Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 8848032, 12 pages
https://doi.org/10.1155/2021/8848032

WILEY | Hindawi

Research Article

Provably Secure ECC-Based Three-Factor Authentication Scheme for Mobile Cloud Computing with Offline Registration Centre

Hongwei Luo,[1,2] **Feifei Wang**,[3] **and Guoai Xu**[1,2]

[1]*Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*National Engineering Laboratory of Mobile Network Security, Beijing 100876, China*
[3]*Chongqing University of Posts and Telecommunications, Chongqing 400065, China*

Correspondence should be addressed to Guoai Xu; xga@bupt.edu.cn

Received 6 August 2020; Revised 22 December 2020; Accepted 9 May 2021; Published 29 May 2021

Academic Editor: Weizhi Meng

Mobile cloud computing (MCC) aims at solving the resource constrain problem of smart mobile devices. It has deeply affected the way modern humans live and work. In MCC, the authentication scheme is indispensable to prevent illegal attacks and privacy breaches. In this paper, we reveal that a recently proposed two-factor authentication scheme for MCC has limitations like stolen-verifier attack and denial of service attack. In addition, its single-server architecture is not applicable to MCC. To enhance the security, we present a provably secure three-factor authentication scheme using the elliptic curve cryptosystem (ECC). It has the merit that the user only needs to register once to access multiple servers with a pair of public and private key, and the registration center is offline in the authentication phase. Security analysis demonstrates that our scheme is immune to known attacks and provides user friendliness. Finally, performance comparisons indicate that our scheme has better security attributes and low computing and communication overheads, and it is more applicable to MCC.

1. Introduction

With the popularity of smart mobile devices, mobile Internet is becoming more and more important in our daily life and deeply affects the way modern humans live and work [1]. Mobile Internet provides high-quality telecommunication services such as voice, fax, data, image, and multimedia. We can obtain a variety of services anytime and anywhere through mobile Internet. Various mobile Internet applications include mobile payment, mobile e-commerce, and mobile entertainment are emerged. Some of these applications such as WeChat and Alipay bring tremendous convenience to people. With the continuous development of mobile Internet, the deficiency that smart mobile devices have limited storage capacity and processing power is gradually revealed. To resolve this issue, cloud computing [2] is introduced into mobile Internet; therefore, a new technology namely mobile cloud computing (MCC) [3] is produced. It aims at solving the resource constrain problem of smart mobile devices, and it can effectively increase the computing power and storage capacity of smart mobile devices.

In an MCC setting, as a trusted third party, the registration center is responsible for issuing the secret key to users and cloud servers in the registration phase. In the authentication phase, the users access the resources and services deployed in distributed cloud servers via mobile and wireless networks, as shown in Figure 1. Due to the openness of the communication networks, the attacker can implement various attacks such as modification, forgery, and replay. It is indispensable to develop an authentication scheme for MCC to achieve identity authentication and secure data transmission, as well as the protection of user privacy.

1.1. Related Works. Since Lamport [4] presented the first password authentication scheme, a large number of schemes [5–18] that are applicable to different scenarios, adopt differ-

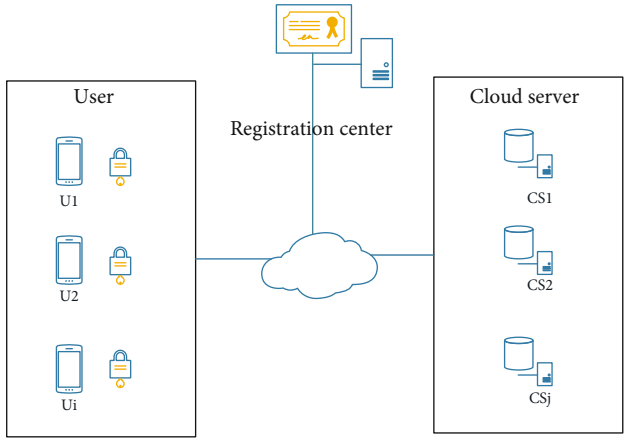

Figure 1: The architecture of MCC.

ent cryptosystems, and employ different kinds of authentication factors were presented. In 2001, Li et al. [17] presented the first multiserver authentication scheme, in which the user can register once and then access multiple servers with a pair of identity and password. Some authentication schemes for MCC [19–22] have been presented in recent years. In 2015, Tsai and Lo [3] introduced an authentication scheme for MCC with offline registration center using bilinear pairing. In 2017, Feng et al. [23] introduced a three-factor mobile multiserver authentication scheme using the elliptic curve cryptosystem (ECC). Amin et al. [24] introduced a lightweight two-factor authentication scheme for MCC. However, their scheme is found to have weaknesses such as offline guessing attack [25]. In 2018, He et al. [26] pointed out that Tsai et al.'s scheme suffers from server impersonation attack. They furthermore proposed an improved scheme by using identity-based signature. Their scheme can provide better security features. In 2019, Irshad et al. [27] presented an enhanced authentication scheme for MCC using bilinear pairing. In 2019, Mo et al. [28] put forward a provably secure two-factor authentication scheme using ECC. In 2020, Li et al. [29] put forward a lattice-based password authenticated key exchange protocol, and their scheme achieves quantum resistance.

1.2. Motivation and Contributions. To improve the security and optimize the efficiency, we design a provably secure authentication scheme using ECC in this paper. Without public key cryptographic techniques, it is difficult to achieve user anonymity and forward secrecy [12]. By using ECC, the proposed scheme provides mutual authentication and user anonymity and establishes secure session key. Compared with the existing schemes with offline registration center using bilinear pairing [3, 26–28], our ECC-based scheme is more efficient. Our major contributions are as follows.

(1) We prove that Mo et al.'s scheme [28] has limitations like stolen-verifier attack, denial of service attack, known session-specific temporary information attack, and its single-server architecture is not applicable to MCC

(2) We put forward a novel authentication scheme for MCC using ECC. It inherits the advantages of existing schemes such as He et al.'s scheme. It enables the user to register once and use a pair of public and private key to access multiple servers. In the authentication phase, the registration center is offline. The user interacts with the cloud server directly. It is conducive to reduce computing and communication overheads

(3) The security analysis demonstrates that the proposed scheme can resist usual attacks and preserve user friendliness. The performance comparisons show that the proposed scheme can remedy the security defects of the existing schemes and incur low computing and communication overheads. The proposed scheme is more suitable for MCC

1.3. Roadmap of Paper. This paper is organized as below. Section 2 gives some preliminaries. Mo et al.'s scheme is cryptanalyzed in Section 3. Section 4 gives the proposed three-factor authentication scheme for MCC. Section 5 is the security analysis. Section 6 is the performance comparisons. Finally, we conclude the paper in Section 7. We summarize some notations in Table 1.

2. Preliminaries

2.1. Elliptic Curve Diffie–Hellman Problem. Elliptic curve Diffie–Hellman problem (ECDHP): E_q is an elliptic curve group over the prime field F_p. P is a generator of E_q. For given αP, βP, where $\alpha, \beta \in Z_q^*$, solving $\alpha\beta P$ is intractable [30].

2.2. Adversary Model. In the light of [31], we suppose that the ability of attacker is as below.

(i) We suppose that the attacker can block, modify, and eavesdrop the message delivered via the public channel

(ii) We suppose that the attacker is able to enumerate all pairs of identity and password subordinate to the dictionary space

(iii) We suppose that the attacker can compromise one type of authentication factor of user, i.e., smart card, password, or biometric

(iv) When evaluating three-factor secrecy, we suppose that the attacker can compromise any two types of authentication factors

3. Analysis of Mo Et al.'s Scheme

3.1. Review of Mo Et al.'s Scheme. We briefly describe Mo et al.'s two-factor single-server authentication scheme for MCC [28] in this section. To initialize the system, the cloud server CS selects the master key s and calculates the public key PUB = sP.

3.1.1. User Registration Phase. This phase is executed as follows.

Table 1: Notations.

Symbols	Description
RC	The registration center
U_i	The user
CS_j	The cloud server
ID_i,PW_i, b_i	U_i's identity, password, and biometric
SID_j	CS_j's identity
P	A generator of elliptic curve group E_q
d_i, PUB_i	U_i's private key and public key
k_j, PUB_j	CS_j's private key and public key
SK	Session key
$\|$	The string concatenation operation
$\oplus$	The bitwise XOR operation
$H_1()$	Hash function
$H_2()$	Biohashing function, it maps the biometric of user to a random string

(Step1) $U_i \longrightarrow \mathrm{CS} : \{\mathrm{ID}_i, R_i\}$. The user U_i selects his identity ID_i, password PW_i, and a nonce r_i and computes $R_i = H_1(r_i \| \mathrm{PW}_i)$.

(Step2) $\mathrm{CS} \longrightarrow U_i$: a smart card. CS picks a nonce N_i and computes $F_i = H_1(H_1(\mathrm{ID}_i \| N_i \| T_i \| \mathrm{SC}_i) \bmod v)$, $A_i = F_i \bigoplus R_i$, where T_i is the current timestamp, v is an integer from $[2^4, 2^8]$, and SC_i is the smart card identification number. CS stores $(\mathrm{ID}_i, N_i, T_i, \mathrm{SC}_i)$ in the database and stores $\{A_i, \mathrm{ID}_S, \mathrm{PUB}, v\}$ in a smart card, where ID_S is the identity of CS

(Step3) U_i computes $B_i = r_i \bigoplus H_1(ID_i \| R_i) \bmod v$ and stores B_i in the smart card

The user U_i selects his identity ID_i, password PW_i, and a nonce r_i and computes $R_i = H_1(r_i \| \mathrm{PW}_i)$.

3.1.2. Authentication Phase. This phase is comprised of the following steps.

(Step1) $U_i \longrightarrow CS : \{PID_i, C_i, L_i\}$. U_i enters ID_i^*, PW_i^*. Then, the smart card computes $R_i^* = H_1(r_i \| \mathrm{PW}_i^*)$, $B_i^* = r_i \bigoplus H_1(ID_i^* \| R_i^*) \bmod v$ and checks if $B_i^* = B_i$. If it holds, the smart card chooses a nonce r_1 and computes $C_i = r_1 P$, $D_i = r_1 \mathrm{PUB}$, $E_i = C_i + D_i$, $F_i = A_i \bigoplus R_i$, the dynamic identity $\mathrm{PID}_i = (\mathrm{ID}_i^* \| F_i) \bigoplus H_1(C_i \| E_i)$, and $L_i = H_1(\mathrm{ID}_i^* \| D_i \| \mathrm{PID}_i)$.

(Step2) $CS \longrightarrow U_i : \{M_1, M_3\}$. CS computes $D_i = sC_i$, $E_i = C_i + D_i$, $(\mathrm{ID}_i \| F_i) = \mathrm{PID}_i \bigoplus H_1(C_i \| E_i)$, and $L_i^* = H_1(\mathrm{ID}_i \| D_i \| \mathrm{PID}_i)$ and checks if $L_i^* = L_i$. If it does not hold, the protocol aborts. Otherwise, CS retrieves $(\mathrm{ID}_i, N_i, T_i, \mathrm{SC}_i)$ from the database based on ID_i and computes $F_i^* = H_1(H_1(\mathrm{ID}_i \| N_i \| T_i \| \mathrm{SC}_i) \bmod v)$ and checks if $F_i^* = F_i$. If they are equal, CS chooses a nonce r_2 and computes $M_1 = r_2 P$, $M_2 = r_2 C_i$, the session key $\mathrm{SK} = H_1(\mathrm{ID}_i \| ID_S \| D_i \| M_1 \| M_2)$, and $M_3 = H_1(\mathrm{ID}_i \| ID_S \| C_i \| D_i \| M_1 \| M_2)$.

(Step3) $U_i \longrightarrow CS : \{M_4\}$. U_i computes $M_2 = r_1 M_1$, $M_3^* = H_1(\mathrm{ID}_i \| \mathrm{ID}_S \| C_i \| D_i \| M_1 \| M_2)$, and checks if $M_3^* = M_3$. If they are equal, U_i computes $\mathrm{SK} = H_1(\mathrm{ID}_i \| \mathrm{ID}_S \| D_i \| M_1 \| M_2)$ and $M_4 = H_1(\mathrm{ID}_i \| \mathrm{ID}_S \| D_i \| M_2 \| \mathrm{SK})$.

(Step4) CS computes $M_4^* = H_1(\mathrm{ID}_i \| \mathrm{ID}_S \| D_i \| M_2 \| \mathrm{SK})$ and checks if $M_4^* = M_4$. If they are not equal, the protocol aborts

U_i enters ID_i^*, PW_i^*. Then, the smart card computes $R_i^* = H_1(r_i \| \mathrm{PW}_i^*)$, $B_i^* = r_i \bigoplus H_1(ID_i^* \| R_i^*) \bmod v$ and checks if $B_i^* = B_i$. If it holds, the smart card chooses a nonce r_1 and computes $C_i = r_1 P$, $D_i = r_1 \mathrm{PUB}$, $E_i = C_i + D_i$, $F_i = A_i \bigoplus R_i$, the dynamic identity $\mathrm{PID}_i = (\mathrm{ID}_i^* \| F_i) \bigoplus H_1(C_i \| E_i)$, and $L_i = H_1(\mathrm{ID}_i^* \| D_i \| \mathrm{PID}_i)$.

3.1.3. Smartcard Revocation Phase. The smart card can be revoked through the following steps.

(Step1) U_i performs step 1 of the authentication phase. U_i sends a revocation request $\{\mathrm{PID}_i, C_i, L_i,$ revoke_request$\}$ to CS

(Step2) CS checks if $L_i^* = L_i$ and $F_i^* = F_i$. If they are equal, CS deletes $(\mathrm{ID}_i, N_i, T_i, \mathrm{SC}_i)$ from the database

Performs step 1 of the authentication phase. U_i sends a revocation request $\{\mathrm{PID}_i, C_i, L_i,$ revoke_request$\}$ to CS

After that, the smart card cannot be used to login CS. The user reregisters with CS to get a new smart card.

3.2. Weaknesses of Mo Et al.'s Scheme. In this section, we prove that Mo et al.'s scheme is not immune to various attacks.

3.2.1. Stolen-Verifier Attack. In Mo et al.'s scheme, CS stores a tuple $(\mathrm{ID}_i, N_i, T_i, \mathrm{SC}_i)$ for each user U_i. If the attacker compromises CS and retrieves $(\mathrm{ID}_i, N_i, T_i, \mathrm{SC}_i)$ from the database, the attacker can masquerade as the legitimate user through the following steps.

(Step1) The attacker computes $F_i = H_1(H_1(\mathrm{ID}_i \| N_i \| T_i \| \mathrm{SC}_i) \bmod v)$

(Step2) The attacker chooses a nonce r_1 and computes $C_i = r_1 P$, $D_i = r_1 \mathrm{PUB}$, $E_i = C_i + D_i$, $\mathrm{PID}_i = (\mathrm{ID}_i \| F_i) \bigoplus H_1(C_i \| E_i)$, $L_i = H_1(\mathrm{ID}_i \| D_i \| \mathrm{PID}_i)$. U_i sends $\{\mathrm{PID}_i, C_i, L_i\}$ to CS

As $L_i^* = L_i$ and $F_i^* = F_i$, CS regards the attacker as the legitimate user U_i. The essential reason for this attack is that the secret authentication value F_i is merely based on the information stored in verification table, rather than the secret key of CS.

3.2.2. Denial of Service Attack. This attack is performed as follows.

(Step1) The adversary intercepts $\{\text{PID}_i, C_i, L_i\}$ from the public channel

(Step2) The attacker sends $\{\text{PID}_i, C_i, L_i, \text{revoke\_request}\}$ to CS

After receiving $\{\text{PID}_i, C_i, L_i, \text{revoke\_request}\}$, as it is valid, CS deletes $(\text{ID}_i, N_i, T_i, \text{SC}_i)$ from the database. After that, the legitimate user U_i is unable to access CS unless reregistration. The essential reason for this attack is that CS does not check the freshness of $\{\text{PID}_i, C_i, L_i, \text{revoke\_request}\}$. The attacker can forge a revocation request using the intercepted $\{\text{PID}_i, C_i, L_i\}$.

3.2.3. Known Session-Specific Temporary Information Attack. Once the attacker compromises the nonce r_1, he can reveal the session key through the following steps.

(Step1) The attacker intercepts $\{\text{PID}_i, C_i, L_i\}$ and $\{M_1, M_3\}$ from the public channel

(Step2) The attacker obtains the user identity by shoulder peeping or computing $D_i = r_1\text{PUB}$, $E_i = C_i + D_i$, $(\text{ID}_i \| F_i) = \text{PID}_i \bigoplus H_1(C_i \| E_i)$

(Step3) The attacker can obtain ID_s by compromising user's smart card or colluding with a user

(Step4) The attacker computes $M_2 = r_1 M_1$, $\text{SK} = H_1(ID_i \| \text{ID}_S \| D_i \| M_1 \| M_2)$.

3.2.4. Not Applicable to Mobile Cloud Computing. Mo et al.'s scheme adopts single server architecture. Only a single server is used to handle the access requests of users. However, in the MCC environment, a large number of users access the cloud server to obtain a variety of services using mobile devices. It is impracticable for a single server to deal with all the access requests in time. MCC aims at integrating the resources and computing power of multiple distributed servers. As depicted in Figure 1, the MCC architecture usually involves multiple distributed servers. In Mo et al.'s scheme, its single-server architecture is not applicable to MCC.

4. The Proposed Scheme

In this section, we put forward an ECC-based three-factor authentication scheme for MCC. It includes three kinds of participants, i.e., the registration center RC, the cloud server CS_j, and the user U_i. As a trusted third party, RC is responsible for issuing the secret key to users and cloud servers in the registration phase. In the authentication phase, RC is offline. U_i and CS_j implement mutual authentication and negotiate a session key without the registration center involved.

4.1. Predeployment Phase. RC selects an elliptic curve group E_q over the prime field F_p. P is a generator of E_q. RC selects the master key s. *RC* chooses a secure hash function $H_1()$ and a biohashing function $H_2()$. RC publishes the parameters $\{E_q, P\}$.

4.2. User Registration Phase. This phase is depicted as Figure 2.

(Step1) The user U_i chooses his identity ID_i and password PW_i, imprints his biometric b_i, and computes $\text{RPW}_i = H_1(\text{ID}_i \| \text{PW}_i \| H_2(b_i) \| y_i)$, where y_i is a nonce. U_i delivers the message $\{\text{ID}_i, \text{RPW}_i\}$ to RC via the reliable channel

(Step2) After getting $\{\text{ID}_i, \text{RPW}_i\}$, RC computes U_i's private key $d_i = H_1(\text{ID}_i \| s \| \text{RPW}_i)$ and public key $\text{PUB}_i = d_i P$, $W_i = d_i \bigoplus \text{RPW}_i$, $Z_i = H_1(\text{RPW}_i) \bmod v$. RC chooses an integer $v \in [2^4, 2^8]$. RC stores the parameters $\{W_i, Z_i, v\}$ in a smart card and publishes U_i's public key $\{\text{ID}_i, \text{PUB}_i\}$. RC issues the smart card to U_i in a credible manner

(Step3) U_i saves y_i in the smart card

4.3. Cloud Server Registration Phase. This phase is depicted as Figure 3.

(Step1) The cloud server CS_j delivers his identity $\{\text{SID}_j\}$ to RC via the reliable channel

(Step2) Upon getting $\{\text{SID}_j\}$, RC computes CS_j's private key $k_j = H_1(\text{SID}_j \| s)$ and public key $\text{PUB}_j = k_j P$. RC publishes the parameters $\{\text{SID}_j, \text{PUB}_j\}$. RC issues $\{k_j\}$ to CS_j in a credible manner

4.4. Authentication Phase. This phase is depicted as Figure 4.

(Step1) U_i enters ID_i^* and PW_i^* and imprints b_i^*. The smart card computes $\text{RPW}_i^* = H_1(\text{ID}_i^* \| \text{PW}_i^* \| H_2(b_i^*) \| y_i)$, $Z_i^* = H_1(RPW_i^*) \bmod v$, and checks if $Z_i^* = Z_i$. If they are equal, the smart card chooses two random numbers r_1 and r_2 and computes $d_i = W_i \oplus \text{RPW}_i^*$, $A_i = r_1 P$, $B_i = r_1 \text{PUB}_j$, $N_i = r_2 P$, $C_i = H_1(A_i \| \text{ID}_i \| N_i)$, $D_i = r_1 + d_i C_i$, $E_i = B_i \bigoplus (\text{ID}_i \| D_i \| N_i)$. U_i sends the message $\{A_i, N_i, E_i\}$ to CS_j via the public channel

(Step2) Upon receiving $\{A_i, N_i, E_i\}$, CS_j computes $B_i = k_j A_i$, $(\text{ID}_i \| D_i \| N_i) = E_i \bigoplus B_i$, $C_i = H_1(A_i \| \text{ID}_i \| N_i)$ and checks if $D_i P = A_i + C_i \bullet \text{PUB}_i$. If it holds, CS_j chooses a random number r_3 and computes $F_i = r_3 P$, the session $\text{keySK} = H_1(r_3 N_i \| D_i)$, $L_i = H_1(\text{SK} \| F_i)$. CS_j sends $\{F_i, L_i\}$ to U_i

(Step3) After receiving $\{F_i, L_i\}$, the smart card computes $\text{SK} = H_1(r_2 F_i \| D_i)$, $L_i^* = H_1(\text{SK} \| F_i)$ and verifies if $L_i^* = L_i$. If so, the smart card computes $M_i = H_1(\text{SK} \| B_i)$. U_i sends $\{M_i\}$ to CS_j

(Step4) Upon getting $\{M_i\}$, CS_j computes $M_i^* = H_1(\text{SK} \| B_i)$ and checks if $M_i^* = M_i$. If they are equal,

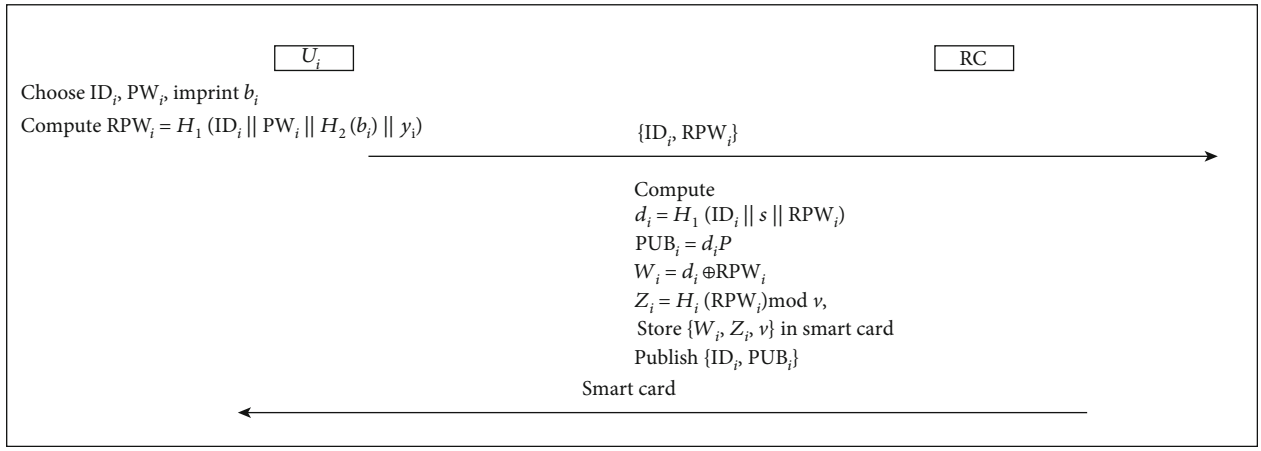

Figure 2: User registration phase of the proposed scheme.

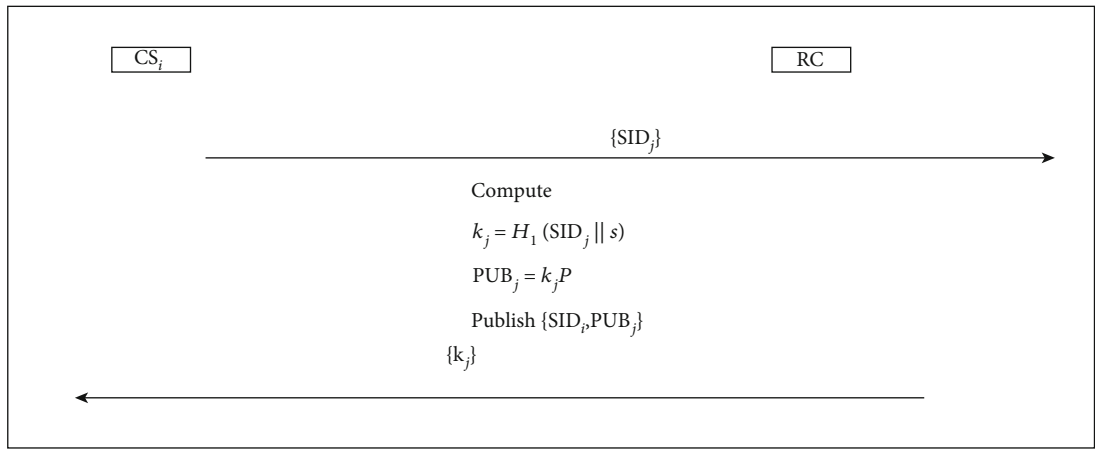

Figure 3: Cloud server registration phase of the proposed scheme.

CS_j and U_i achieve mutual authentication and establish a session key SK

4.5. Smart Card Revocation Phase. If user's smart card is lost or stolen, the user suspects that the data of smart card is leaked. The user reregisters with RC. RC publishes user's new public key information $\{ID_i, PUB_i^{new}\}$ and issues a new smart card to U_i. Afterwards, the user's old smart card is unable to be used to login any cloud server.

4.6. Password and Biometric Update Phase. This phase is executed as follows.

(Step1) U_i inputs ID_i^* and PW_i^* and imprints b_i^*. The smart card computes $RPW_i^* = H_1(ID_i^* \| PW_i^* \| H_2(b_i^*) \| y_i)$, $Z_i^* = H_1(RPW_i^*) \bmod v$ and checks if $Z_i^* = Z_i$. If they are equal, ask the user to input his new password and imprint his new biometric

(Step2) The smart card chooses a new nonce y_i^{new} and computes $RPW_i^{new} = H_1(ID_i^* \| PW_i^{new} \| H_2(b_i^{new}) \| y_i^{new})$, $Z_i^{new} = H_1(RPW_i^{new}) \bmod v$, and $W_i^{new} = W_i \oplus RPW_i^* \oplus RPW_i^{new}$. The smart card saves W_i^{new}, Z_i^{new} and deletes W_i, Z_i

U_i inputs ID_i^* and PW_i^* and imprints b_i^*. The smart card computes $RPW_i^* = H_1(ID_i^* \| PW_i^* \| H_2(b_i^*) \| y_i)$, $Z_i^* = H_1(RPW_i^*) \bmod v$ and checks if $Z_i^* = Z_i$. If they are equal, ask the user to input his new password and imprint his new biometric

5. Security Analysis

In this section, we prove the security of the proposed scheme by using the following security analysis methods.

5.1. BAN Logic Proof. In this section, we show that the proposed scheme preserves mutual authentication and session key agreement by using BAN logic proof. We present the notations and rules of BAN logic [32] in Table 2.

The proposed scheme should be able to achieve the following goals.

G1: $U_i \mid \equiv CS_j \mid \equiv (U_i \overset{SK}{\leftrightarrow} CS_j)$

G2: $U_i \mid \equiv (U_i \overset{SK}{\leftrightarrow} CS_j)$

G3: $CS_j \mid \equiv U_i \mid \equiv (U_i \overset{SK}{\leftrightarrow} CS_j)$

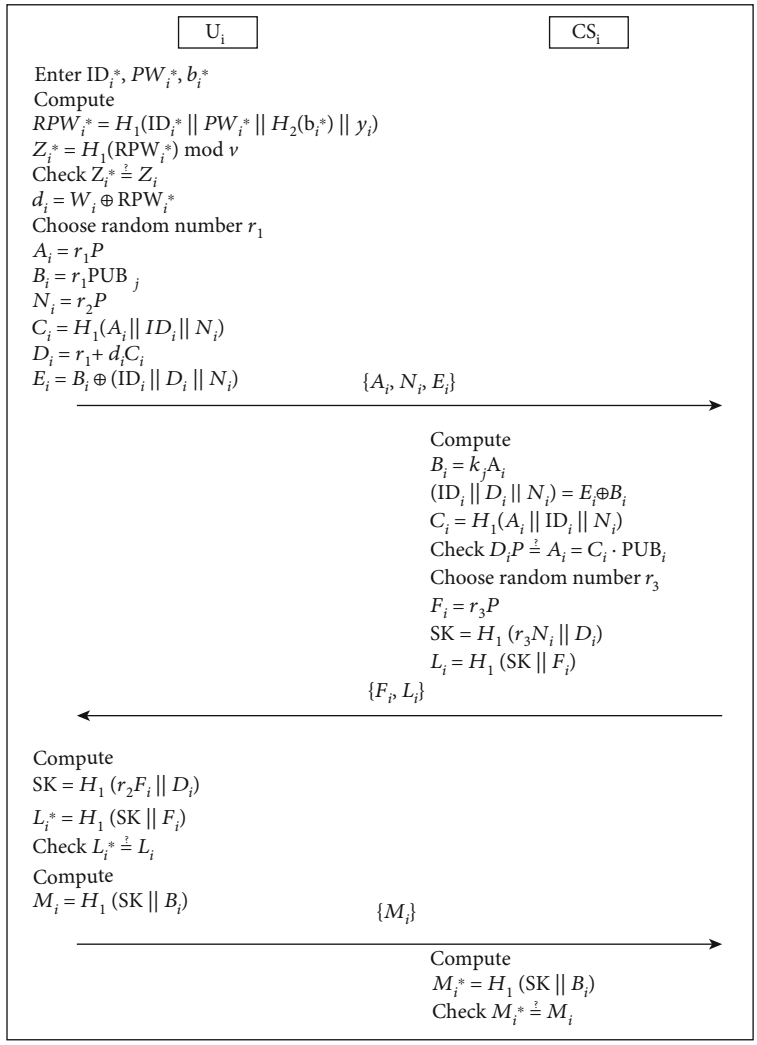

Figure 4: Authentication phase of the proposed scheme.

G4: $CS_j| \equiv (U_i \overset{SK}{\leftrightarrow} CS_j)$

The proposed scheme is idealized as below.

M1: $U_i \rightarrow CS_j\{D_i = \{ID_i, r_1P, r_2P\}_{d_i}, r_2P\}_{B_i}$

M2: $CS_j \rightarrow U_i < U_i \overset{SK}{\leftrightarrow} CS_j, r_3P>_{D_i}$

M3: $U_i \rightarrow CS_j < U_i \overset{SK}{\leftrightarrow} CS_j, r_1P>_{B_i}$

The initial assumptions of the proposed scheme are as below.

A1: $CS_j| \equiv U_i \overset{B_i}{\leftrightarrow} CS_j$

A2: $CS_j| \equiv \overset{PUB_i}{\longrightarrow} U_i$

A3: $CS_j| \equiv \#(r_1P)$

A4: $CS_j| \equiv U_i \Rightarrow r_2P$

A5: $U_i| \equiv U_i \overset{D_i}{\leftrightarrow} CS_j$

A6: $U_i| \equiv \#(r_3P)$

A7: $U_i| \equiv CS_j \Rightarrow U_i \overset{SK}{\leftrightarrow} CS_j$

A8: $CS_j| \equiv U_i \Rightarrow U_i \overset{SK}{\leftrightarrow} CS_j$

The proof is as follows.

From M1, we have

(1) $CS_j \triangleleft \{D_i, r_2P\}_{B_i}$

Apply Rule 1 to (1) and A1, we have

(2) $CS_j| \equiv U_i| \sim (D_i, r_2P)$

From (2), we have

(3) $CS_j| \equiv U_i| \sim D_i(\{ID_i, r_1P, r_2P\}_{d_i})$

Apply Rule 1 to (3) and A2, we have

(4) $CS_j| \equiv U_i| \sim (ID_i, r_1P, r_2P)$

Apply Rule 2 to (4) and A3, we have

(5) $CS_j| \equiv U_i| \equiv (ID_i, r_2P)$

Apply Rule 3 to (5) and A4, we have

(6) $CS_j| \equiv r_2P$

From M2, we have

(7) $U_i \triangleleft < U_i \overset{SK}{\leftrightarrow} CS_j, r_3P>_{D_i}$

Apply Rule 1 to (7) and A5, we have

(8) $U_i| \equiv CS_j| \sim (U_i \overset{SK}{\leftrightarrow} CS_j, r_3P)$

Apply Rule 2 to (8) and A6, we have

(9) $\boldsymbol{U_i| \equiv CS_j| \equiv U_i \overset{SK}{\leftrightarrow} CS_j}$ **(G1)**

Apply Rule 3 to (9) and A7, we have

Table 2: The notations and rules of BAN logic.

Symbols	Description
P, Q	A principal
X	A statement
$\#(X)$	X is fresh
$P \triangleleft X$	P gets X
$P \mid \sim X$	X is sent by P
$P \mid \equiv X$	P believes X
$P \overset{K}{\leftrightarrow} Q$	P and Q have a common secret K
$\{X\}_K$	X is encrypted under K
$\overset{K}{\rightarrow} P$	K is the public key of P
$P \Rightarrow X$	P has jurisdiction over X
$<X>_K$	X is merged with K
Message meaning rule (rule 1)	$\frac{P \mid \equiv P \overset{K}{\leftrightarrow} Q, P \triangleleft <X>_K}{P \mid \equiv Q \mid \sim X}$ or $\frac{P \mid \equiv \overset{K}{\rightarrow} Q, P \triangleleft \{X\}_{K^{-1}}}{P \mid \equiv Q \mid \sim X}$
Nonce-verification rule (rule 2)	$\frac{P \mid \equiv \#(X), P \mid \equiv Q \mid \sim X}{P \mid \equiv Q \mid \equiv X}$
Jurisdiction rule (rule 3)	$\frac{P \mid \equiv Q \Rightarrow X, P \mid \equiv Q \mid \equiv X}{P \mid \equiv X}$

(10) $\boldsymbol{U_i} \mid \equiv \boldsymbol{U_i} \overset{\boldsymbol{SK}}{\leftrightarrow} \boldsymbol{CS_j}$**(G2)**

From M3, we have

(11) $CS_j \triangleleft <U_i \overset{SK}{\leftrightarrow} CS_j, r_1 P>_{B_i}$

Apply Rule1 to (11) and A1, we have

(12) $CS_j \mid \equiv U_i \mid \sim (U_i \overset{SK}{\leftrightarrow} CS_j, r_1 P)$

Apply Rule 2 to (14) and A3, we have

(13) $\boldsymbol{CS_j} \mid \equiv \boldsymbol{U_i} \mid \equiv (\boldsymbol{U_i} \overset{\boldsymbol{SK}}{\leftrightarrow} \boldsymbol{CS_j})$**(G3)**

Apply Rule 3 to (15) and A8, we have

(14) $\boldsymbol{CS_j} \mid \equiv (\boldsymbol{U_i} \overset{\boldsymbol{SK}}{\leftrightarrow} \boldsymbol{CS_j})$**(G4)**

5.2. Formal Security Analysis. In this section, we show that the proposed scheme is provably secure under the security model introduced in [33].

5.2.1. Security Model

(1) Participants. The proposed scheme involves three kinds of participants, i.e., the registration center RC, the cloud server CS_j, and the user U_i. RC^a, CS_j^a, and U_i^a are the a -th instances of RC, CS_j, and U_i, respectively.

(2) Queries. The adversary capability is simulated through the following queries.

Execute (CS_j^a/U_i^a). It simulates the passive attack. It returns back the transcript of messages to the adversary.

Send $(CS_j^a/U_i^a, m)$. It simulates the active attack. The adversary masquerades as the instance CS_j^a/U_i^a by sending a message m. The oracle processes m and returns a response to the adversary.

Reveal (CS_j^a/U_i^a). It returns back CS_j^a/U_i^a's session key to the adversary.

Corrupt (U_i^a, z). It returns back one or two kinds of user authentication factors to the adversary.

If $z = 1$, it returns back the password.

If $z = 2$, it returns back the data of smart card.

If $z = 3$, it returns back the biometric.

Corrupt (RC^a/CS_j^a). It simulates forward secrecy. The oracle returns back the master key of RC^a or the private key of CS_j^a to the adversary.

Test (CS_j^a/U_i^a). It simulates the semantic security of the session key, If the instance CS_j^a/U_i^a is accepted by its partner and establishes a session key SK, and the adversary never makes Corrupt (RC^a/CS_j^a) or Reveal (CS_j^a/U_i^a) query, we say the instance CS_j^a/U_i^a is fresh. If CS_j^a/U_i^a is fresh, the oracle tosses a coin b. If $b = 1$, it answers SK. Otherwise, it chooses an equal-length string and sends it to the adversary. The adversary is allowed to make this query no more than once.

(3) Semantic Security. After receiving the answer from Test (CS_j^a/U_i^a) query, the adversary tries to reveal the value of b. We define the advantage that adversary breaks the semantic security of the proposed scheme as

$$\mathrm{Adv}_P^{\mathrm{ake}}(\mathscr{A}) = 2\ \Pr\left(b' = b\right) - 1. \tag{1}$$

If $\mathrm{Adv}_P^{\mathrm{ake}}(\mathscr{A})$ is negligible, the proposed scheme achieves semantic security.

5.2.2. Security Analysis

Theorem 1. *As demonstrated in [34], the password distribution follows Zipf's law. $|D_{PW}|$ denotes the password dictionary space. C' and s' are parameters of the Zipf distribution. Adv_P^{ECDHP} denotes the advantage that the adversary $\mathscr{A}$ solves ECDHP. The adversary $\mathscr{A}$ can make at most q_e Execute queries, q_s Send queries, q_h Hash queries, and q_b Biohashing queries in polynomial time t. We have*

$$\begin{aligned} Adv_P^{ake}(\mathscr{A}) \le 2C' * {q_s}^{s'} + \frac{(q_s + q_e)^2}{p} + \frac{6q_s + q_h^2}{2^{l_1}} \\ + \frac{2q_s + q_b^2}{2^{l_2}} + 2q_h Adv_P^{ECDHP}, \end{aligned} \tag{2}$$

where l_1 is the length of the hash value, and l_2 is the length of the biohashing value, in terms of the Tianya password dictionary [35] of size $|D_{PW}| \approx 13$ million, $C' = 0.062239$, $s' = 0.155478$.

Proof. The security of the proposed scheme is demonstrated through a series of games Φ_i $(0 \le i \le 6)$, and $\Pr[\chi_i]$ denotes the advantage that $\mathscr{A}$ guesses b in Φ_i.

Φ_0: this game represents the real attack. Hence,

$$\mathrm{Adv}_P^{\mathrm{ake}}(\mathscr{A}) = 2(\Pr[\chi_0]) - 1. \quad (3)$$

Φ_1: the hash oracle and biohashing oracle are simulated by setting up two lists Λ_H and Λ_{BH}. For a Hash query $H_1(\tau)$, the oracle uses τ to search Λ_H. If an item (τ, γ) is found, it sends back γ to the adversary. Otherwise, it returns a random number γ to the adversary and adds a new item (τ, γ) to Λ_H. The biohashing oracle is simulated in the same way. There is no difference between Φ_1 and Φ_0. Hence,

$$\Pr[\chi_1] - \Pr[\chi_0] = 0. \quad (4)$$

Φ_2: This game is terminated when some collisions occur.

(1) A collision appears in random numbers. The probability is no more than $(q_s + q_e)^2/2p$

(2) A collision appears in hash values or biohashing values. The probability is no more than $q_h^2/2^{l_1+1} + q_b^2/2^{l_2+1}$

Hence,

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

Φ_3: we abort the game when $\mathscr{A}$ has guessed (D_i, L_i, M_i). Its advantage is no more than $q_s/2^{l_1}$. Hence,

$$|\Pr[\chi_3] - \Pr[\chi_2]| \le q_s/2^{l_1}. \quad (6)$$

Φ_4: we abort the game when $\mathscr{A}$ has guessed user's secret key d_i. Its advantage is no more than $q_s/2^{l_1}$. Hence,

$$|\Pr[\chi_4] - \Pr[\chi_3]| \le q_s/2^{l_1}. \quad (7)$$

Φ_5: we abort the game when $\mathscr{A}$ has computed d_i having the aid of Corrupt (U_i^a, z) query.

(1) If $\mathscr{A}$ has obtained user's password and biometric, he is able to reveal the key parameter W_i with probability $q_s/2^{l_1}$

(2) If $\mathscr{A}$ has obtained user's password and the data of smart card, he is able to reveal the biometric with probability $q_s/2^{l_2}$

(3) If $\mathscr{A}$ has obtained user's biometric and the data of smart card, he is able to reveal the password with probability $C' * {q_s}^{s'}$

Hence,

$$|\Pr[\chi_5] - \Pr[\chi_4]| \le q_s/2^{l_2} + C' * {q_s}^{s'} + q_s/2^{l_1}. \quad (8)$$

Φ_6: in this game, the hash oracle H_1 is replaced by the private hash oracle H_1' to calculate the session key. H_1' is unavailable to $\mathscr{A}$. Hence,

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

Φ_6 has no difference with Φ_5, unless $\mathscr{A}$ has asked Hash query $H_1(r_3 N_i \| D_i)$. This event is denoted by Γ_1. Hence,

$$|\Pr[\chi_6] - \Pr[\chi_5]| \le \Pr[\Gamma_1]. \quad (10)$$

If $\mathscr{A}$ has asked Hash query $H_1(r_3 N_i \| D_i)$, when picking an item from Λ_H, we can get a solution of ECDHP with probability $1/q_h$. Hence,

$$\Pr[\Gamma_1] \le q_h \mathrm{Adv}_P^{\mathrm{ECDHP}}. \quad (11)$$

From (3)–(11), we have

$$\mathrm{Adv}_P^{\mathrm{ake}}(\mathscr{A}) \le 2C' * {q_s}^{s'} + \frac{(q_s + q_e)^2}{p} + \frac{6q_s + q_h^2}{2^{l_1}} + \frac{2q_s + q_b^2}{2^{l_2}} + 2q_h \mathrm{Adv}_P^{\mathrm{ECDHP}}. \quad (12)$$

5.3. Further Security Analysis. This section demonstrates that the proposed scheme is immune to known attacks and provides various desirable security properties.

5.3.1. Mutual Authentication. In our scheme, the cloud server authenticates the user by checking if $D_i P = A_i + C_i \bullet \mathrm{PUB}_i$. D_i is a signature calculated based on user private key d_i. Only the user U_i who has the private key d_i can calculate a valid D_i. In addition, the user validates the cloud server by checking if $L_i^* = L_i$. Actually, the user authenticates the cloud server based on $B_i = r_1 \mathrm{PUB}_j = k_j A_i$. In the login request, D_i is encrypted under the key B_i. Except the user U_i, only the cloud server CS_j who has the secret key k_j can compute B_i and retrieve D_i from E_i and generate a valid authenticate value L_i.

5.3.2. Session Key Agreement. The user and the cloud server generate a session key $\mathrm{SK} = H_1(r_3 r_2 P \| D_i)$. The session key is composed of $r_3 r_2 P$ and D_i. $r_3 r_2 P$ is generated using elliptic curve Diffie-Hellman key exchange, and it guarantees forward secrecy. D_i is generated based on user's private key, and it guarantees the resistance of session-specific temporary information attack.

5.3.3. User Anonymity. In our scheme, the user identity is encrypted under the key B_i. As ECDHP is intractable, only the user who knows the random number r_1 and the cloud server who has the secret key k_j can retrieve ID_i from E_i. Additionally, the random numbers r_1 and r_2 are involved in the login request $\{A_i, E_i\}$. The login requests are different in each session. Thus, the proposed scheme preserves user untraceability.

5.3.4. Offline RC. In the authentication phase, the user and the cloud server can perform mutual authentication and

Table 3: Security features Comparisons.

Security properties	Tsai and Lo [3]	He et al. [26]	Irshad et al. [27]	Mo et al. [28]	Our scheme
User anonymity	×	√	√	√	√
Resist server impersonation attack	×	√	×	√	√
Resist offline guessing attack	√	√	√	√	√
Resist stolen-verifier attack	√	√	√	×	√
Resist denial of service attack	√	√	√	×	√
Resist replay attack	√	√	√	×	√
Resist known session-specific temporary information attack	×	×	×	×	√
Forward secrecy	√	√	√	√	√
Three-factor secrecy	−	−	√	−	√
Efficiency for wrong password and biometric detection	×	×	√	√	√
Offline RC	√	√	√	−	√
Single/multi server	Multiserver	Multiserver	Multiserver	Single-server	Multiserver
Cryptography primitives	Paring	Paring	Paring	ECC	ECC

session key agreement without the aid of RC. It reduces the number of interacted messages. Correspondingly, it helps to reduce communication and computing overheads.

5.3.5. Forward Secrecy. The session key is computed based on $SK = H_1(r_3r_2P \| D_i \| r_3\mathrm{PUB}_i)$. r_3r_2P is generated using Diffie-Hellman key exchange. Due to the intractability of ECDHP, even the attacker obtains the long-term secret, he is unable to retrieve r_3r_2P from F_i and N_i. The proposed scheme preserves forward secrecy.

5.3.6. Resist Session-Specific Temporary Information Attack. Suppose that the random numbers r_2 is compromised. The adversary computes r_3F_i. However, as B_i is unavailable, the adversary cannot obtain D_i.

Suppose that the random number r_3 is compromised. The adversary cannot obtain N_i and D_i, as B_i is unavailable. The adversary can neither obtain D_i or r_3N_i.

As a result, the adversary cannot reveal the session key when the random number is compromised.

5.3.7. Resist Forgery Attack. In our scheme, the user computes the signature D_i based on the private key d_i to authenticate the message $\{A_i, E_i, N_i\}$. Afterwards, the cloud server uses the shared session key SK to authenticate the message $\{F_i, L_i\}$. Finally, the user uses the shared session key SK to authenticate the message $\{M_i\}$. As the secret key d_i and SK are unavailable, the adversary cannot produce a valid message.

5.3.8. Resist Replay Attack. In the proposed scheme, the cloud server authenticates the user by checking the validity of the messages $\{A_i, E_i, N_i\}$ and $\{M_i\}$. If the adversary replays $\{A_i, E_i, N_i\}$, as he cannot produce a valid $\{M_i\}$, ultimately, the authentication fails. If the adversary replays $\{F_i, L_i\}$ and $\{M_i\}$, as the random numbers selected in each session are different, the authentication fails. Hence, the proposed scheme can resist replay attack.

5.3.9. Resist Insider Attack. The user cannot impersonate the cloud server without cloud server's private key. Similarly, the cloud server cannot impersonate the user without user's private key. The other users cannot pretend to be the user U_i, as he cannot generate a valid signature of U_i. The other cloud servers cannot pretend to be the cloud server CS_j, as he cannot decrypt E_i to get D_i. Our scheme is resistance to insider attack.

Table 4: Executing time of some cryptography operations.

Cryptography operations	Symbols	Running time (ms)	
		User	Server
Map-to-point hash function	T_{PH}	33.582	5.493
Bilinear paring	T_B	32.713	5.427
Elliptic curve point multiplication	T_P	13.405	2.165
Elliptic curve point addition	T_A	0.081	0.013
Exponentiation operation	T_E	2.249	0.339
Hash function	T_H	0.056	0.007

5.3.10. User Friendliness. The proposed scheme provides user friendliness. Firstly, the proposed scheme adopts multiserver architecture. The user only needs to register once to access multiple servers. Secondly, in the authentication phase, the registration center is offline, and the user can access the cloud server directly without interacting with the registration center. Thirdly, the proposed scheme supports smartcard revocation, efficiency for wrong password and biometric detection, and password and biometric update.

5.3.11. Three-Factor Secrecy. The fuzzy verification Z_i makes our scheme that is immune to offline guessing attack. Even if the adversary compromises two kinds of authentication factors, the other one is still unavailable. In addition, for the adversary, the only way to retrieve d_i is to break the password, the biometric, and the smart card at the same time. Without d_i, the adversary cannot impersonate the user. Hence, the proposed scheme preserves three-factor secrecy.

Table 5: Computation costs of related schemes.

Computation cost	User (ms)	Server (ms)	Total (ms)
Tsai and Lo [3]	$T_{PH} + 4T_P + 2T_A + T_E + 5T_H$ (89.893)	$2T_B + 2T_P + 2T_A + 2T_E + 4T_H$ (16.096)	105.989
He et al. [26]	$T_{PH} + 3T_P + 2T_E + 4T_H$ (78.519)	$2T_P + 2T_A + 2T_E + 5T_H$ (11.773)	90.292
Irshad et al. [27]	$1T_B + 4T_P$ (86.333)	$2T_B + 3T_P$ (17.349)	103.682
Mo et al. [28]	$3T_P + T_A + 6T_H$ (40.632)	$3T_P + T_A + 7T_H$ (6.557)	47.189
Our scheme	$4T_P + 6T_H$ (53.956)	$5T_P + 4T_H$ (10.853)	64.809

Table 6: Communication costs of related schemes.

	Tsai and Lo [3]	He et al. [26]	Irshad et al. [27]	Mo et al. [28]	Our scheme
Communication cost	4320 bits	3296 bits	4288 bits	2720 bits	3584 bits

6. Performance Comparisons

The comparative analysis of our scheme and the relevant schemes [3, 26–28] is presented in this section. Our scheme and the relevant schemes are evaluated from two aspects, i.e., security properties and computation and communication overheads.

Table 3 presents the security analysis results of relevant schemes. The security attributes include user anonymity and three-factor secrecy, as well as the resistance of usual attacks. Besides, the characteristics of the proposed schemes and relevant schemes are also detailed in Table 3. The relevent schemes [3, 26 ,27] adopt multiserver architecture, and RC is offline in the authentication phase, while Mo et al.'s scheme adopts single-server architecture. Tsai et al.'s scheme, He et al.'s scheme, and Irshad et al.'s scheme are bilinear paring-based schemes, while Mo et al.'s scheme and our scheme are ECC-based schemes. From Table 3, we witness that the relevant schemes have more or less weaknesses, while the proposed scheme can remedy the security defects of relevant schemes and provides desirable security properties. It shows that the proposed scheme has better security than the relevant schemes.

In accordance with [26], the user uses a mobile device to access the cloud server, the cloud server is deployed in a personal computer, and the executing time of relevant cryptography operations is presented in Table 4. The computation costs of our scheme and the relevant schemes are evaluated as shown in Table 5. The running time of the proposed scheme is 80.379 ms. The running time of the relevant schemes [3, 26–28] is 105.989 ms, 90.292 ms, 103.682 ms, and 47.189 ms, respectively.

To evaluate the communication cost, we suppose that the user identity is 32 bits, the point on the elliptic curve group is 1024 bits, and the hash value is 160 bits. The login request query in [3, 26, 27] is 32 bits. As shown in Table 6, the communication cost of the proposed scheme is 3584 bits. The communication costs of the relevant schemes [3, 26–28] are 4320 bits, 3296 bits, 4288 bits, and 2720 bits, respectively.

Figure 5 presents the comparison of total computation costs, the computation costs of user end, and the computation costs of cloud server. Figure 6 presents the communication cost comparison. In terms of the communication cost,

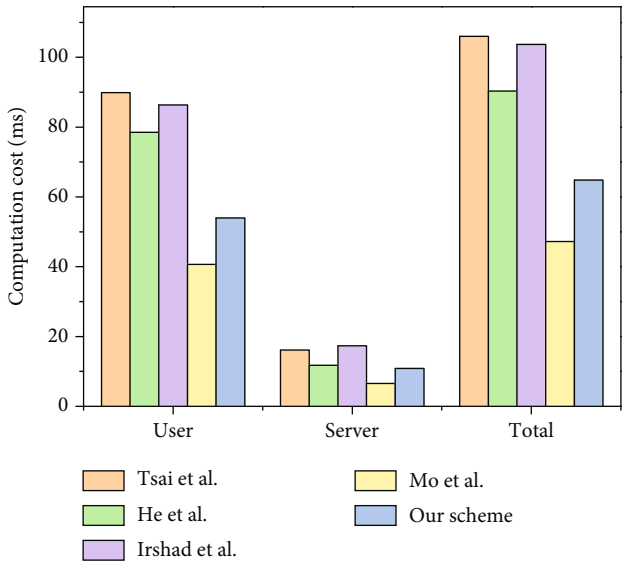

Figure 5: Computation cost comparisons.

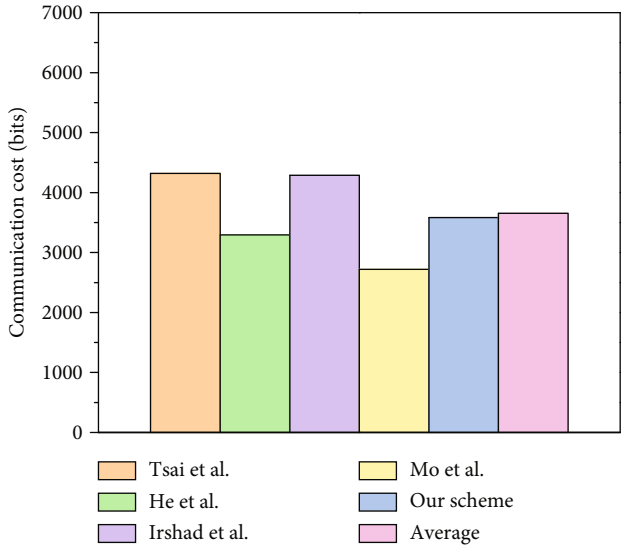

Figure 6: Communication cost comparison.

our scheme is in third place and better than the average communication cost. In terms of the total computation cost, user's computation cost, and server's computation cost, the proposed scheme is second only to Mo et al.'s scheme. However, Mo et al.'s scheme has limitations like stolen-verifier attack and denial of service attack; particularly, its single-server architecture is not applicable to the mobile cloud computing environment.

In a nutshell, our scheme provides more security attributes and has low computation and communication costs. Among the relevant schemes, the security features of He et al.'s scheme are the closest to our scheme. However, the computation cost of our scheme is 0.72 times of He et al.'s scheme. Our scheme achieves balanced security and efficiency. Compared with the relevant schemes, our scheme is more applicable to mobile cloud computing.

7. Conclusion

In this paper, we demonstrate that Mo et al.'s scheme has limitations such as stolen-verifier attack and denial of service attack. Most notably, its single-server architecture is not applicable to MCC. To enhance the security, we present a provably secure ECC-based three-factor authentication scheme. Security analysis shows that our scheme is immune to known attacks and provides user friendliness. Performance comparisons indicate that our scheme provides more security attributes and incus low computation and communication cost. Our scheme is more applicable to MCC. As post-quantum security has become the focus issue of researchers, we plan to use lattice-based key exchange [36] and smooth projective hash functions [37] to construct a quantum-resistant scheme at the next step.

Data Availability

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

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

This research was funded by the National Key Research and Development Program of China (No. 2018YFB0803600 and No. 2017YFB0801903) and by the National Natural Science Foundation of China (No. 61831003 and No. 61897069).

References

[1] A. Ghose, A. Goldfarb, and S. P. Han, *How is the mobile internet different?*, vol. 24, no. 3, 2012Social Science Electronic Publishing, 2012.

[2] P. Gope and A. K. Das, "Robust anonymous mutual authentication scheme for *n*-Times ubiquitous mobile cloud computing services," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1764–1772, 2017.

[3] 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.

[4] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[5] E. J. Yoon, K. Y. Yoo, C. Kim, Y. S. Hong, M. Jo, and H. H. Chen, "A secure and efficient sip authentication scheme for converged VOIP networks," *Computer Communications*, vol. 33, no. 14, pp. 1674–1681, 2010.

[6] F. Wang, G. Xu, and L. Gu, "A secure and efficient ECC-based anonymous authentication protocol," *Security and Communication Networks*, vol. 2019, Article ID 4656281, 13 pages, 2019.

[7] F. Wei, P. Vijayakumar, Q. Jiang, and R. Zhang, "A mobile intelligent terminal based anonymous authenticated key exchange protocol for roaming service in global mobility networks," *IEEE Transactions on Sustainable Computing*, vol. 99, pp. 2377–3782, 2018.

[8] C. Wang, D. Wang, Y. Tu, G. Xu, and H. Wang, "Understanding node capture attacks in user authentication schemes for wireless sensor networks," *IEEE Transactions on Dependable and Secure Computing*, p. 1, 2020.

[9] X. Li, W. Qiu, D. Zheng, K. Chen, and J. Li, "Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 2, pp. 793–800, 2010.

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

[11] 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.

[12] D. Wang and P. Wang, "Understanding security failures of two-factor authentication schemes for real- time applications in hierarchical wireless sensor networks," *Ad Hoc Networks*, vol. 20, pp. 1–15, 2014.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] L. H. Li, L. C. Lin, and M. S. Hwang, "A remote password authentication scheme for multiserver architecture using neural networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1498–1504, 2001.

[18] C. Wang, G. Xu, and J. Sun, "An enhanced three-factor user authentication scheme using elliptic curve cryptosystem for

wireless sensor networks," *Sensors*, vol. 17, no. 12, p. 2946, 2017.

[19] M. B. Mollah, M. A. K. Azad, and A. Vasilakos, "Security and privacy challenges in mobile cloud computing: survey and way ahead," *Journal of Network and Computer Applications*, vol. 84, pp. 38–54, 2017.

[20] 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.

[21] L. Xiong, D. Peng, T. Peng, and H. Liang, "An enhanced privacy-aware authentication scheme for distributed mobile cloud computing services," *KSII Transactions on Internet and Information Systems*, vol. 11, no. 12, pp. 6169–6187, 2017.

[22] Q. Jiang, J. Ma, and F. Wei, "On the security of a privacy-aware authentication scheme for distributed mobile cloud computing services," *IEEE Systems Journal*, vol. 12, no. 2, pp. 2039–2042, 2018.

[23] Q. Feng, D. He, S. Zeadally, and H. Wang, "Anonymous biometrics-based authentication scheme with key distribution for mobile multi-server environment," *Future Generation Computer Systems*, vol. 84, pp. 239–251, 2017.

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

[25] 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.

[26] 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.

[27] A. Irshad, S. A. Chaudhry, M. Shafiq, M. Usman, M. Asif, and A. Ghani, "A provable and secure mobile user authentication scheme for mobile cloud computing services," *International Journal of Communication Systems*, vol. 32, no. 14, article e3980, 2019.

[28] 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.

[29] Z. Li, D. Wang, and E. Morais, "Quantum-safe round-optimal password authentication for mobile devices," *IEEE Transactions on Dependable and Secure Computing*, 2020.

[30] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.

[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] F. Wang, G. Xu, G. Xu, Y. Wang, and J. Peng, "A robust IoT-based three-factor authentication scheme for cloud computing resistant to session key exposure," *Wireless Communications and Mobile Computing*, vol. 2020, Article ID 3805058, 15 pages, 2020.

[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] D. Wang and P. Wang, "On the implications of Zipf's law in passwords," in *Proc. Eur. Symp. Res. Comput. Secur*, pp. 111–131, Cham, 2016.

[36] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, "Post-quantum key exchange—a new hope," in *Proc. USENIX SEC 2016*, pp. 327–343, Austin, TX, 2016.

[37] J. Katz and V. Vaikuntanathan, "Smooth projective hashing and password-based authenticated key exchange from lattices," in *Proc. ASIACRYPT 2009*, pp. 636–652, Berlin, Heidelberg, 2009.