_4 = H_1(\mathrm{ID}_i \| \mathrm{PID}_i \| \mathrm{PID}_i^{\mathrm{new}} \| \alpha \| f_3)$, where α is a nonce. The smart card delivers the login request $\{\mathrm{PID}_i, f_1, f_2, f_3, f_4\}$ to S_j via the public channel.

Step 2: upon receiving $\{\mathrm{PID}_i, f_1, f_2, f_3, f_4\}$, S_j picks a new pseudoidentity $\mathrm{PSID}_j^{\mathrm{new}}$. S_j computes $f_5 = sm_1 \oplus \beta$, $f_6 = H_1(\beta \| \mathrm{PSID}_j \| \mathrm{ID}_{\mathrm{CS}}) \oplus \mathrm{SID}_j$, $f_7 = sm_2 \oplus \mathrm{PSID}_j^{\mathrm{new}} \oplus H_1(\beta \| \mathrm{SID}_j)$, and $f_8 = H_1(\mathrm{SID}_j \| \mathrm{PSID}_j \| \mathrm{PSID}_j^{\mathrm{new}} \| \beta \| f_7)$, where β is a random number. S_j sends $\{\mathrm{PID}_i, f_1, f_2, f_3, f_4, \mathrm{PSID}_j, f_5, f_6, f_7, f_8\}$ to CS via the public channel.

Step 3: after receiving $\{\mathrm{PID}_i, f_1, f_2, f_3, f_4, \mathrm{PSID}_j, f_5, f_6, f_7, f_8\}$, CS computes $\alpha = f_1 \oplus H_1(\mathrm{PID}_i \| \mathrm{ID}_{\mathrm{CS}} \| s)$, $\mathrm{ID}_i = f_2 \oplus H_1(\alpha \| \mathrm{PID}_i \| \mathrm{ID}_{\mathrm{CS}})$, $\mathrm{PID}_i^{\mathrm{new}} = f_3 \oplus H_1(\mathrm{ID}_i \| s) \oplus H_1(\alpha \| \mathrm{ID}_i)$, and $f_4' = H_1(\mathrm{ID}_i \| \mathrm{PID}_i \| \mathrm{PID}_i^{\mathrm{new}} \| \alpha \| f_3)$, and verifies if $f_4' = f_4$. If the equation holds, CS computes $\beta = f_5 \oplus H_1(\mathrm{PSID}_j \| \mathrm{ID}_{\mathrm{CS}} \| s)$, $\mathrm{SID}_j = f_6 \oplus H_1(\beta \| \mathrm{PSID}_j \| \mathrm{ID}_{\mathrm{CS}})$, $\mathrm{PSID}_j^{\mathrm{new}} = f_7 \oplus H_1(\mathrm{SID}_j \| s) \oplus H_1(\beta \| \mathrm{SID}_j)$, and $f_8' = H_1(\mathrm{SID}_j \| \mathrm{PSID}_j \| \mathrm{PSID}_j^{\mathrm{new}} \| \beta \| f_7)$, and verifies if $f_8' = f_8$. If it holds, proceed next step, otherwise, the protocol aborts.

Step 4: CS computes $\mathrm{SK} = H_1(\alpha \oplus \beta \oplus \gamma)$, $f_9 = H_1(\mathrm{PSID}_j^{\mathrm{new}} \| \mathrm{ID}_{\mathrm{CS}} \| s) \oplus H_1(\beta \| \mathrm{PSID}_j^{\mathrm{new}})$, $f_{10} = H_1(\mathrm{PSID}_j^{\mathrm{new}} \| \beta \| \mathrm{PSID}_j) \oplus (\alpha \oplus \gamma)$, $f_{11} = H_1(\mathrm{SK} \| f_9 \| f_{10} \| H_1(\mathrm{SID}_j \| s))$, $f_{12} = H_1(\mathrm{PID}_i^{\mathrm{new}} \| \mathrm{ID}_{\mathrm{CS}} \| s) H_1(\alpha \| \mathrm{PID}_i^{\mathrm{new}})$, $f_{13} = H_1(\mathrm{PID}_i^{\mathrm{new}} \| \alpha \| \mathrm{PID}_i) \oplus (\beta \oplus \gamma)$, and $f_{14} = H_1(\mathrm{SK} \| f_{12} \| f_{13} \| H_1(\mathrm{ID}_i \| s))$, where γ is a nonce. CS sends $\{f_9, f_{10}, f_{11}, f_{12}, f_{13}, f_{14}\}$ to S_j.

Step 5: after receiving $\{f_9, f_{10}, f_{11}, f_{12}, f_{13}, f_{14}\}$, S_j computes $(\alpha \oplus \gamma) = f_{10} \oplus H_1(\mathrm{PSID}_j^{\mathrm{new}} \| \beta \| \mathrm{PSID}_j)$, $\mathrm{SK} = H_1(\alpha \oplus \gamma \oplus \beta)$, $f_{11}' = H_1(\mathrm{SK} \| f_9 \| f_{10} \| sm_2)$ and verifies if $f_{11}' = f_{11}$. If it holds, S_j computes $sm_1^{\mathrm{new}} = f_9 \oplus H_1(\beta \| \mathrm{PSID}_j^{\mathrm{new}})$. S_j keeps $sm_1^{\mathrm{new}}, \mathrm{PSID}_j^{\mathrm{new}}$ as secret and removes sm_1, PSID_j. S_j delivers $\{f_{12}, f_{13}, f_{14}\}$ to U_i.

Step 6: after receiving $\{f_{12}, f_{13}, f_{14}\}$, the smart card computes $(\beta \oplus \gamma) = f_{13} \oplus H_1(\mathrm{PID}_i^{\mathrm{new}} \| \alpha \| \mathrm{PID}_i)$, $\mathrm{SK} = H_1(\alpha \oplus \beta \oplus \gamma)$, $f_{14}' = H_1(\mathrm{SK} \| f_{12} \| f_{13} \| B_i^*)$ and checks if $f_{14}' = f_{14}$. If they are equal, the smart card calculates $C_i^{\mathrm{new}} = f_{12} \oplus H_1(\alpha \| \mathrm{PID}_i^{\mathrm{new}}) \oplus P_i$, stores $C_i^{\mathrm{new}}, \mathrm{PID}_i^{\mathrm{new}}$, and removes C_i, PID_i.

3.2. Cryptanalysis of Zhou et al.'s Scheme. In this section, we reveal that Zhou et al.'s scheme suffers from replay attack, user impersonation attack, server impersonation attack, and known session-specific temporary information attack and fails to provide forward secrecy.

3.2.1. Forward Secrecy. Forward secrecy ensures that when the long-term secret is compromised, the attacker still cannot reveal the established session key. In Zhou et al.'s scheme, with the master key s, the attacker can reveal SK as follows:

Step 1: the attacker intercepts $\{\mathrm{PID}_i, f_1, f_2, f_3, f_4\}$ and $\{f_{12}, f_{13}, f_{14}\}$ from the public channel

Step 2: the attacker computes $\alpha = f_1 \oplus H_1(\mathrm{PID}_i \| \mathrm{ID}_{\mathrm{CS}} \| s)$, $\mathrm{ID}_i = f_2 \oplus H_1(\alpha \| \mathrm{PID}_i \| \mathrm{ID}_{\mathrm{CS}})$, $\mathrm{PID}_i^{\mathrm{new}} = f_3 \oplus H_1(\mathrm{ID}_i \| s) \oplus H_1(\alpha \| \mathrm{ID}_i)$, $(\beta \oplus \gamma) = f_{13} \oplus H_1(\mathrm{PID}_i^{\mathrm{new}} \| \alpha \| \mathrm{PID}_i)$, and $\mathrm{SK} = H_1(\alpha \oplus \beta \oplus \gamma)$

When the long-term secret s is compromised, all the established session keys will be disclosed.

3.2.2. User Impersonation Attack. User impersonation attack denotes that the attacker can masquerade as a valid user to login the cloud server. This attack is performed as follows:

Step 1: the attacker intercepts $\{\mathrm{PID}_i, f_1, f_2, f_3, f_4\}$ from the public channel and sends this message to S_j.

Step 2: upon receiving $\{\mathrm{PID}_i, f_1, f_2, f_3, f_4\}$, S_j handles this message and sends $\{\mathrm{PID}_i, f_1, f_2, f_3, f_4, \mathrm{PSID}_j^{\mathrm{new}}, f_5^*, f_6^*, f_7^*, f_8^*\}$ to CS.

Step 3: CS handles the message $\{\mathrm{PID}_i, f_1, f_2, f_3, f_4, \mathrm{PSID}_j^{\mathrm{new}}, f_5^*, f_6^*, f_7^*, f_8^*\}$. As $f_4' = f_4$, $f_8' = f_8^*$, CS sends back $\{f_9^*, f_{10}^*, f_{11}^*, f_{12}^*, f_{13}^*, f_{14}^*\}$ to S_j.

Step 4: S_j handles $\{f_9^*, f_{10}^*, f_{11}^*, f_{12}^*, f_{13}^*, f_{14}^*\}$. As $f_{11}' = f_{11}^*$, S_j sends $\{f_{12}^*, f_{13}^*, f_{14}^*\}$ to the attacker.

S_j and CS believe that $\{\mathrm{PID}_i, f_1, f_2, f_3, f_4\}$ comes from the legitimate user U_i. Without compromising the smart card, the attacker can impersonate the user U_i by replaying $\{\mathrm{PID}_i, f_1, f_2, f_3, f_4\}$. Zhou et al.'s scheme is vulnerable to user impersonation attack.

3.2.3. Server Impersonation Attack. Server impersonation attack denotes that the attacker can masquerade as a valid cloud server to deceive the user. This attack is performed as follows:

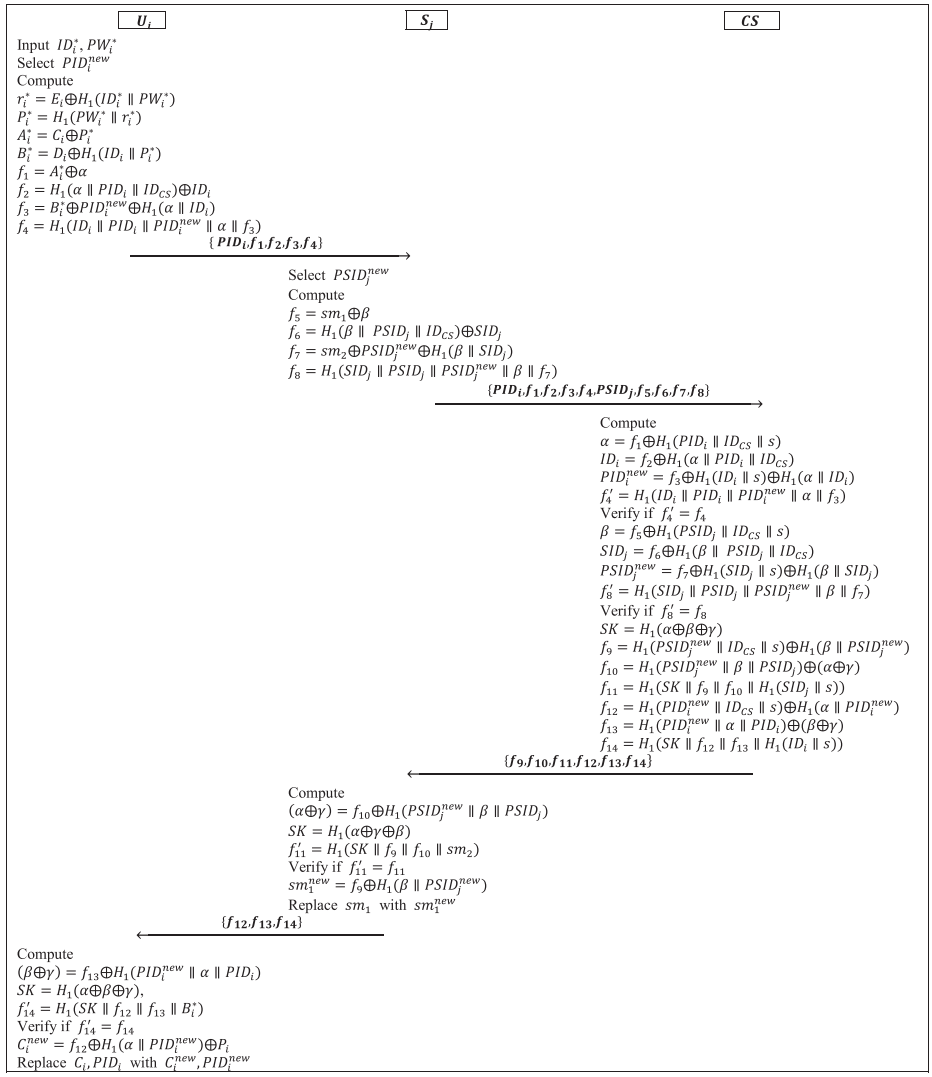

Figure 1: Login and authentication phase of Zhou et al.'s scheme.

Step 1: the attacker intercepts the message $\{\text{PID}_i, f_1, f_2, f_3, f_4, \text{PSID}_j, f_5, f_6, f_7, f_8\}$ from the public channel.

Step 2: when intercepting a new login request $\{\text{PID}_i^{\text{new}}, f_1^*, f_2^*, f_3^*, f_4^*\}$ from the public channel, the attacker sends $\{\text{PID}_i^{\text{new}}, f_1^*, f_2^*, f_3^*, f_4^*, \text{PSID}_j, f_5, f_6, f_7, f_8\}$ to CS.

Step 3: CS handles $\{\text{PID}_i^{\text{new}}, f_1^*, f_2^*, f_3^*, f_4^*, \text{PSID}_j, f_5, f_6, f_7, f_8\}$. As $f_4' = f_4^*$, $f_8' = f_8$, CS returns $\{f_9^*, f_{10}^*, f_{11}^*, f_{12}^*, f_{13}^*, f_{14}^*\}$ to the attacker.

Step 4: the attacker delivers $\{f_{12}^*, f_{13}^*, f_{14}^*\}$ to U_i.

Step 5: after receiving $\{f_{12}^*, f_{13}^*, f_{14}^*\}$, as $f_{14}' = f_{14}^*$, U_i believes that the attacker is the legitimate cloud server S_j.

In $\{\text{PID}_i, f_1, f_2, f_3, f_4, \text{PSID}_j, f_5, f_6, f_7, f_8\}$ that S_j delivers to CS, $\{\text{PSID}_j, f_5, f_6, f_7, f_8\}$ is completely independent with $\{\text{PID}_i, f_1, f_2, f_3, f_4\}$. CS verifies the validity of the two messages independently. It leads that the attacker can impersonate the cloud server by replaying $\{\text{PSID}_j, f_5, f_6, f_7, f_8\}$.

3.2.4. Known Session-Specific Temporary Information Attack. This attack denotes that when the temporary secret such as random number is compromised, the attacker can reveal the established session key. With the random number α, the attacker reveals the session key as follows:

Step 1: the attacker intercepts $\{\text{PID}_i, f_1, f_2, f_3, f_4\}$ and $\{f_{12}, f_{13}, f_{14}\}$ from the public channel.

Step 2: when U_i generates a new login request using $\text{PID}_i^{\text{new}}$ and sends $\{\text{PID}_i^{\text{new}}, f_1^*, f_2^*, f_3^*, f_4^*\}$ to S_j. The attacker intercepts $\{\text{PID}_i^{\text{new}}, f_1^*, f_2^*, f_3^*, f_4^*\}$ from the public channel.

Step 3: the attacker computes $(\beta \oplus \gamma) = f_{13} \oplus H_1(\text{PID}_i^{\text{new}} \| \alpha \| \text{PID}_i)$ and $\text{SK} = H_1(\alpha \oplus \beta \oplus \gamma)$.

In Zhou et al.'s scheme, if the random number α is compromised, the attacker can reveal the session key. Zhou et al.'s scheme is vulnerable to known session-specific temporary information attack.

4. The Proposed Scheme

A robust three-factor authentication scheme for cloud computing is put forward in this section. The proposed scheme is described as below.

4.1. System Setup Phase. CS picks its master key s. CS also selects a nonce y as its secret value. CS picks a hash function $H_1()$ and a symmetric cryptosystem $E_k()/D_k()$. In addition, CS publishes the Chebyshev polynomial's parameters x, p.

4.2. User Registration Phase. U_i transmits the enrollment request to CS in this phase, as shown in Figure 2.

Step 1: U_i selects his identity ID_i and password PW_i as he wishes, imprints his biometric b_i, and calculates $A_i = H_1(\mathrm{PW}_i \| H_2(b_i))$. Then, U_i delivers $\{\mathrm{ID}_i, A_i\}$ to CS through the secure channel.

Step 2: after receiving $\{\mathrm{ID}_i, A_i\}$, CS picks two random numbers t_i, r_1, calculates $C_i = H_1(s \| \mathrm{ID}_i)$, $D_i = A_i \oplus C_i$, $Z_i = H_1(A_i \oplus \mathrm{ID}_i) \bmod \mu$, and $\mathrm{PID}_i = E_y(\mathrm{ID}_i \| r_1)$, where the integer μ satisfies $2^8 \le \mu \le 2^{10}$. CS saves $\{\mathrm{ID}_i, t_i, \mathrm{Counter} = 0\}$ into its database. Moreover, CS stores parameters $\langle D_i, Z_i, \mathrm{PID}_i, t_i, H_1(y \| r_1), \mu \rangle$ in a smart card and delivers it to U_i.

4.3. Cloud Server Registration Phase. CS issues the secret key to S_j in this phase.

Step 1: S_j delivers its identity SID_j to CS through the secure channel.

Step 2: after receiving $\{\mathrm{SID}_j\}$, CS calculates $sm_j = H_1(\mathrm{SID}_j \| s)$. CS sends back $\{sm_j\}$ to S_j through the secure channel.

Step 3: S_j keeps sm_j as secret.

4.4. Login and Authentication Phase. U_i and S_j perform mutual authentication by the aide of CS in this phase, as shown in Figure 3.

Step 1: U_i inputs his identity ID_i^* and password PW_i^* and imprints the biometric b_i^*. Then, the smart card calculates $A_i^* = H_1(\mathrm{PW}_i^* \| H_2(b_i^*))$ and $Z_i^* = H_1(A_i^* \oplus \mathrm{ID}_i^*) \bmod \mu$ and verifies whether Z_i^* is equal to Z_i. If it holds, the smart card calculates $O_i = t_i \oplus H_1(T_1 \| H_1(y \| r_1))$, $C_i^* = D_i \oplus A_i^*$, $R_i = T_\alpha(x)$, and $L_i = H_1(\mathrm{ID}_i^* \| C_i^* \| R_i \| \mathrm{SID}_j \| O_i \| T_1)$, where α is a nonce and T_1 is the current timestamp. $\{\mathrm{PID}_i, R_i, L_i, O_i, T_1\}$ is delivered to S_j through the open channel.

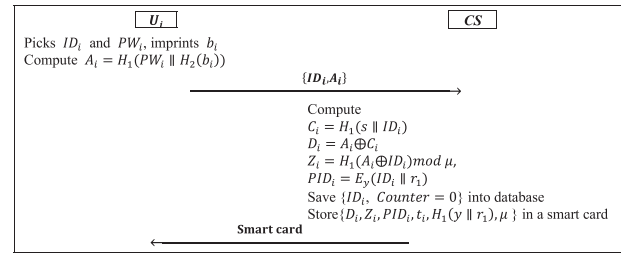

Figure 2: User registration phase.

Step 2: after getting $\{\mathrm{PID}_i, R_i, L_i, O_i, T_1\}$, S_j verifies whether T_1 is fresh. If it holds, S_j generates a random number β and calculates $R_S = T_\beta(x)$, $G_i = H_1(\mathrm{PID}_i \| R_i \| R_S \| sm_j \| T_2)$, $Q_i = H_1(sm_j \| R_i)$, and $N_i = E_{Q_i}(R_S \| G_i)$, where T_2 is the current timestamp. S_j sends $\{\mathrm{PID}_i, R_i, L_i, O_i, T_1, N_i, T_2\}$ to CS via the public channel.

Step 3: after receiving $\{\mathrm{PID}_i, R_i, L_i, O_i, T_1, N_i, T_2\}$, CS checks the freshness of T_2 and computes $(\mathrm{ID}_i \| r_1') = D_y(\mathrm{PID}_i)$, $t_i' = O_i \oplus H_1(T_1 \| H_1(y \| r_1'))$, $C_i = H_1(s \| \mathrm{ID}_i)$, and $L_i' = H_1(\mathrm{ID}_i \| C_i \| R_i \| \mathrm{SID}_j \| O_i \| T_1)$ and verifies $t_i' \stackrel{?}{=} t_i$, $L_i' \stackrel{?}{=} L_i$. If $t_i' = t_i$, $L_i' \ne L_i$, the smart card is probably compromised. For the item $\{\mathrm{ID}_i,\ t_i,\ \mathrm{Counter}\}$, CS performs Counter = Counter + 1. When it reaches the preset value, CS suspends U_i. If $t_i' = t_i$, $L_i' = L_i$, CS believes the authenticity of U_i and performs the next step.

Step 4: CS computes $sm_j = H_1(\mathrm{SID}_j \| s)$, $Q_i = H_1(sm_j \| R_i)$, $(R_S' \| G_i') = D_{Q_i}(N_i)$, and $G_i'' = H_1(\mathrm{PID}_i \| R_i \| R_S' \| sm_j \| T_2)$, and verifies $G_i' \oplus G_i''$. If it holds, CS believes the authenticity of S_j. Otherwise, the protocol terminates.

Step 5: CS picks a nonce r_2, computes $\mathrm{PID}_i^{\mathrm{new}} = E_y(\mathrm{ID}_i \| r_2)$, $K_i = H_1(C_i \| R_i)$, $F_i = E_{K_i}(Q_i \| R_S' \| \mathrm{PID}_i^{\mathrm{new}})$, and $M_1 = H_1(sm_j \| R_i \| F_i \| T_3)$, where T_3 is the current timestamp. CS transmits $\{F_i, M_1, T_3\}$ to S_j through the open channel.

Step 6: after getting $\{F_i, M_1, T_3\}$, S_j checks the freshness of T_3 and computes $M_1' = H_1(sm_j \| R_i \| F_i \| T_3)$, and verifies $M_1' \stackrel{?}{=} M_1$. If they are equal, S_j authenticates the user U_i successfully. S_j computes $E_i = T_\beta(R_i)$, $\mathrm{SK} = H_1(E_i \| Q_i)$, and $M_2 = H_1(\mathrm{SK} \| F_i \| Q_i \| T_4)$. S_j transmits $\{F_i, M_2, T_4\}$ to U_i via the public channel.

Step 7: upon receiving $\{F_i, M_2, T_4\}$, the smart card checks the freshness of T_4. Then, the smart card computes $K_i = H_1(C_i^* \| R_i)$, $(Q_i' \| R_S'' \| \mathrm{PID}_i^{\mathrm{new}}) = D_{K_i}(F_i)$, $E_i = T_\alpha(R_S'')$, $\mathrm{SK} = H_1(E_i \| Q_i')$, and $M_2' = H_1(\mathrm{SK} \| F_i \| Q_i' \| T_4)$, and checks $M_2' \stackrel{?}{=} M_2$. If it holds, U_i authenticates the cloud server S_j successfully. The smart card replaces PID_i with $\mathrm{PID}_i^{\mathrm{new}}$ in the memory.

4.5. Password Update Phase. The original password is replaced by a new password in this phase.

Step 1: U_i enters ID_i^* and PW_i^* and imprints b_i^*. The smart card calculates $A_i^* = H_1(\mathrm{PW}_i^* \| H_2(b_i^*))$ and $Z_i^* = H_1(A_i^* \oplus \mathrm{ID}_i^*) \bmod \mu$, and verifies whether Z_i^* is equal to Z_i. If they are not equal, the protocol terminates.

Step 2: U_i keys a new password $\mathrm{PW}_i^{\mathrm{new}}$. The smart card computes $A_i^{\mathrm{new}} = H_1(\mathrm{PW}_i^{\mathrm{new}} \| H_2(b_i^*))$, $Z_i^{\mathrm{new}} =$

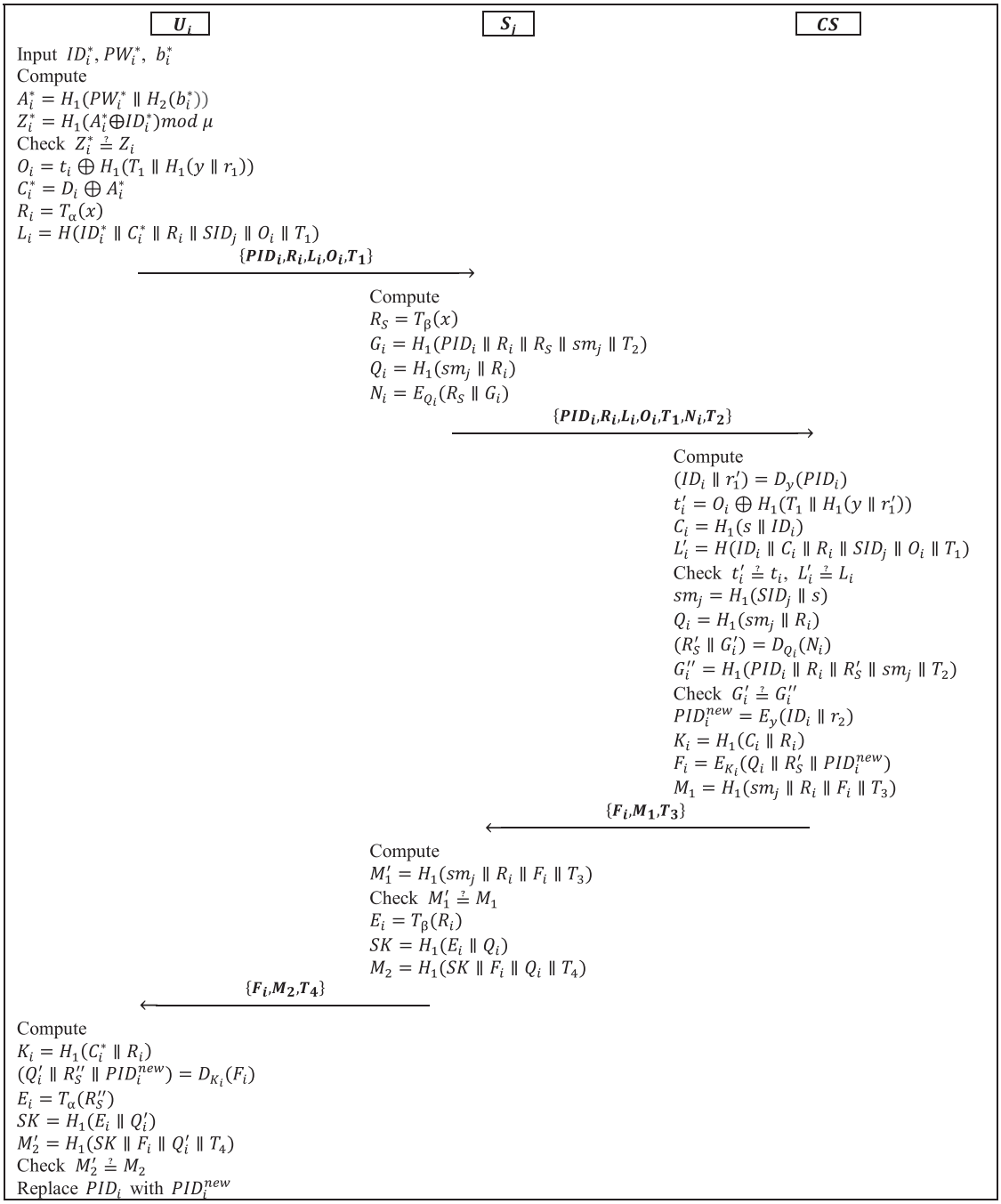

Figure 3: Login and authentication phase of the proposed scheme.

$H_1(A_i^{\text{new}} \oplus \text{ID}_i^*) \text{mod}\, \mu$, and $D_i^{\text{new}} = D_i \oplus A_i^* \oplus A_i^{\text{new}}$. The smart card replaces Z_i, D_i, with Z_i^{new}, D_i^{new}.

5. Security Analysis

In this section, Burrows–Abadi–Needham (BAN) logic [32] proof demonstrates the completeness of our scheme. The formal analysis under the random oracle model shows that our scheme provides semantic security. Moreover, the informal analysis proves that our scheme is not susceptible to known attacks.

5.1. BAN Logic Proof. We confirm the correctness of our scheme in this section. Table 3 lists the notations and rules of BAN logic.

Our scheme ought to fulfil the goals as below.

Goal 1: $S_j| \equiv U_i| \equiv (U_i \overset{\text{SK}}{\longleftrightarrow} S_j)$

Goal 2: $S_j| \equiv (U_i \overset{\text{SK}}{\longleftrightarrow} S_j)$

Goal 3: $U_i| \equiv S_j| \equiv (U_i \overset{\text{SK}}{\longleftrightarrow} S_j)$

Goal 4: $U_i| \equiv (U_i \overset{\text{SK}}{\longleftrightarrow} S_j)$

We idealize our scheme as below:

Table 3: The notations and rules in BAN logic.

Symbols	Description
P, Q	The principals
X	A statement
$P \triangleleft X$	P see X, P gets a message that consists of X
$P\mid\sim X$	P said X, P sent a message that consists of X
$P\mid\equiv X$	P is convinced that X is true
$P \Longrightarrow X$	P has jurisdiction over X
$P \overset{K}{\longleftrightarrow} Q$	K is a secret shared by P and Q
$\langle X \rangle_K$	X is combined with a secret K
$\{X\}_K$	X is encrypted with a key K
$\#(X)$	X is fresh
Belief rule	$(P\mid\equiv Q\mid\equiv (X,Y)/P\mid\equiv Q\mid\equiv X)$
Message meaning rule (Rule 1)	$(P\mid\equiv P \overset{K}{\longleftrightarrow} Q, P\triangleleft\langle X\rangle_K/P\mid\equiv Q\mid\sim X)$ or $(P\mid\equiv P \overset{K}{\longleftrightarrow} Q, P\triangleleft\{X\}_K/P\mid\equiv Q\mid\sim X)$
Nonce-verification rule (Rule 2)	$(P\mid\equiv \#(X), P\mid\equiv Q\mid\sim X/P\mid\equiv Q\mid\equiv X)$
Jurisdiction rule (Rule 3)	$(P\mid\equiv Q \Longrightarrow X, P\mid\equiv Q\mid\equiv X/P\mid\equiv X)$

M1: $U_i \longrightarrow S_j\langle \mathrm{ID}_i, U_i\mid\equiv R_i, T_1\rangle_{C_i}$

M2a: $S_j \longrightarrow \mathrm{CS}\langle \mathrm{ID}_i, U_i\mid\equiv R_i, T_1\rangle_{C_i}$

M2b: $S_j \longrightarrow \mathrm{CS}\left\{\langle \mathrm{PID}_i, R_i, R_S, T_2\rangle_{sm_j}, R_s\right\}_{Q_i}$

M3: $\mathrm{CS} \longrightarrow S_j\left\{U_i \overset{Q_i}{\longleftrightarrow} S_j, R_S, \mathrm{PID}_i^{\mathrm{new}}, R_i\right\}_{K_i}, \langle U_i\mid\equiv R_i, F_i, T_3\rangle_{sm_j}$

M4a: $S_j \longrightarrow U_i\left\{U_i \overset{Q_i}{\longleftrightarrow} S_j, R_S, \mathrm{PID}_i^{\mathrm{new}}, R_i\right\}_{K_i}$

M4b: $S_j \longrightarrow U_i\langle U_i \overset{\mathrm{SK}}{\longleftrightarrow} S_j, F_i, T_4\rangle_{Q_i}$

The analysis of our scheme is based on the following initial assumptions:

A1: $\mathrm{CS}\mid\equiv U_i \overset{C_i}{\longleftrightarrow} \mathrm{CS}$

A2: $\mathrm{CS}\mid\equiv \#(T_1)$

A3: $\mathrm{CS}\mid\equiv U_i \overset{sm_j}{\Rightarrow} U_i\mid\equiv R_i$

A4: $S_j\mid\equiv \mathrm{CS} \overset{sm_j}{\longleftrightarrow} S_j$

A5: $S_j\mid\equiv \#(T_3)$

A6: $S_j\mid\equiv \mathrm{CS}\Rightarrow U_i\mid\equiv R_i$

A7: $S_j\mid\equiv U_i \Rightarrow U_i \overset{\mathrm{SK}}{\longleftrightarrow} S_j$

A8: $U_i\equiv U_i \overset{K_i}{\longleftrightarrow} \mathrm{CS}$

A9: $U_i\mid\equiv \#(R_i)$

A10: $U_i\mid\equiv \mathrm{CS}\Rightarrow U_i \overset{Q_i}{\longleftrightarrow} S_j$

A11: $U_i\mid\equiv \#(T_4)$

A12: $U_i\mid\equiv S_j \Rightarrow U_i \overset{\mathrm{SK}}{\longleftrightarrow} S_j$

The analysis of the proposed scheme is as follows. According to M2a, we get

(1) $\mathrm{CS}\triangleleft\langle \mathrm{ID}_i, U_i\mid\equiv R_i, T_1\rangle_{C_i}$
From (1), A1, applying Rule 1, we get

(2) $\mathrm{CS}\mid\equiv U_i\mid\sim\langle \mathrm{ID}_i, U_i\mid\equiv R_i, T_1\rangle$
From (2), A2, applying Rule 2, we get

(3) $\mathrm{CS}\equiv U_i\mid\equiv (U_i\equiv R_i)$
From (3), A3, applying Rule 3, we get

(4) $\mathrm{CS}\mid\equiv U_i\mid\equiv R_i$
According to M3, we get

(5) $S_j\triangleleft\langle U_i\mid\equiv R_i, F_i, T_3\rangle_{sm_j}$
From (5), A4, applying Rule 1, we get

(6) $S_j\mid\equiv \mathrm{CS}\mid\sim\langle U_i\mid\equiv R_i, F_i, T_3\rangle$
From (6), A5, applying Rule 2, we get

(7) $S_j\mid\equiv \mathrm{CS}\mid\equiv (U_i\mid\equiv R_i)$
From (7), A6, applying Rule 3, we get

(8) $S_j\mid\equiv U_i\mid\equiv R_i$
From (8), and $\mathrm{SK} = H_1(T_\beta(R_i)\|H_1(sm_j\|R_i))$, we get

(9) $S_j\mid\equiv U_i\mid\equiv U_i \overset{\mathrm{SK}}{\longleftrightarrow} S_j$ **Goal 1**
From (9), A7, applying Rule 3, we get

(10) $S_j\mid\equiv U_i \overset{\mathrm{SK}}{\longleftrightarrow} S_j$ **Goal 2**
According to M4a, we get

(11) $U_i\triangleleft\left\{U_i \overset{Q_i}{\longleftrightarrow} S_j, R_S, \mathrm{PID}_{\mathrm{new}}, R_i\right\}_{K_i}$
From (11), A8, applying Rule 1, we get

(12) $U_i\mid\equiv \mathrm{CS}\mid\sim\left\{U_i \overset{Q_i}{\longleftrightarrow} S_j, R_S, \mathrm{PID}_{\mathrm{new}}, R_i\right\}$
From (12), A9, applying Rule 2, we get

(13) $U_i\mid\equiv \mathrm{CS}\mid\equiv (U_i \overset{Q_i}{\longleftrightarrow} S_j)$
From (13), A10, applying Rule 3, we get

(14) $U_i\mid\equiv U_i \overset{Q_i}{\longleftrightarrow} S_j$
According to M4b, we get

(15) $U_i\triangleleft\langle U_i \overset{\mathrm{SK}}{\longleftrightarrow} S_j, F_i, T_4\rangle_{Q_i}$
From (14), (15), applying Rule 1, we get

(16) $U_i\mid\equiv S_j\mid\sim\langle U_i \overset{\mathrm{SK}}{\longleftrightarrow} S_j, F_i, T_4\rangle$
From (16), A11, applying Rule 2, we get

(17) $U_i\mid\equiv S_j\mid\equiv U_i \overset{\mathrm{SK}}{\longleftrightarrow} S_j$ **Goal 3**
From (17), A12, applying Rule 3, we get

(18) $U_i\mid\equiv U_i \overset{\mathrm{SK}}{\longleftrightarrow} S_j$ **Goal 4**

5.2. Formal Security Analysis. Based on the security model of two-factor authentication presented by Wang and Wang [33], we put forward a security model of three-factor authentication for cloud computing. Afterwards, we prove the semantic security of our scheme in this model.

5.2.1. Formal Security Model

(1) Participants. There are multiple instances of the control server CS, the cloud server S_j, and the user U_i in the authentication scheme for cloud computing. We use CS^a, S_j^a, and U_i^a to denote these instances.

(2) Queries. The attacker is capable of making the queries as follows:

Execute $(CS^a/S_j^a n/U_i^a)$: by making this query, the attacker can obtain the messages delivered via the open channel.

Send $(CS^a/S_j^a n/U_i^a h, xm)$: by making this query, the attacker can impersonate the principal (U_i^a, S_j^a, GWN^a) to send a message m. If m is valid, a response is sent back to the attacker.

Reveal (S_j^a/U_i^a): by making this query, the attacker can get the session key of (S_j^a/U_i^a), if the principal involves a session key.

Corrupt (U_i^a, τ): by making this query, the attacker is capable of getting one or two types of user authentication information.

When $\tau = 1$, the attacker acquires the password.
When $\tau = 2$, the attacker acquires the smart card.
When $\tau = 3$, the attacker acquires the biometric.

Corrupt (S_j^a/CS^a): by making this query, the attacker can obtain cloud server's secret key or CS's master key. This oracle corresponds to the forward secrecy.

Test (S_j^a/U_i^a): if the principal is fresh (see below) and involves a session key SK, the oracle spins a coin b. When $b = 1$, it sends back SK to the attacker. When $b = 0$, it sends back a random string to the attacker. This oracle is used to simulate the semantic security of session key. The attacker is capable of asking this query only once.

(3) Freshness. We say (S_j^a/U_i^a) is fresh, if the following conditions are met:

(1) (S_j^a/U_i^a) is accepted and involves a SK
(2) The attacker never makes Corrupt (S_j^a/CS^a) or Reveal (S_j^a/U_i^a) query

(4) Semantic Security. After making the above queries, the attacker tries to reveal the value of b in test query. The advantage of the attacker in breaking the semantic security is defined as follows:

$$Adv_P^{ake}(\mathscr{A}) = 2\Pr(b' = b) - 1. \tag{1}$$

If for all the attackers, $Adv_P^{ake}(\mathscr{A})$ is negligible, the authentication scheme provides semantic security.

5.2.2. Formal Security Analysis

Theorem 1. *Let the password space D_{PW} be subject to Zipf distribution [34]. A polynomial-time attacker $\mathscr{A}$ runs against our scheme. We presume $\mathscr{A}$ can ask less than q_e Execute queries, q_s Send queries, q_b Biohash queries, q_h Hash queries, and q_ε Encryption/Decryption queries. We have*

$$Adv_P^{ake}(\mathscr{A}) \le 2C' * {q_s}^{s'} + \frac{(q_s + q_e)^2}{p} + \frac{6q_s + q_h^2}{2^{l_1}} + \frac{2q_s + q_b^2}{2^{l_2}} + \frac{q_\varepsilon^2}{2^{l_3}} + 2q_h Adv_P^{\mathrm{CHDHP}}, \tag{2}$$

where l_1, l_2, l_3 are the length of hash output, bio-hash output, and symmetric encryption output, respectively. Adv_P^{CHDHP} is the advantage of $\mathscr{A}$ in solving CHDHP. When using the Tianya password distribution [34], we have $|D_{PW}| \approx 13$ million, $C' = 0.062239$, and $s' = 0.155478$.

Proof. In order to obtain $Adv_P^{ake}(\mathscr{A})$, we define the games Φ_i $(0 \le i \le 6)$, where Φ_0 corresponds to the real attack. $\Pr[\chi_i]$ is the advantage of $\mathscr{A}$ in revealing b in game G_i.

Φ_0: as it simulates the real attack, we get,

$$Adv_P^{ake}(\mathscr{A}) = 2(\Pr[\chi_0]) - 1. \tag{3}$$

Φ_1: in this game, a hash list Λ_H is used to simulate the hash oracle. A biohash list Λ_{BH} is used to simulate the biohash oracle. And an encryption/decryption list Λ_ε is used to simulate the encryption/decryption oracle. For a hash query $H_1(\alpha)$, if the hash value of α already exists in Λ_H, the oracle sends back the hash value. Otherwise, the oracle selects a nonce β as the answer of $H_1(\alpha)$ and stores (α, β) in Λ_H. The biohash oracle is performed in the similar way. For an encryption query $E_k(\varphi)$, the oracle firstly uses φ and k to search Λ_ε, If there exists an tuple (k, φ, ω), it answers ω. Otherwise, it sends back a random string ω to the adversary and stores (k, φ, ω) in Λ_ε. For an decryption query $D_k(\omega)$, the oracle uses ω and k to search Λ_ε. If there exists a tuple (k, φ, ω), it answers φ. Otherwise, it sends back a random string φ to the adversary, and stores (k, φ, ω) in Λ_ε. Φ_1 is indistinguishable from Φ_0. We get

$$\Pr[\chi_0] - \Pr[\chi_1] = 0. \tag{4}$$

Φ_2: in this game, we terminate the execution when encountering some collisions.

(1) The collision occurs on the outputs of hash function or biohash function with the probability of $(q_h^2/2^{l_1+1}) + (q_b^2/2^{l_2+1})$
(2) The collision occurs on the outputs of symmetric encryption with the probability of $(q_\varepsilon^2/2^{l_3+1})$
(3) The collision occurs on the transcripts of messages, with the probability of $((q_s + q_e)^2/2p)$

We get

$$\left|\Pr[\chi_1] - \Pr[\chi_2]\right| \le \frac{q_h^2}{2^{l_1+1}} + \frac{q_b^2}{2^{l_2+1}} + \frac{q_\varepsilon^2}{2^{l_3+1}} + \frac{(q_s + q_e)^2}{2p}. \tag{5}$$

Φ_3: in this game, we terminate the execution when $\mathscr{A}$ guesses $\{L_i, G_i, M_1, M_2\}$. The probability is at most $(q_s/2^{l_1})$. We get

$$\left|\Pr[\chi_3] - \Pr[\chi_2]\right| \leq \frac{q_s}{2^{l_1}}. \quad (6)$$

Φ_4: in this game, we terminate the execution when $\mathscr{A}$ guesses user's authentication value C_i. The probability is less than $(q_s/2^{l_1})$. We obtain

$$\left|\Pr[\chi_4] - \Pr[\chi_3]\right| \leq \frac{q_s}{2^{l_1}}. \quad (7)$$

Φ_5: in this game, we terminate the execution when $\mathscr{A}$ has computed C_i with the help of Corrupt (U_i^a, z).

(1) When Corrupt $(U_i^a, z = 1, 2)$, $\mathscr{A}$ is able to guess the biometric with the probability of $(q_s/2^{l_1})$
(2) When Corrupt $(U_i^a, z = 2, 3)$, $\mathscr{A}$ is able to guess the password with the probability of $C' * {q_s}^{s'}$.
(3) When Corrupt $(U_i^a, z = 1, 3)$, $\mathscr{A}$ is able to guess D_i with the probability of $(q_s/2^{l_1})$.

We obtain

$$\left|\Pr[\chi_5] - \Pr[\chi_4]\right| \leq \frac{q_s}{2^{l_1}} + C' * {q_s}^{s'} + \frac{q_s}{2^{l_1}}. \quad (8)$$

Φ_6: we use the private hash oracle H_1' rather than the hash oracle H_1 to compute the session key. $\mathscr{A}$ knows nothing about H_1'. Thus, we have

$$\Pr[\chi_6] = \frac{1}{2}. \quad (9)$$

Φ_6: it is indistinguishable from Φ_5, unless a hash query $H_1(E_i \| Q_i)$ is made by $\mathscr{A}$. We use Γ_1 to denote this event. We have

$$\left|\Pr[\chi_6] - \Pr[\chi_5]\right| \leq \Pr[\Gamma_1]. \quad (10)$$

□

If the hash query $H_1(E_i \| Q_i)$ has been asked, by selecting randomly in Λ_H, we can obtain $E_i = T_\alpha(R_S) = T_\beta(R_i)$ with the probability of $(1/q_h)$. We get

$$\Pr[\Gamma_1] \leq q_h Adv_P^{\mathrm{CHDHP}}. \quad (11)$$

From (3)–(11), we have

$$Adv_P^{ake}(\mathscr{A}) \leq 2C' * {q_s}^{s'} + \frac{(q_s + q_e)^2}{p} + \frac{6q_s + q_h^2}{2^{l_1}} + \frac{2q_s + q_b^2}{2^{l_2}} + \frac{q_\varepsilon^2}{2^{l_3}} + 2q_h Adv_P^{\mathrm{CHDHP}}. \quad (12)$$

5.3. Informal Security Analysis. In this section, we prove that our scheme is resistant to diverse attacks. Particularly, our scheme is secure against all kinds of session key exposure attacks, as the session key is generated based on the long-term secret and Chebyshev chaotic-based Diffie–Hellman key exchange. Besides, we demonstrate that the proposed scheme preserves desired properties such as user anonymity and three-factor secrecy.

5.3.1. User Anonymity. In our scheme, only the control server who has the secret key y can retrieve ID_i from PID_i. In addition, after authenticating U_i, CS generates a new pseudoidentity $\mathrm{PID}_i^{\mathrm{new}}$ and delivers the encrypted $\mathrm{PID}_i^{\mathrm{new}}$ to U_i. $\mathrm{PID}_i^{\mathrm{new}}$ is encrypted with the secret key K_i by CS. Only U_i is able to compute K_i and obtain $\mathrm{PID}_i^{\mathrm{new}}$. The attacker knows nothing about $\mathrm{PID}_i^{\mathrm{new}}$, and hence he cannot link $\mathrm{PID}_i^{\mathrm{new}}$ with PID_i. It makes the user identity untraceable. Consequently, our scheme achieves user anonymity.

5.3.2. Resistance to Off-Line Guessing Attack. As the fuzzy verifier Z_i is employed in our scheme as suggested in [33], even if the attacker obtains the smart card as well as biometric at the same time, he is unable to reveal the password. With the smart card and biometric, the attacker chooses one pair of identity and password from dictionary space and checks if $Z_i^* = Z_i$. However, there are a great many candidates conforming to $Z_i^* = Z_i$. In order to distinguish the correct one from so many candidates, there is no alternative but to launch online guessing attack. However, we employ the "honeywords" technique [33] to prevent this attack. When the number of online guessing attacks reaches the preset value, for example, 10, U_i is suspended. Consequently, our scheme can resist off-line guessing attack.

5.3.3. Resistance to Session Key Disclosure Attack. The session key SK is computed using E_i and Q_i. E_i is the secret key generated by the Chebyshev chaotic-based Diffie–Hellman key exchange. Only U_i and S_j who know the random number α or β are able to compute E_i. Q_i is computed based on the long-term secret sm_j. Only S_j and CS who have sm_j are able to compute Q_i. Besides, Q_i is transmitted to U_i by means of symmetric encryption with the secret key $H_1(C_i \| R_i)$. Only U_i and CS who have C_i are able to reveal Q_i. Both E_i and Q_i are unavailable to the attacker. Therefore, our scheme can resist this attack.

5.3.4. Forward Secrecy. Suppose that the attacker has acquired the master key s, he is able to compute Q_i. However, E_i is the secret key generated by the Chebyshev chaotic-based Diffie–Hellman key exchange. To reveal E_i, there is no alternative but to solve the CHDHP. Therefore, our scheme preserves forward secrecy.

5.3.5. Resistance to Session-Specific Temporary Information Attack. Assume that the attacker has acquired the random number α. To compute $E_i = T_\alpha(R_s)$, R_s is required. R_S is encrypted using the secret key Q_i or K_i, where $Q_i = H_1(sm_j \| R_i)$ and $K_i = H_1(C_i \| R_i)$. To retrieve R_S, the attacker needs to reveal C_i or sm_j. Assume that the attacker has acquired the random number β, the attacker can calculate $E_i = T_\beta(R_i)$. Afterwards, to derive Q_i, the attacker has to get C_i or sm_j. However, sm_j is only known to S_j and CS, C_i is only known to U_i and CS. Therefore, the attacker cannot reveal the session key when the nonce is disclosed.

5.3.6. Resistance to Forgery Attack. In our scheme, the hash values (L_i, G_i, M_1, M_2) of the transmitted parameters and

the secret value A_i, sm_j are used to ensure message integrity and verify the sender's identity. As C_i and sm_j are unavailable to the attacker, he cannot generate a message that is verified to be valid by the recipient.

5.3.7. Resistance to Desynchronization Attack. In each message, the hash value (L_i, G_i, M_1, M_2) is used to ensure that the transmitted parameters are not tampered with. If the attacker alters a parameter of a message, the receiver will find that the received hash value is not equal to the one he computes, and the protocol terminates. Besides, if the attacker blocks a message, as it does not change the long-term parameters the participants have, it does not affect the user's next login. For instance, if the attacker blocks the message $\{F_i, M_2, T_4\}$, the user fails to update his pseudoidentity. But with PID_i, the user still is able to access the cloud server.

5.3.8. Resistance to Replay Attack. In the proposed scheme, every message contains a timestamp. And the timestamps are involved in the hash values (L_i, G_i, M_1, M_2). Upon receiving a message, the receiver first verifies whether the timestamp is fresh. If it holds, the receiver continues to process the message. Otherwise, the protocol aborts.

5.3.9. Resistance to Privileged Insider Attack. The user never submits his biometric or password to CS at registration. On the other hand, the user cannot masquerade as a cloud server or CS, as sm_j is unavailable. The cloud server cannot masquerade as a user or CS, as C_i is unavailable. Therefore, our scheme is immune to such an attack.

5.3.10. Resistance to Man-in-the-Middle Attack. In each message, the hash value (L_i, G_i, M_1, M_2) of the transmitted parameters and the secret A_i, sm_j are computed to ensure message integrity and verify the sender's identity. As A_i and sm_j are unavailable, the attacker is unable to generate a valid message to replace the intercepted one. Consequently, the attacker is unable to launch man-in-the-middle attack.

5.3.11. Mutual Authentication. In our scheme, based on the authentication value C_i, CS verifies the authenticity of U_i by checking $L_i' \stackrel{?}{=} L_i$. Based on the secret key sm_j, CS verifies the authenticity of S_j by checking $G_i' \stackrel{?}{=} G_i''$. If $L_i' = L_i$ and $G_i' = G_i''$, CS believes that U_i and S_j are legitimate and sends a response message to S_j. After getting the response message from CS, S_j verifies the authenticity of CS and U_i by checking $M_1' \stackrel{?}{=} M_1$. If it holds, S_j believes that U_i and CS are legitimate and sends back a response message to U_i. Afterwards, U_i verifies the authenticity of CS and S_j by checking $M_2' \stackrel{?}{=} M_2$. If it holds, U_i believes that CS and S_j are legitimate. Therefore, our scheme provides mutual authentication among CS, U_i, and S_j.

5.3.12. Resistance to Eavesdropping Attack. The attacker can intercept messages from public channel. However, the secret parameters such as the user authentication value C_i, the user identity, the cloud server's secret key sm_j, and the session key SK are protected with hash function and symmetric encryption. The attacker cannot acquire any useful information from the intercepted messages and uses them to launch active attacks.

5.3.13. Resistance to All Kinds of Session Key Exposure. The session key consists of two parts, E_i and Q_i. E_i is generated by Chebyshev chaotic-based Diffie–Hellman key exchange. Q_i is computed using the cloud server's secret key as well as a Chebyshev polynomial. The purpose of E_i is to make sure that our scheme can be resistant to known key attack, as well as preserve forward secrecy. The purpose of Q_i is to make sure that our scheme can withstand session-specific temporary information attack. The attacker can reveal neither E_i nor Q_i. Hence, our scheme is resistant to all kinds of session key exposure attacks.

5.3.14. Three-Factor Secrecy. When the attacker reveals the biometric and smart card, he cannot retrieve the password as shown in 5.3.2. When the attacker reveals the smart card and password, he cannot retrieve the biometric from Z_i as the hash function is irreversible. With the biometric and password, the attacker cannot retrieve the critical data of smart card. Hence, our scheme preserves three-factor secrecy.

Most of the existing three-factor authentication schemes fail to preserve three-factory secrecy, because when the biometric and smart card is disclosed, the attacker can guess user's password based on the verification value that is used to verify the validity of the inputted password and biometric in smart card. However, our scheme employs the fuzzy verifier and "honeywords" technique to prevent revealing of the password.

6. Performance Comparisons

We give the comparisons of our scheme and the recently proposed schemes [20, 21, 25, 26, 35, 36] with regard to security attributes, communication, and computation costs in this section.

The security comparison is given in Table 4. Note that, as Mo et al.'s scheme is designed for single-server environment, it only involves a single cloud server. Ghani et al.'s scheme does not establish session key. We summarize the security requirements of authentication protocol, and based on it, we analyzes the security properties of related schemes in Table 4. It indicates that only the proposed scheme meets all the security requirements, while the related schemes have diverse weaknesses. The security of our scheme is superior to the hash function-based schemes [21, 26], the symmetric cryptosystem-based schemes [35, 36], as well as ECC-based schemes [20, 25].

In Table 5, we presents the computation cost, the communication overhead, and the smart card storage cost of the related schemes concerning the login and authentication phase. Furthermore, the computation cost comparison, communication overhead comparison, and smart card

Table 4: Security features of related schemes.

Security properties	Zhou et al. [26]	Amin et al. [21]	Kumari et al. [20]	Mo et al. [25]	Ghani et al. [35]	Martinez-Pelaez et al. [36]	Our scheme
User anonymity	√	√	√	√	×	√	√
Efficient typo detection	×	√	√	√	×	√	√
Resist desynchronization attack	√	√	√	√	×	×	√
Resist off-line guessing attack	√	×	√	√	√	×	√
Resist session key disclosure attack	√	×	√	√	−	×	√
Resist forgery attack	×	×	√	×	×	×	√
Resist replay attack	×	√	√	√	×	√	√
Resist known session-specific temporary information attack	×	×	×	×	−	×	√
Forward secrecy	×	×	√	√	−	×	√
Three-factor secrecy	−	−	×	−	−	−	√
Stolen-verifier attack	√	√	√	×	×	√	√
Resist all kinds of session key exposure attacks	×	×	×	×	−	×	√

Table 5: Computation and communication costs and smart card storage cost of related schemes.

	Computation overhead			Running time (ms)	Communication cost (bits)	Storage cost (bits)
	User	Server	CS			
Zhou et al. [26]	$10T_H$	$7T_H$	$19T_H$	18	3584	640
Amin et al. [21]	$9T_H$	$4T_H$	$10T_H$	11.5	2560	512
Kumari et al. [20]	$2T_{BH} + 5T_H + 3T_P$	$5T_H + 3T_P$	$6T_H + 2T_P$	554.64	2624	576
Mo et al. [25]	$7T_H + 1T_A + 3T_P$	$6T_H + 1T_A + 3T_P$	−	385.998	960	712
Ghani et al. [35]	$2T_H$	$2T_H$	$4T_H + 2T_s$	21.4	1792	384
Martinez-Pelaez et al. [36]	$7T_H + 3T_s$	$6T_H + 3T_s$	$26T_H + 2T_s$	89.1	3456	640
Our scheme	$1T_{BH} + 2T_C + 7T_H + 1T_s$	$5T_H + 2T_C + 1T_s$	$9T_H + 3T_s$	159.1	2304	532

storage cost comparison are shown in Figure 4, Figure 5, and Figure 6, respectively. As shown in Figure 4, the computation overhead of our scheme is not as good as the hash-based schemes and the symmetric cryptosystem-based schemes, as public key operations are used to guarantee the security. But it is obviously better than the ECC-based schemes. As shown in Figure 5, the communication cost of our scheme is higher than Mo et al.'s scheme and Ghani et al.'s scheme, but it is lower than the other schemes. As shown in Figure 6, the storage cost of our scheme is inferior to Amin et al.'s scheme and Ghani et al.'s scheme, but it is superior to the other schemes.

T_H, T_{BH}, T_S, and T_C denote a hash function, a biohash function, a symmetric encryption/decryption, and a Chebyshev polynomial, respectively. T_P and T_A denote a point multiplication and a point addition on elliptic curve group. The computing time of lightweight operation "XOR" is negligible. In accordance with [3, 37], when performed on a smart phone with a Hisilicon kirin 960 CPU, 6 GB RAM, and the storage of 64 GB, the computations of T_H, T_S, T_P, T_A, T_C, and T_{BH} take 0.5 ms, 8.7 ms, 63.075 ms, 0.262 ms, 21.02 ms, and 21.02 ms, respectively. Besides, we assume that the bit length of timestamp, random number, the user identity, the identity of cloud server, the hash value, Chebyshev polynomial, and the output of symmetric encryption are 128 bits. The point on elliptic curve group is 160 bits.

The hash-based schemes and the symmetric cryptosystem-based schemes have obvious advantage in efficiency as they just involve lightweight cryptographic operations. However, they suffer from various security vulnerabilities. In Zhou et al.'s scheme, the attacker is capable of impersonating the user and the cloud server by replaying the intercepted message. In Amin et al.'s scheme, the attacker can retrieve user's password, disclose the session key, and impersonate the user when smart card is compromised. Martinez-Pelaez et al.'s scheme and Ghani et al.'s scheme are vulnerable to diverse and serious security weaknesses. They are unable to provide the essential security protection.

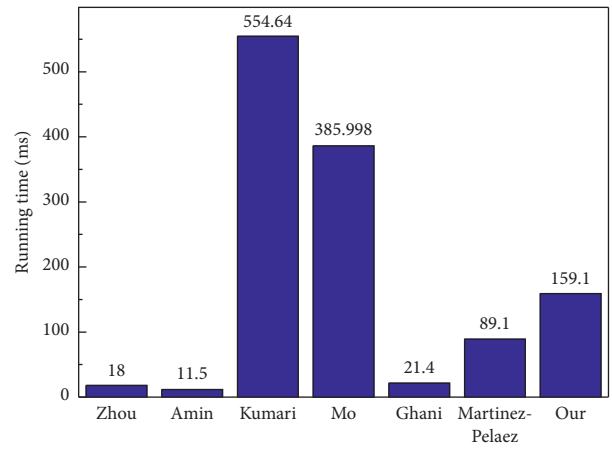

Figure 4: The computation cost of related schemes.

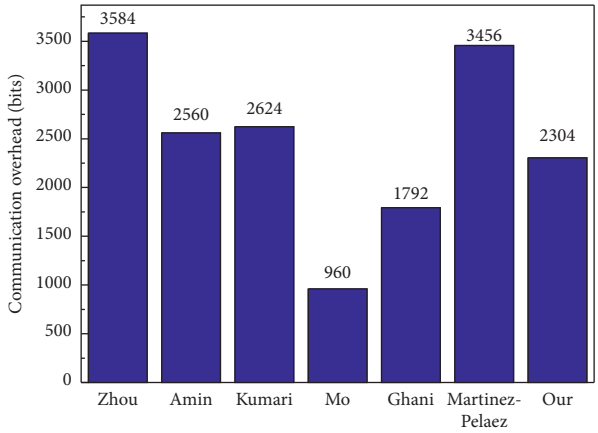

Figure 5: The communication cost of related schemes.

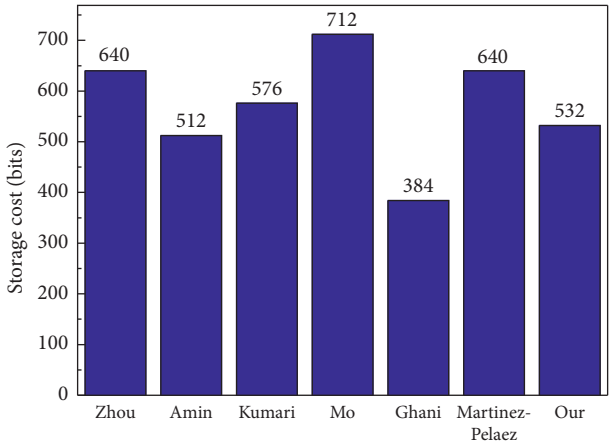

Figure 6: The smart card storage cost of related schemes.

The ECC-based schemes have low efficiency. They have better security than the hash-based schemes and the symmetric cryptosystem-based schemes, but still have security flaws. In Mo et al.'s scheme, the attacker can impersonate the user when the verifier table is leaked. Kumari et al.'s scheme achieves many security features, but it does not provide three-factor secrecy.

In terms of the security of session key, only our scheme is resistant to all kinds of session key exposure attacks, as Chebyshev chaotic-based Diffie–Hellman key exchange and the long-term secret are used to establish the session key. None of the related schemes can withstand known session-specific temporary information attack. If one communication end uses unsecure random number generator, it will lead to the disclosure of the session key. The hash-based schemes and the symmetric cryptosystem-based schemes are unable to provide forward secrecy. The ECC-based schemes provide forward secrecy, as the elliptic curve-based Diffie–Hellman key exchange is employed. Furthermore, in Amin et al.'s scheme and Martinez-Pelaez et al.'s scheme, the attacker can retrieve the session key when the smart card is compromised.

To sum up, the security of our scheme is optimal. In addition, its computation and communication overheads are obviously lower than the ECC-based schemes. Hence, our scheme is more practical.

7. Conclusion

In this paper, we pointed out that Zhou et al.'s scheme is unable to provide the essential security protection for cloud computing, as it does not consider replay attack, known session-specific temporary information attack, and forward secrecy. Furthermore, we present a novel IoT-based three-factor authentication scheme for cloud computing using chaotic maps. The use of Chebyshev chaotic maps guarantees the security of session key and simultaneously reduces the computation cost. In addition, the BAN logic analysis demonstrates that our scheme achieves mutual authentication as well as session key negotiation. The formal analysis confirms the semantic security of session key. The informal analysis proves that our scheme can withstand known attacks and achieve desired attributes such as user anonymity and resistance to all kinds of session key exposure attacks. Finally, the performance comparisons show that our scheme has significant advantage compared with the related schemes. As our scheme has high security, it is especially applicable to the security-critical cloud applications such as cloud-based healthcare systems. Afterwards, on the basis of our current work, we plan to make further study on the authentication protocol for smart healthcare systems.

Data Availability

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

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was funded by the National Key Research and Development Program of China under Grant no. 2018YFB0803600, the National Natural Science Foundation of China under Grant no. 61831003, and the National Natural Science Foundation of China under Grant no. 61873069.

References

[1] W. Ding, X. Jing, Z. Yan, and L. T. Yang, "A survey on data fusion in Internet of Things: towards secure and privacy-preserving fusion," *Information Fusion*, vol. 51, pp. 129–144, 2019.

[2] Q. Jiang, Y. Qian, J. Ma, X. Ma, Q. Cheng, and F. Wei, "User-centric three-factor authentication protocol for cloud-assisted wearable devices," *International Journal Communication Systems*, vol. 32, no. 6, p. e3900, 2019.

[3] S. Roy, S. Chatterjee, A. K. Das, S. Chattopadhyay, S. Kumari, and M. Jo, "Chaotic map-based anonymous user authentication scheme with user biometrics and fuzzy extractor for crowdsourcing internet of things," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2884–2895, 2018.

[4] J. Tang, Y. Cui, Q. Li, K. Ren, J. Liu, and R. Buyya, "Ensuring security and privacy preservation for cloud data services," *ACM Computing Surveys*, vol. 49, no. 1, pp. 1–39, 2016.

[5] W. Meng, W. H. Lee, S. R. Murali, and S. P. T. Krishnan, "Charging me and I know your secrets!: towards juice filming attacks on smartphones," in *Proceedings of the 2015 ACM Workshop on Cyber-Physical System Security*, ACM, Singapore, April 2015.

[6] P. Gope and A. K. Das, "Robust anonymous mutual authentication scheme for," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1764–1772, 2017.

[7] W. H. Yang and S. P. Shieh, "Password authentication schemes with smart cards," *Computers & Security*, vol. 18, no. 8, pp. 727–733, 1999.

[8] F. Wu, L. Xu, and X. Li, "A new chaotic map-based authentication and key agreement scheme with user anonymity for multi-server environment," in *Proceedings of the International Conference on Frontiers Computing 2017*, vol. 464, pp. 335–344, Osaka, Japan, July 2017.

[9] Q. Jiang, J. Ma, C. Yang, X. Ma, J. Shen, and S. A. Chaudhry, "Efficient end-to-end authentication protocol for wearable health monitoring systems," *Computers & Electrical Engineering*, vol. 63, pp. 182–195, 2017.

[10] A. Ostad-Sharif, H. Arshad, M. Nikooghadam, and D. Abbasinezhad-Mood, "Three party secure data transmission in IoT networks through design of a lightweight authenticated key agreement scheme," *Future Generation Computer Systems*, vol. 100, pp. 882–892, 2019.

[11] X. Huang, X. Chen, J. Li, Y. Xiang, and L. Xu, "Further observations on smart-card-based password-authenticated key agreement in distributed systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1767–1775, 2014.

[12] 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.

[13] F. Wang, G. Xu, C. Wang, and J. Peng, "A provably secure biometrics-based authentication scheme for multiserver environment," *Security and Communication Networks*, vol. 2019, Article ID 2838615, 15 pages, 2019.

[14] V. Odelu, A. K. Das, S. Kumari, X. Huang, and M. Wazid, "Provably secure authenticated key agreement scheme for distributed mobile cloud computing services," *Future Generation Computer Systems*, vol. 68, pp. 74–88, 2017.

[15] P. Chandrakar and H. Om, "A secure and robust anonymous three-factor remote user authentication scheme for multi-server environment using ECC," *Computer Communications*, vol. 110, pp. 26–34, 2017.

[16] C. Wang, K. Ding, B. Li et al., "An enhanced user authentication protocol based on elliptic curve cryptosystem in cloud computing environment," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 3048697, 13 pages, 2018.

[17] M. R. Ogiela and L. Ogiela, "Expert knowledge-based authentication protocols for cloud computing applications," in *Proceedings of the 10th International Conference on Intelligent Networking and Collaborative Systems*, vol. 23, pp. 82–27, Oita, Japan, September 2019.

[18] J. L. Tsai and N. W. Lo, "A privacy-aware authentication scheme for distributed mobile cloud computing services," *IEEE Systems Journal*, vol. 9, no. 3, pp. 805–815, 2015.

[19] D. He, N. Kumar, M. K. Khan, L. Wang, and J. Shen, "Efficient privacy-aware authentication scheme for mobile cloud computing services," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1621–1631, 2018.

[20] S. Kumari, X. Li, F. Wu, A. K. Das, K. R. Choo, and J. Shen, "Design of a provably secure biometrics-based multi-cloud-server authentication scheme," *Future Generation Computer Systems*, vol. 68, pp. 320–330, 2017.

[21] R. Amin, N. Kumar, G. P. Biswas, R. Iqbal, and V. Chang, "A lightweight authentication protocol for IoT-enabled devices in distributed cloud computing environment," *Future Generation Computer Systems*, vol. 78, pp. 1005–1019, 2018.

[22] K. Xue, P. Hong, and C. Ma, "A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture," *Journal of Computer and System Sciences*, vol. 80, no. 1, pp. 195–206, 2014.

[23] M. C. Chuang and M. C. Chen, "An anonymous multi-server authenticated key agreement scheme based on trust computing using smartcards and biometrics," *Expert Systems with Applications*, vol. 41, no. 4, pp. 1411–1418, 2014.

[24] P. Wang, B. Li, H. Shi, Y. Shen, and D. Wang, "Revisiting anonymous two-factor authentication schemes for IoT-enabled devices in cloud computing environments," *Security and Communication Networks*, vol. 2019, Article ID 2516963, 13 pages, 2019.

[25] J. Mo, Z. Hu, H. Chen, and W. Shen, "An efficient and provably secure anonymous user authentication and key agreement for mobile cloud computing," *Wireless Communications and Mobile Computing*, vol. 2019, Article ID 4520685, 12 pages, 2019.

[26] L. Zhou, X. Li, K. Yeh, C. Su, and W. Chiu, "Lightweight IoT-based authentication scheme in cloud computing circumstance," *Future Generation Computer Systems*, vol. 91, pp. 244–251, 2019.

[27] D. Wang, D. He, P. Wang, and C. 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.

[28] D. Wang and P. Wang, "On the anonymity of two-factor authentication schemes for wireless sensor networks: attacks, principle and solutions," *Computer Networks*, vol. 73, pp. 41–57, 2014.

[29] S. Chatterjee, S. Roy, A. K. Das, S. Chattopadhyay, N. Kumar, and A. V. Vasilakos, "Secure biometric-based authentication scheme using Chebyshev chaotic map for multi-server environment," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp. 824–839, 2018.

[30] L. Zhang, "Cryptanalysis of the public key encryption based on multiple chaotic systems," *Chaos Solitons & Fractals*, vol. 37, no. 3, pp. 669–674, 2008.

[31] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[32] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Tansactions on Computer Systems*, vol. 8, no. 1, pp. 18–36, 1990.

[33] D. Wang and P. Wang, "Two birds with one stone: two-factor authentication with security beyond conventional bound," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 708–722, 2018.

[34] D. Wang, H. Cheng, P. Wang, X. Huang, and G. Jian, "Zipf's law in passwords," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2776–2791, 2017.

[35] A. Ghani, K. Mansoor, S. Mehmood, S. A. Chaudhry, A. U. Rahman, and M. N. Saqib, "Security and key

management in IoT-based wireless sensor networks: an authentication protocol using symmetric key," *International Journal of Communication Systems*, vol. 32, no. 16, p. e4139, 2019.

[36] R. Martínez-Peláez, H. Toral-Cruz, J. Parra-Michel et al., "An enhanced lightweight IoT-based authentication scheme in cloud computing circumstances," *Sensors*, vol. 19, no. 9, p. 2098, 2019.

[37] D. Abbasinezhad-Mood and M. Nikooghadam, "Efficient anonymous password-authenticated key exchange protocol to read isolated smart meters by utilization of extended Chebyshev chaotic maps," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4815–4828, 2018.