A Provably Secure Biometrics-Based Authentication Scheme for Multiserver Environment

Feifei Wang,1 Guoai Xu,1 Chenyu Wang,1 and Junhao Peng2

1Beijing University of Posts and Telecommunications, Beijing 100876, China
2Guangzhou University, Guangzhou 510006, China

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

Received 9 March 2019; Accepted 9 June 2019; Published 25 June 2019

With the rapid development of mobile services, multiserver authentication protocol with its high efficiency has emerged as an indispensable security mechanism for mobile services. Recently, Ali et al. introduced a biometric-based multiserver authentication scheme and claimed the scheme is resistant to various attacks. However, after a careful examination, we find that Ali et al.’s scheme is vulnerable to various security attacks, such as user impersonation attack, server impersonation attack, privileged insider attack, denial of service attack, fails to provide forward secrecy and three-factor secrecy. To overcome these weaknesses, we propose an improved biometric-based multiserver authentication scheme using elliptic curve cryptosystem. Formal security analysis under the random oracle model proves that our scheme is provably secure. Furthermore, BAN (Burrows-Abadi-Needham) logic analysis demonstrates our scheme achieves mutual authentication and session key agreement. In addition, the informal analysis proves that our scheme is secure against all current known attacks and achieves desirable features. Besides, the performance and security comparison shows that our scheme is superior to related schemes.

1. Introduction

Nowadays, millions of people enjoy various mobile services such as mobile shopping, mobile entertainment, and mobile learning, by using various mobile devices. Due to the openness of mobile network, when the users are enjoying great conveniences brought by mobile services, they simultaneously face a great deal of security threats, such as diverse network attacks and privacy leaks. Authentication protocol plays a great role in protecting the security and privacy of users as an indispensable security mechanism for various mobile services. It provides mutual authentication, user anonymity, and establishes secure session key for server and users [1].

With the continuous expansion of the scale of mobile services, multiserver mode has been widely adopted by numerous mobile service application systems [2]. When the traditional single-server authentication schemes are applied to multiserver environment, it is extremely inconvenient for user to register himself with every server and keep many pairs of identity and password. To overcome this problem, multiserver authentication schemes have been introduced [3–10]. These schemes make the user registers once with registration center and keeps one pair of identity and password to obtain all the services. Multiserver authentication schemes are more attractive as high efficiency and convenience. But on the other hand, multiserver authentication schemes have more requirements for security. The user employs the same authentication information to access diverse servers. If the authentication information is compromised, it will bring tremendous damage to user’s assets. Besides, the malicious server may masquerade another server to defraud the user or impersonate user to access server based on the secret it has. This privileged insider attack should be overcome.

In the past 20 years, many multiserver authentication schemes using password and smart card have been put forward [11–16]. However, the smart card may be lost or stolen, and the malicious attacker can retrieve the data in smart card by side channel attack. It increases the risk of security breach [17]. To overcome this weakness, biometric authentication element has been added in authentication schemes in recent years because of its good characteristics.
Three-factor authentication schemes that adopt password, smart card, and biometric facilitate better security.

Recently, some three-factor multiserver authentication schemes have been introduced. In 2010, Yoon et al. [18] introduced an efficient biometric-based multiserver authentication scheme using elliptic curve cryptosystem (ECC). Later on, Kim et al. [19] pointed out Yoon et al.’s scheme cannot resist smart card loss attack, forgery attack, and fails to provide forward secrecy. In 2015, Amin et al. [20] proposed a three-factor multiserver authentication scheme using bilinear pairing. Afterwards, Chandrakar et al. [21] proved Amin et al.’s scheme is susceptible to offline password guessing attack, impersonation attack, and fails to achieve user anonymity. He et al. [22] introduced a biometric-based multiserver authentication scheme using fuzzy extractor and ECC and claimed their scheme achieves intrinsically three-factor secrecy. But we observed He et al.’s scheme is susceptible to known session-specific temporary information attack and cannot detect wrong password and biometric immediately. In 2016, Wang et al. [23] presented a three-factor multiserver authentication scheme using hash function and fuzzy extractor. But Yang et al. [24] pointed out Wang et al.’s scheme cannot resist user impersonation attack and fails to achieve forward secrecy. In 2017, Kumari et al. [25] proposed a biometric-based multi-cloud-server authentication scheme using ECC and bio-hash function. However, Feng et al. [26] demonstrated that Kumari et al.’s scheme suffers from server impersonation attack and introduced an enhanced scheme. Unfortunately, we found Feng et al.’s scheme fails to achieve three-factor secrecy and suffers from known session-specific temporary information attack. Ali et al. [27] introduced a three-factor multiserver authentication scheme using symmetric encryption and ECC and claimed their scheme is resistant to a variety of serious security attacks.

Either the existing three-factor multiserver authentication schemes [18–30] have more or less vulnerabilities, or their communication and computation costs need to be improved. This moves us to design a secure three-factor multiserver authentication scheme with higher efficiency. Our contributions are summed up as follows.

1. We prove that Ali et al.'s scheme suffers from user impersonation attack, privileged insider attack, server impersonation attack, denial of service attack, and known session-specific temporary information attack. Besides, the scheme fails to achieve forward secrecy and three-factor secrecy.

2. We propose a novel biometric-based multiserver authentication scheme using ECC. Formal security analysis under the random oracle model proves our scheme is provably secure. BAN logic proof proves the completeness of our scheme. Moreover, informal analysis demonstrates our scheme achieves various desirable features and is resistant to all known attacks.

3. In addition, the performance and security comparison shows that our scheme achieves superior security properties. Moreover, our scheme has the least communication overhead and computation cost.

1.1. Adversary Model. When evaluating a three-factor multiserver authentication scheme, the capacities of adversary $\mathcal{A}$ are described as follows.

1. $\mathcal{A}$ may be an external attacker or a privileged insider.

2. $\mathcal{A}$ can fully control the public channel; namely, $\mathcal{A}$ is able to interrupt, eavesdrop, forge, and modify the messages transmitted via public channel.

3. $\mathcal{A}$ is able to enumerate all the values in $D_{PW} \ast D_{ID}$ in polynomial time, where $D_{PW}$ denotes the password space and $D_{ID}$ denotes the identity space [31].

4. $\mathcal{A}$ is able to get user’s password by shoulder surfing. $\mathcal{A}$ can retrieve the data in smart card by power consumption analysis. $\mathcal{A}$ is able to get the biometric of user by a malicious terminal [32].

5. When evaluating three-factor secrecy, $\mathcal{A}$ is able to get any two kinds of authentication elements at the same time but cannot get all [26].

6. When evaluating forward secrecy, $\mathcal{A}$ can get the master key of $RC$ or the secret key of server.

The user tends to choose an easy-to-remember password with low strength. The user identity is usually based on the predefined format. The identity and password may be of low entropy and can be easily guessed. According to the adversary model presented by Wang et al. [31], we assume the adversary $\mathcal{A}$ is able to enumerate all the values in $D_{PW} \ast D_{ID}$ in polynomial time. Three-factor secrecy denotes that if any two kinds of authentication elements are compromised, the attacker still cannot breach the other one and damage the security of the system [26]. Such a consideration is of practical significance. The adversary may get user’s password by shoulder surfing or the data in smart card via side channel attack. Moreover, the adversary is able to obtain the biometric of user by a malicious biometric-based terminal.

1.2. The Organization of Paper. The structure of this paper is arranged as follows. We briefly review and cryptanalyze Ali et al.’s scheme in Sections 2 and 3. Section 4 introduces a novel biometric-based authentication scheme for multiserver environment. We give the security proof and informal security analysis of the proposed scheme in Sections 5 and 6. Section 7 is security and performance comparison of the relevant schemes. Section 8 concludes the paper. In addition, we sum up the notations of this paper in Table 1.

2. Review of Ali et al.’s Scheme

Ali et al.’s scheme consists of four phases: initial phase, server registration phase, user registration phase, login and authentication phase.

2.1. Initial Phase. $RC$ chooses its master key $x$. Then $RC$ selects an elliptic curve group $E_q$ and a generator $P$ of $E_q$. 
Table I: Notations.

<table>
<thead>
<tr>
<th>Symbols</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_i$</td>
<td>$i^{th}$ user</td>
</tr>
<tr>
<td>$S_j$</td>
<td>Server $S_j$</td>
</tr>
<tr>
<td>$RC$</td>
<td>Registration center</td>
</tr>
<tr>
<td>$\mathcal{A}$</td>
<td>Malicious adversary</td>
</tr>
<tr>
<td>$x$</td>
<td>Master key of RC</td>
</tr>
<tr>
<td>$ID_i$, $PW_i$, $b_i$</td>
<td>Identity, password and biometric of $U_i$</td>
</tr>
<tr>
<td>$SID_j$</td>
<td>Identity of $S_j$</td>
</tr>
<tr>
<td>$P$</td>
<td>A generator of elliptic curve group $E_q$</td>
</tr>
<tr>
<td>$E_{Key}()$</td>
<td>Symmetric encryption/decryption algorithm with key $Key$</td>
</tr>
<tr>
<td>$SK$</td>
<td>Session key between $U_i$ and $S_j$</td>
</tr>
<tr>
<td>$|$</td>
<td>The string concatenation operation</td>
</tr>
<tr>
<td>$\oplus$</td>
<td>The bitwise XOR operation</td>
</tr>
<tr>
<td>$H_i()$</td>
<td>Hash function</td>
</tr>
<tr>
<td>$H_j()$</td>
<td>Bio-hash function, it maps the biometric $b_i$ and a tokenised random number to a random binary string</td>
</tr>
</tbody>
</table>

2.2. Server Registration Phase. $S_j$ enrolls with RC in the following steps.

Step 1. The server picks its identity $SID_j$ and sends $[SID_j]$ as a registration request to RC through the reliable channel.

Step 2. Upon receiving $[SID_j]$ from $S_j$, RC computes $SM_j = H_i(SID_j || x)$ and returns $[SM_j]$ to $S_j$ through the reliable channel.

Step 3. $S_j$ keeps $SM_j$ as secret.

2.3. User Registration Phase. $U_i$ enrolls with RC in the following steps.

Step 1. $U_i$ picks his identity $ID_i$ and password $PW_i$ freely and imprints his biometric $b_i$. $U_i$ sends the registration request $\{ID_i, H_2(b_i)\}$ to RC through the reliable channel.

Step 2. Upon receiving $\{ID_i, H_2(b_i)\}$ from $U_i$, RC computes $DID_i = E_{H_i(x)}(ID_i || N_i)$, $A_i = H_1(ID_i || x)P$, $B_i = H_2(b_i)P$, $E_j = A_i + B_i$, where $N_i$ is a random number. RC stores $\{DID_i, E_i, P, E_{Key}\}$ in a smart card and transmits it to $U_i$ through the reliable channel.

Step 3. $U_i$ calculates $V_i = H_j(ID_i || PW_i || H_2(b_i))$ and stores $V_i$ in his smart card.

2.4. Login and Authentication Phase. $U_i$ and $S_j$ authenticate each other and establish a session key drawing support from RC as shown in Figure 1.

Step 1. $U_i$ attaches the smart card to a terminal, inputs $ID_i^*$ and $PW_i^*$, and imprints $b_i^*$. The smart card calculates $V_i^* = H_j(ID_i^* || PW_i^* || H_2(b_i^*))$ and checks if $V_i^* = V_i$. If the equation holds, proceed to the next step.

Step 2. $U_i$ computes $R_i = r_iP$, $B_i^* = H_2(b_i^*)P$, $C_i = R_i + B_i^*$, $D_i = H_j(ID_i^* || R_i || SID_j || E_j)$, where $r_i$ is a random number. $U_i$ sends $\{DID_i, E_i, C_i, D_i\}$ to RC through the public communication channel.

Step 3. After receiving $\{DID_i, E_i, C_i, D_i\}$, RC computes $(ID_i || N_i) = D_{H_i(x)}(DID_i)$, $A_i = H_1(ID_i || x)P$, $B_i = E_i - A_i$, $R_i^1 = C_i - B_i^1$, $D_i^1 = H_1(ID_i || R_i^1 || SID_j || E_j)$ and compares $D_i^1$ with $D_i$. If they are equal, proceed to the next step.

Step 4. RC computes $DID_i^{new} = E_{H_i(x)}(ID_i || r_{RC})$, $F_i = E_{H_i(SID_j,x)}(ID_i || R_i || H_2(b_i))$, $R_{RC} = r_{RC}P$, $K_i = R_2 + R_i^1$, $M_i = H_i(DID_i^{new} || R_S || R_{RC} || H_2(b_i)^*P)$, where $r_{RC}$ is a random number. RC sends $\{DID_i^{new}, K_i, L_i, F_i\}$ to $S_j$.

Step 5. Upon receiving $\{DID_i^{new}, K_i, L_i, F_i\}$, $S_j$ computes $(ID_i || R_i^1 || H_2(b_i)^*) = D_{SM_i}(F_i)$, $R_{RC} = K_i - H_2(b_i)^*P$, $L_i = H_i(R_i^1 || DID_i^{new} || ID_i || SID_j)$ and checks if $L_i = L_i$. If it holds, $S_j$ computes $R_S = r_SQ_i$, $Q_i = R_S + R_i^1$, $M_i = H_i(DID_i^{new} || R_S || R_{RC} || H_2(b_i)^*P)$, where $r_S$ is a random number. $S_j$ sends $\{DID_i^{new}, Q_i, M_i, K_i\}$ to $U_i$.

Step 6. Upon receiving $\{DID_i^{new}, Q_i, M_i, K_i\}$, $U_i$ computes $R_S = Q_i - R_i^1$, $R_{RC} = K_i - B_i^1$, $M_i = H_i(DID_i^{new} || R_i || R_{RC}^i || B_i^1)$ and checks if $M_i = M_i$. If the equation holds, $U_i$ computes $SK = H_i(R_S || R_i^1 || R_{RC}^i)$, $Z_i = SK \cdot P + H_1(ID_i || H_2(b_i)^*) \cdot P$. $U_i$ replaces $DID_i^i$ with $DID_i^{new}$ in his smart card and sends $\{Z_i\}$ to $S_j$.

Step 7. Upon receiving $\{Z_i\}$, $S_j$ computes $SK = H_i(R_i^1 || R_S || R_{RC}^i)$, $Z_i^* = SK \cdot P + H_1(ID_i || H_2(b_i)^*) \cdot P$ and checks if $Z_i^* = Z_i$. If the equation holds, $S_j$ and $U_i$ authenticate each other and establish a session key $SK$ successfully.

3. Cryptanalysis of Ali et al.’s Scheme

In this section, we demonstrate that Ali et al.’s scheme is susceptible to several security attacks. Note that, we cryptanalyze Ali et al.’s scheme on the basis of the adversary capacities mentioned in Section 1.

3.1. Forward Secrecy. The adversary $\mathcal{A}$ compromises the master key $x$ and intercepts $\{DID_i^{new}, K_i, L_i, F_i\}$ and $\{DID_i^{new}, Q_i, M_i, K_i\}$ from public channel. Then $\mathcal{A}$ is able to retrieve the session key in the following steps.

Step 1. Compute $SM_j = H_i(SID_j || x), (ID_i || R_i^1 || H_2(b_i)^*) = D_{SM_i}(F_i)$.

Step 2. Compute $R_S = Q_i - R_i^1$.

Step 3. Compute $R_{RC} = K_i - H_2(b_i)^*P$.

Step 4. Compute $SK = H_i(R_i^1 || R_S || R_{RC}^i)$.

3.2. User Impersonation Attack. The adversary $\mathcal{A}$ gets $U_i$’s identity $ID_i$ by shoulder surfing and $U_i$’s biometric $b_i$ by
Upon receiving a malicious terminal and intercepts \( \{\text{ID}_I\}, \text{DID}, C, D \) from public channel. Then \( \mathcal{A} \) performs user impersonation attack in the following steps.

**Step 1.** \( \mathcal{A} \) computes \( R_I = r_I \cdot p, B_I = H_2(b_I) \cdot P, C_A = R_I + B_I, D_A = H_1(\text{ID}_A \parallel R_I \parallel \text{SID}_J \parallel E_I) \), where \( r_I \) is a random number. \( \mathcal{A} \) sends \( \{\text{ID}_I, E_I, C_A, D_A\} \) to RC.

**Step 2.** Upon receiving \( \{\text{ID}_I, E_I, C_A, D_A\} \), RC computes \( R'_I = C_A - B'_I, D'_A = H_1(\text{ID}_A \parallel R'_I \parallel \text{SID}_J \parallel E_I) \), obviously \( D'_A = D_A \). Then RC computes \( D^{\text{new}}_I = E_{H_1(x)}(\text{ID}_A \parallel r_{RC}) \) and \( F_a = E_{H_1(\text{SID}_J \parallel E_I)}(\text{ID}_A \parallel R'_I \parallel H_2(b_I)) \). Then \( R_{RC} = R_{RC} \cdot P, K_I = R_{RC} + H_2(b_I) \cdot P, L_I = H_1(\text{ID}_A \parallel D^{\text{new}}_I \parallel \text{SID}_J \parallel E_I) \), where \( r_{RC} \) is a random number. RC sends \( \{D^{\text{new}}_I, K_I, L_I, F_a\} \) to \( S_j \).

**Step 3.** Upon receiving \( \{D^{\text{new}}_I, K_I, L_I, F_a\} \), \( S_j \) computes \( R_S = r_S \cdot P, Q_a = Q_a + r_S, M_i = H_1(D^{\text{new}}_I \parallel R_S \parallel H_2(b_I) \cdot P) \). where \( r_S \) is a random number. \( S_j \) sends \( \{D^{\text{new}}_I, Q_a, M_i, K_I\} \) to \( \mathcal{A} \).

**Step 4.** Upon receiving \( \{D^{\text{new}}_I, Q_a, M_i, K_I\} \), \( \mathcal{A} \) computes \( R'_S = Q_a - r_S, R''_C = K_I - B_I, S_K = H_1(R_S \parallel R''_C \parallel \text{ID}_A \parallel \text{SID}_J \parallel E_I) \) and \( Z_a = S_K \cdot P + H_1(\text{ID}_A \parallel H_2(b_I)) \cdot P \) and sends \( \{Z_a\} \) to \( S_j \).

**Step 5.** Upon receiving \( \{Z_a\} \), \( S_j \) computes \( S_K = H_1(R''_C \parallel \text{ID}_A \parallel \text{SID}_J \parallel E_I) \) and \( Z'_a = S_K \cdot P + H_1(\text{ID}_A \parallel H_2(b_I)) \cdot P \). Obviously \( Z'_a = Z_a \). \( S_j \) regards \( \mathcal{A} \) as legitimate user \( U_I \).

### 3.3. Server Impersonation Attack

The adversary \( \mathcal{A} \) obtains \( U_I \)'s biometric \( b_I \) and intercepts \( \{\text{ID}_I, E_I, C_A, D_I\} \) from public channel. Afterwards, \( \mathcal{A} \) performs server impersonation attack in the following steps.

**Step 1.** \( \mathcal{A} \) chooses two random numbers \( r_{S_a}, r_{RC_a} \) and computes \( R_{S_a} = r_{S_a} \cdot P, R_{RC_a} = r_{RC_a} \cdot P, B_I = H_2(b_I) \cdot P, R'_I = C_I - B_I, Q_a = R_{S_a} + R'_I, K_a = R_{RC_a} + B_I, M_a = H_1(D^{\text{new}}_I \parallel R_{S_a} \parallel R_{RC_a} \parallel B_I) \), where \( D^{\text{new}}_I \) is a random binary string whose length is equal with \( \text{ID}_A \). \( \mathcal{A} \) sends \( \{D^{\text{new}}_I, Q_a, M_a, K_a\} \) to \( U_I \).

**Step 2.** Upon receiving \( \{D^{\text{new}}_I, Q_a, M_a, K_a\} \), \( U_I \) computes \( S_K = H_1(R_{S_a} \parallel R_{RC_a} \parallel B_I) \) and \( Z_a = S_K \cdot P + H_1(\text{ID}_A \parallel \text{SID}_J \parallel E_I) \). Therefore, \( \mathcal{A} \) impersonates \( U_I \) successfully.
When the smart card and biometric of user are compromised, the attacker is able to breach the password. On the other hand, \(a\) is able to impersonate user successfully as long as he gets the biometric of user. Ali et al's scheme fails to achieve three-factor secrecy.

### 4. The Proposed Scheme

In this section, we present a biometric-based remote user authentication scheme for multiserver environment. The proposed scheme includes the following five phases.

#### 4.1. Initial Phase

RC chooses an elliptic curve group \(E_q\) of order \(p\) and a generator \(P\) of \(E_q\). RC generates a random number \(x\) and computes \(P_{pub} = xP\). RC publishes \(\{E_q, P, P_{pub}, H_1(), H_2()\}\) and keeps \(x\) as secret.

#### 4.2. Server Registration Phase

The server \(S_j\) registers with RC in the following steps.

**Step 1.** \(S_j\) picks its identity \(SID_j\) freely and delivers \(\{SID_j\}\) to RC through the reliable channel.

**Step 2.** Upon receiving \(\{SID_j\}\), RC calculates \(SM_j = H_j(SID_j \parallel x)\) and returns \(SM_j\) to \(S_j\) via the reliable channel.

**Step 3.** \(S_j\) keeps \(SM_j\) as secret.

#### 4.3. User Registration Phase

The user \(U_i\) registers with RC in the following steps. As described in Figure 2.

**Step 1.** \(U_i\) chooses its identity \(ID_i\) and password \(PW_i\) freely and imprints his biometric \(b_i\). \(U_i\) chooses his identity \(ID_i\) and password \(PW_i\) freely and imprints his biometric \(b_i\), \(U_i\) calculates \(P_i = H_i(PW_i \parallel H_2(b_i) \parallel r_i)\), where \(r_i\) is a random number. Afterwards, \(\{ID_i, P_i\}\) is transmitted to RC through the reliable channel.

**Step 2.** Upon receiving \(\{ID_i, P_i\}\), RC computes \(A_i = H_i(x \parallel ID_i)\), \(B_i = A_i \oplus P_i\), \(V_i = H_i(P_i \oplus H_1(ID_i))\mod n\), where \(2^k \leq n \leq 2^k\). RC stores \(\{B_i, V_i, E_{Key}(), P_i, P_{pub}, n\}\) in a smart card and transmits it to \(U_i\) via the reliable communication channel.

**Step 3.** \(U_i\) stores \(r_i\) in the smart card.

#### 4.4. Login and Authentication Phase

The user \(U_i\) and the server \(S_j\) authenticate each other and establish a session key by the aide of RC in the following steps. As shown in Figure 3.

**Step 1.** \(U_i\) attaches the smart card to a terminal, enters \(ID_i^*\) and \(PW_i^*\), and imprints \(b_i^*\). Then the smart card calculates \(P_i^* = H_i(PW_i^* \parallel H_2(b_i^*) \parallel r_i)\), \(V_i^* = H_i(P_i^* \oplus H_2(ID_i^*))\mod n\) and checks if \(V_i^* = V_i\). If this equation holds, the smart card computes \(A_i^* = B_i \oplus P_i^*\), \(R_i = N_i P, C_i = H_i(N_i P_{pub}), E_i = E_{C_i}(ID_i^* \parallel A_i^* \parallel SID_i)\), where \(N_i\) is a random number. \(\{R_i, L_i\}\) is transmitted to RC via the public channel.

**Step 2.** After receiving \(\{R_i, L_i\}\), RC computes \(C_i^* = H_i(xR_i)\), \(ID_i^* \parallel A_i^* \parallel SID_i\) = \(D_{C_i^*}(L_i)\), \(A_i = H_i(x \parallel ID_i)\) and...
checks if $A_1 = A'_1$. If the equation holds, $RC$ computes $SM_j = H_1(SID'_j \parallel x), Y_i = H_1(SID'_j \parallel SM_j), M_4 = E_{SM_j}(ID'_j \parallel R_i \parallel Y_i \parallel H_1(A_1 \parallel C_1'))$. \{M_4\} is transmitted to $S_j$.

Step 3. After receiving \{M_4\}, $S_j$ computes $(ID''_i \parallel R''_i \parallel Y''_i \parallel H_1(A_1 \parallel C_1')) = D_{SM_j}(M_4), Y''_i = H_1(SID' \parallel SM_j)$ and checks if $Y''_i$ is equal to $Y'_i$. If it holds, $S_j$ computes $R_5 = N_2 \cdot P, E_5 = N_2 \cdot R_5, SK = H_1(E_i \parallel H_1(A_1 \parallel C_1')), F_5 = H_1(ID''_i \parallel SK \parallel R_5 \parallel SID_j)$, where $N_2$ is a random number. \{R_5, F_5\} is transmitted to $U_i$.

Step 4. After receiving \{R_5, F_5\}, $U_i$ computes $E'_i = N_1 \cdot R_5, SK = H_1(E'_i \parallel H_1(A'_1 \parallel C_1')), E'_i = H_1(ID''_i \parallel SK \parallel R_5 \parallel SID_j)$ and checks if $F'_5 = F_5$. If the equation holds, $U_i$ computes $Q_i = H_1(SK \parallel R_5)$ and sends \{Q_i\} to $S_j$.

Step 5. After receiving \{Q_i\}, $S_j$ computes $Q'_i = H_1(SK \parallel R_5)$ and checks if $Q'_i = Q_i$. If the equation holds, $S_j$ establishes a session key $SK$ with $U_i$ successfully.

4.5. Password Update Phase. $U_i$ changes his original password to a new one in the following steps. As described in Figure 4.

Step 1. $U_i$ attaches his smart card to a terminal, enters $ID'_i$ and $PW'_i$, and imprints $b'_i$. The smart card calculates $P'_i = H_1(PW'_i \parallel H_1(b'_i) \parallel r_1), Y'_i = H_1(P'_i \parallel H_1(ID'_i)) \mod n$. 

Figure 2: User registration phase of the proposed scheme.

Figure 3: Login and authentication phase of the proposed scheme.
and checks $V_1^* = V_1$. If it holds, the smart card asks the user to input a new password.

Step 2. $U_i$ enters his new password $PW_{\text{new}}$. Then the smart card calculates $P_i^* = H_1(PW_{\text{new}} || H_2(b_i) || r_i)$, $B_i' = B_i \oplus B_i^* \oplus P_i^*$, $V_i^* = H_1(P_i^* \oplus H_2(U_i')) \mod n$. The smart card stores $B_i^*, V_i^*$ in the smart card and removes $B_i, V_i$.

5. Security Proof

5.1. Formal Security Analysis. We describe the formal security model for three-factor multiserver authentication schemes proposed by Feng et al. [26] and prove the proposed scheme is provably secure in this model.

5.1.1. Security Model

Participants. There are three types of principals in multiserver authentication scheme, that is, the user $U_i$, the server $S_j$, and the registration center $RC$. Every kind of participant has many instances. We use $U_i^n$, $S_j^a$, and $RC^a$ denote them.

Queries. The abilities of adversary are modeled by asking the following queries.

Execute ($U_i^n, S_j^a, RC^a$). The query simulates the eavesdropping attack. It returns the transcripts of the transmitted messages in public channel to the adversary.

Send ($U_i^n, S_j^a, RC^a, m$). It allows the adversary masquerades as a principal to send a message $m$. The oracle handles the message and gives a response to the adversary.

Reveal ($U_i^n, S_j^a$). This query discloses the session key of instance $U_i^n$ or $S_j^a$ to the adversary. However, if instance $U_i^n$ or $S_j^a$ does not establish a session key, it returns an invalid symbol $\perp$.

Corrupt ($U_i^n, z$). This query reveals one or two authentication factors of user to the adversary. Note that the adversary cannot get all the three authentication factors at the same time, as he has no difference with a legitimate user.

When $z = 1$, it returns the password of $U_i^n$ to the adversary.

When $z = 2$, it returns the data in $U_i^n$’s smart card.

When $z = 3$, it returns the biometric of $U_i^n$.

Corrupt ($S_j^a, RC^a$). This query simulates the forward secrecy attack; it answers the master key $x$ or the secret key $SM_i$ to the adversary.

Test ($U_i^n, S_j^a$). The query is used to evaluate the semantic security of session key. The adversary is allowed to make the query no more than once. If the instance $U_i^n$ or $S_j^a$ is fresh (see below), the oracle flips a coin $b$. If $b = 1$, it returns the session key to the adversary. If $b = 0$, it returns a random string of the same size to the adversary.

Freshness. The instance $U_i^n$ or $S_j^a$ is fresh, if the following conditions are satisfied.

1. (1) The instance is accepted and establishes a session key.
2. (2) The instance and its partner that belongs to the same session are never made a reveal query.
3. (3) The adversary never asks the Corrupt ($U_i^n, z = 1, 2, 3$) query.
4. (4) The adversary never makes a Corrupt ($S_j^a, RC^a$) query.

Semantic Security. The adversary makes a series of aforementioned queries in polynomial time. Eventually, the adversary deduces the value of $b$ involved in test query to be $b'$. We denote the advantage that the adversary breaches the semantic security of our scheme as

$$Adv_{\mathcal{P}}^{\text{sk}}(\mathcal{A}) = 2 \Pr(b' = b) - 1$$

(1)

Our protocol is secure, if for any adversary the advantage is negligible.

5.1.2. Formal Security Proof. The formal security proof of the proposed scheme relies on the presumed hardness of the elliptic curve Diffie–Hellman problem defined below.

The Elliptic Curve Diffie–Hellman Problem (ECDHP). Let $E_q$ be an elliptic curve group of order $p$. And $P$ is a generator of $E_q$. For given $R_1, R_2 \in E_p$, where $R_1 = N_1P, R_2 = N_2P$, it is infeasible to compute $N_1N_2P$ in polynomial time.

Theorem 1. We use $P$ to denote the proposed scheme. There is an adversary $\mathcal{A}$ who tries to break the semantic security of our scheme. We assume that $\mathcal{A}$ is able to make at most $q_i$ Send-queries, $q_e$ Execute queries, $q_h$ Hash queries, $q_b$ Bio-hash queries, and $q_e$ Encryption/Decryption queries in polynomial time $t$. Then we have

$$Adv_{\mathcal{P}}^{\text{sk}}(\mathcal{A}) \leq \frac{q_i^2}{2^{\ell_1}} + \frac{q_h^2 + q_e + 2q_s}{2^{\ell_2}} + \frac{q_b}{2^{\ell_3}} + \frac{(q_h + q_e)^2}{p}$$

$$+ \frac{q_h}{y} + 2q_i \cdot Adv_{\mathcal{P}}^{ECDHP}$$

(2)
where $l_1$ is the bit length of hash output, $l_2$ is the bit length of Bio-hash output, $l_3$ is the bit length of symmetric encryption output. The password dictionary space is $Y$. $\text{Adv}_{P}^{\text{ECDHP}}$ is the probability that the adversary solves the ECDHP in polynomial time $t$.

The Proof: The advantage of breaking our scheme is deduced via a series of games from $G_0$ to $G_7$. $S_i$ denotes the event that the adversary correctly guesses the value of $b$ involved in test query in game $G_i$. And $\Pr[S_i]$ is the probability of the event $S_i$.

$G_0$: it represents the real attack; obviously, we have

$$\text{Adv}_{P}^{\text{ECDHP}} (\mathcal{A}) = 2(\Pr[S_0]) - 1 \quad (3)$$

By a further transformation, we have

$$2(\Pr[S_0]) - 1 = 2(\Pr[S_0] - \Pr[S_0]) + 2\Pr[S_0] - 1 = 2\Pr[S_0] - 1 + 2\sum_{i=0}^{5}(\Pr[S_i] - \Pr[S_{i+1}]) \quad (4)$$

$G_1$: in this game, the hash oracle, bio-hash oracle, and encryption/decryption oracle are simulated by maintaining a hash list $\Lambda_H$, a bio-hash list $\Lambda_{bio}$, and an encryption/decryption list $\Lambda$. For a hash query $H_i(\alpha)$, if there is an item $(\alpha, \beta)$ in $\Lambda_H$, the adversary returns $\beta$ to the adversary. Otherwise, the oracle chooses a random number $\beta$ returns $\beta$ to the adversary, and adds the item $(\alpha, \beta)$ to $\Lambda_H$. The bio-hash oracle is simulated in the same way. For an encryption query $E_i(str)$, if there is an item $(k, str, y)$ in $\Lambda$, the oracle returns $y$ to the adversary. Otherwise, the oracle chooses a value $y$ from cipher text space, returns $y$ to the adversary, and adds the item $(k, str, y)$ to $\Lambda$. For a decryption query $D_i(y)$, if there is an item $(k, str, y)$ in $\Lambda$, the oracle returns $str$ to the adversary. Otherwise, the oracle chooses a value $str$ from plaintext space, returns $str$ to the adversary, and adds the item $(k, str, y)$ to $\Lambda$. Besides, all oracles involved in security model are simulated in this game. Obviously, this game has no difference with $G_0$. We have

$$\Pr[S_0] - \Pr[S_1] = 0 \quad (5)$$

$G_2$: we avoid the occurrence of some collisions in this game. $G_2$ is indistinguishable from $G_1$, unless the following conditions occur.

1. A collision happens in the output of hash function; the probability is less than $q_h^2/2^{l_1+1}$.
2. A collision happens in the output of bio-hash; the probability is no more than $q_h^2/2^{l_1+1}$.
3. A collision happens in the output of symmetric encryption; the probability is less than $q_e^2/2^{l_2+1}$.
4. A collision happens on $R_i$ or $R_{5}$, the probability is no more than $(q_s + q_{e})^2/2p$.

So we have

$$\Pr[S_1] - \Pr[S_2] \leq \frac{q_h^2}{2^{l_1+1}} + \frac{q_h^2}{2^{l_1+1}} + \frac{q_e^2}{2^{l_2+1}} + \frac{(q_s + q_e)^2}{2p} \quad (6)$$

$G_3$: in this game, we avoid the situation that the adversary correctly guesses $F_1$ or $Q_1$ without making the corresponding hash query. The probability is at most $q_s/2^l$. Thus,

$$\Pr[S_2] - \Pr[S_3] \leq \frac{q_s}{2^l} \quad (7)$$

$G_4$: this game averts the execution when the adversary correctly guesses the authentication value $A_i$ directly. The probability is at most $q_s/2^l$. We get

$$\Pr[S_3] - \Pr[S_4] \leq \frac{q_s}{2^l} \quad (8)$$

$G_5$: in this game, we avoid the occurrence that the adversary has computed the authentication value $A_i$ with the help of Corrupt $(U^a, z)$. The following three cases are included.

Case 1. The adversary queries Corrupt $(U^a, 1)$ and Corrupt $(U^a, 2)$. To derive $A_i$, the adversary still needs to get the biometric. The probability that he correctly guesses the biometric is at most $q_s/2^l$.

Case 2. The adversary queries Corrupt $(U^a, 2)$ and Corrupt $(U^a, 3)$. The probability that he correctly guesses the password is less than $q_s/2^l$.

Case 3. The adversary queries Corrupt $(U^a, 1)$ and Corrupt $(U^a, 3)$. The probability that he correctly guesses the parameter $B_i$ is no more than $q_s/2^l$.

The probability that the adversary gets $A_i$ is less than $q_s \ast (1/Y + 1/2^l + 1/2^l)$. We have

$$\Pr[S_4] - \Pr[S_5] \leq q_s \ast \left( \frac{1}{Y} + \frac{1}{2^l} + \frac{1}{2^l} \right) \quad (9)$$

$G_5$: in this game, we compute the session key $SK$ using the private oracles $H'_1$ instead of the hash oracle $H_1$. As the private oracles $H_1'$ is unknown to the adversary. We have

$$\Pr[S_6] = \frac{1}{2} \quad (10)$$

$G_6$: has no difference with $G_5$, unless the adversary makes a hash query $H_i(E_i \parallel H_1(A_i \parallel C_i))$; we denote the event as $\Lambda_1$. We have

$$\Pr[S_5] - \Pr[S_6] \leq \Pr[\Lambda_1] \quad (11)$$

$G_7$: we simulate the random self-reducibility of ECDHP in this game. For $R_i = N_i P, R_S = N_2 P$, through selecting randomly in $\Lambda_H$, we can obtain the item containing $E_i = N_1 N_2 P$ with the probability $1/q_h$. Since the event $\Lambda_1$ denotes that the adversary makes a hash query $H_i( E_i \parallel H_1(A_i \parallel C_i))$. We have

$$\Pr[\Lambda_1] \leq q_h \text{Adv}_{P}^{\text{ECDHP}} \quad (12)$$
Table 2: The notations and rules of BAN logic.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P, Q$</td>
<td>A principal</td>
</tr>
<tr>
<td>$X, Y$</td>
<td>A statement</td>
</tr>
<tr>
<td>$K$</td>
<td>A key</td>
</tr>
<tr>
<td>$P \triangleleft X$</td>
<td>$P$ sees $X$, $P$ receives a message containing $X$</td>
</tr>
<tr>
<td>$P</td>
<td>\sim X$</td>
</tr>
<tr>
<td>$P\equiv Q$</td>
<td>$P$ believes $X$ is true</td>
</tr>
<tr>
<td>$P K$</td>
<td>$P$ and $Q$ share a secret $Y$</td>
</tr>
<tr>
<td>${X}_K$</td>
<td>$X$ is fresh</td>
</tr>
<tr>
<td>$&lt;X&gt;$</td>
<td>$X$ is combined with a secret $Y$</td>
</tr>
<tr>
<td>$P \equiv Q \Rightarrow X$</td>
<td>$P$ has jurisdiction over $X$</td>
</tr>
</tbody>
</table>

Message meaning rule

Belief rule

Nonce-verification rule

Jurisdiction rule

Through the series of games above, we have

$$Ad_{\mathcal{P}}^{^{se}}(\mathcal{A}) \leq \frac{q_s^2 + 6q_t}{2^h} + \frac{q_s^2 + 2q_t}{2^h} + \frac{(q_s + q_t)^2}{p} + \frac{2q_t}{Y} + 2q_\rho Ad_{\mathcal{P}}^{^{ECDHP}}$$

5.2. Security Proof Using BAN Logic. In this section, we use BAN logic [33] to prove that our scheme achieves mutual authentication and establishes a secure session key. Table 2 describes the symbols and rules of BAN logic.

The goals that our scheme should achieve are as follows.

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

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

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

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

We idealized the proposed scheme as follows.

M1: $U_i \rightarrow RC < N_1 P, U_i \overset{C_i}{\leftrightarrow} RC > U_i \overset{N_1}{\leftrightarrow} RC$

M2: $RC \rightarrow S_j \{N_1 P, U_i \overset{H(A,K_C)}{\leftrightarrow} S_j\}_{SM_j}$

M3: $S_j \rightarrow U_i < N_1 R_s, R_s, U_i \overset{SK}{\rightarrow} S_j \overset{H(A,K_C)}{\leftrightarrow}$

M4: $S_j \rightarrow U_i < N_2 P, U_i \overset{SK}{\rightarrow} S_j \overset{H(A,K_C)}{\leftrightarrow}$

The initiative assumption of our scheme is given as follows.

S1: $RC | \equiv U_i \overset{A_i}{\leftrightarrow} RC$

S2: $RC \equiv \#(N_1)$

S3: $RC \equiv (U_i \overset{C_i}{\leftrightarrow} RC)$

S4: $S_j | \equiv RC \overset{SM_j}{\leftrightarrow} S_j$

S5: $S_j | \equiv \#(N_1)$

S6: $S_j | \equiv RC | \rightarrow (U_i \overset{H(A,K_C)}{\leftrightarrow} S_j)$

S7: $U_i | \equiv U_i \overset{H(A,K_C)}{\leftrightarrow} S_j$

S8: $U_i | \equiv \#(N_2)$

S9: $U_i | \equiv S_j \rightarrow (U_i \overset{SK}{\leftrightarrow} S_j)$

S10: $S_j | \equiv \#(N_2)$

S11: $S_j | \equiv U_i \rightarrow (U_i \overset{SK}{\leftrightarrow} S_j)$

The proof of our scheme is performed as follows.

From M1, we have

(1) $RC \triangleleft < N_1 P, U_i \overset{C_i}{\leftrightarrow} RC > U_i \overset{N_1}{\leftrightarrow} RC$

According to S1, (1) and message meaning rule, we obtain

(2) $RC | \equiv U_i | \sim < N_1 P, U_i \overset{C_i}{\leftrightarrow} RC >$

According to S2, (2) and nonce-verification rule, we obtain
(3) \( RC \equiv U_i \equiv < N_i, P, U_i, C_i > \)

According to S3, (3) and jurisdiction rule, we obtain

(4) \( RC \equiv U_i \equiv RC \)

From M2, we have

(5) \( S_j \equiv [N_1, P, U_i, H_{(A_i, C_i)}] \equiv S_j \)

According to S4, (5) and message meaning rule, we obtain

(6) \( S_j \equiv RC \sim \{ N_1, P, U_i, H_{(A_i, C_i)} \} \equiv S_j \)

According to S5, (6) and nonce-verification rule, we obtain

(7) \( S_j \equiv RC \equiv \{ N_1, P, U_i, H_{(A_i, C_i)} \} \equiv S_j \)

According to S6, (7) and jurisdiction rule, we obtain

(8) \( S_j \equiv U_i \equiv H_{(A_i, C_i)} \)

From M3, we have

(9) \( U_i \equiv < N_1, R_S, P, U_i \equiv R_C \equiv S_j \)

According to S7, (9) and message meaning rule, we obtain

(10) \( U_i \equiv S_j \equiv < N_1, R_S, R_S, U_i \equiv R_C \equiv S_j > \)

According to S8, (10) and nonce-verification rule, we obtain

(11) \( U_i \equiv S_j \equiv U_i \equiv R_C \equiv S_j (\text{Goal 1}) \)

According to S9, (11) and jurisdiction rule, we obtain

(12) \( U_i \equiv U_i \equiv R_C \equiv S_j (\text{Goal 2}) \)

From M4, we have

(13) \( S_j \equiv < N_2, P, U_i \equiv R_C \equiv S_j > H_{(A_i, C_i)} \)

According to (8), (13) and message meaning rule, we obtain

(14) \( S_j \equiv U_i \equiv < N_2, P, U_i \equiv R_C \equiv S_j > \)

According to S10, (14) and nonce-verification rule, we obtain

(15) \( S_j \equiv U_i \equiv U_i \equiv R_C \equiv S_j (\text{Goal 3}) \)

According to S11, (15) and jurisdiction rule, we obtain

(16) \( S_j \equiv U_i \equiv R_C \equiv S_j (\text{Goal 4}) \)

### 6. Informal Security Analysis

In this section, we demonstrate that our scheme achieves user anonymity, forward secrecy, and three-factor secrecy and is resistant to several known attacks.

#### 6.1. User Anonymity

In our scheme, user’s identity \( ID_i \) is protected with symmetric encryption. As the key \( C_i \) and \( SM_i \) is unavailable. \( \mathcal{A} \) cannot get any information about \( ID_i \) from the transmitted messages in public channel. In addition, \( \mathcal{A} \) cannot link two distinct messages to one user due to the existence of random number. Our scheme achieves user anonymity.

#### 6.2. Forward Secrecy

Suppose that \( \mathcal{A} \) compromises the master key of \( RC \) and intercepts \( \{ R_i, L_i \}, \{ R_S, E \} \) from public channel. Then \( \mathcal{A} \) tries to compute the session key \( SK = H_1(E_i || H_1(A_i || C_i)) \). \( \mathcal{A} \) can get \( C_i \) and \( A_i \) by computing \( C_i = H_1(xR_i), (ID_i^{A_i} || A_i^{C_i} || SID) = D_{C_i}(L_i) \). To get \( E_i \), \( \mathcal{A} \) needs to derive \( E_i \) from \( R_i, R_S \). It means that \( \mathcal{A} \) has to solve the elliptic curve Diffie–Hellman problem. It is absolutely impossible. Our scheme achieves forward secrecy.

#### 6.3. Offline Password Guessing Attack

In the case that \( \mathcal{A} \) extracts \( \{ B_i, V_i, E_{Key} \}, P, P_{pub}, n, r_i \) from \( U_i \)’s smart card and obtains \( U_i \)’s biometric \( b_i \), \( \mathcal{A} \) tries to acquire the password of \( U_i \) in the following steps.

**Step 1.** Choose an identity \( ID_i^* \) from identity dictionary space and a password \( PW_i^* \) from password dictionary space.

**Step 2.** Compute \( P_i^* = H_1(PW_i^* || H_2(b_i^*) || r_i), V_i^* = H_1(P_i^* \oplus H_1(ID_i^*)) \mod n \). Check \( V_i^* = \tilde{V}_i \).

**Step 3.** Repeat Steps 1 and 2, until \( \mathcal{A} \) finds a pair of \( < ID_i^*, PW_i^* > \) satisfying \( V_i^* = \tilde{V}_i \).

However, even if \( \mathcal{A} \) finds a pair of \( < ID_i^*, PW_i^* > \) satisfying \( V_i^* = \tilde{V}_i \), he cannot determine whether they are the real identity and password of \( U_i \). The proposed scheme employs the fuzzy validation of inputted authentication information. When \( n = 2^8 \) and the identity and password both are 64 bits, there will be \( (2^{64} * 2^{64})/(2^8) \) pairs of identity and password satisfying \( V_i^* = \tilde{V}_i \). The probability that each candidate is equal to the pair of identity and password of \( U_i \) is \( 2^8 / (2^{64} * 2^{64}) \), this is negligible. In our scheme, it is unable to reveal the identity and password of user even if both the smart card and biometric are compromised.

#### 6.4. User Impersonation Attack

Assume that \( \mathcal{A} \) tries to impersonate user and forge a login requested message \( \{ R_i, L_i \} \). \( \mathcal{A} \) computes \( R_i = N_i \cdot P \), where \( N_i \) is a random number. To compute \( L_i \), \( \mathcal{A} \) needs to know \( A_i \). However, \( \mathcal{A} \) cannot get any information about \( A_i \) from the transmitted messages in public channel, as \( A_i \) is protected with symmetric encryption and hash function. In the case that the smart card is compromised, \( \mathcal{A} \) tries to retrieve \( A_i \) from \( B_i \). As \( A_i = B_i \oplus P_i \), \( \mathcal{A} \) needs to get \( P_i \) at first. To compute \( P_i \), \( \mathcal{A} \) requires \( r_i, b_i, PW_i \). That is to say, \( \mathcal{A} \) cannot get \( A_i \), unless he obtains all the three authentication factors at the same time. This is beyond the capacity of \( \mathcal{A} \). The proposed scheme is secure against user impersonate attack.

#### 6.5. Server Impersonation Attack

Suppose that \( \mathcal{A} \) intercepts \( \{ R_i, L_i \} \) and \( \{ M_i \} \) from public channel and tries to masquerade...
as the server $S_j$ by sending a forged message $\{R_o, F_j\}$ to $U_i$. At first, $\mathcal{A}$ generates a random number $N_2$ and computes $R_o = N_2 \cdot P_i$. Next, to compute $\tilde{S}K = H_1(E_i \| H_1(A_i \| C_j))$, $\mathcal{A}$ still needs $A_i$ and $C_i$. As analyzed above, $\mathcal{A}$ cannot obtain $A_i$. To derive $C_j$, the adversary needs to compromise the master key $x$ or break the elliptic curve Diffie–Hellman problem. It is beyond the capacity of $\mathcal{A}$. Our scheme is secure against server impersonation attack.

6.6. Replay Attack. In our scheme, we adopt random numbers instead of timestamp to guarantee the freshness of exchanged messages. It decreases the communication overhead and avoids clock synchronization problem. In the following four cases, we demonstrate that our scheme is resistant to replay attack.

Case 1. Suppose that the adversary $\mathcal{A}$ intercepts $\{R_o, L_i\}$ from public channel and sends it to $RC$ as a new login request. $RC$ and $S_j$ deal with this message and return $\{M_i\}$ to $\mathcal{A}$. Then $\mathcal{A}$ needs to generate a response message $\{Q\}$ and sends it to $S_j$. As $N_1, A_i$ are unavailable, the adversary is unable to return a valid $\{Q\}$ to $S_j$. The protocol finally aborts.

Case 2. In case $\mathcal{A}$ replays $\{M_i\}$ to $S_j$, as $\mathcal{A}$ is unable to return a valid $\{Q\}$ to $S_j$, the protocol finally aborts.

Case 3. If $\mathcal{A}$ intercepts $\{M_i\}$ from public channel and replays $\{R_o, F_j\}$ to $U_i$. The user deals with this message and finds that $F_j \neq F_i$. The protocol aborts.

Case 4. Assume that $\mathcal{A}$ intercepts $\{R_o, F_i\}$ from public channel and replays $\{Q\}$ to $S_j$. The server deals with this message and finds that $Q_j' \neq Q_j$. The protocol aborts.

6.7. Known Session-Specific Temporary Information Attack. Suppose that random number $N_1$ or $N_2$ is compromised; the adversary computes $E_i = N_1 \cdot R_2$ or $E_i = N_2 \cdot R_1$. It still requires $A_i$ and $C_i$ to compute the session key $SK$. However, $A_i, C_i$ are unavailable. It is unable to compromise the session key in our scheme.

6.8. Privileged Insider Attack. On one hand, the password and biometric of $U_i$ are protected with hash function in registration phase. On the other hand, the user never reveals any authentication information (password, biometric, or the parameters of smart card) to $RC$ or server in login and authentication phase. Hence, our scheme is resistant to privileged insider attack.

6.9. Three-Factor Secrecy. As analyzed above, in the absence of any one authentication factor, $\mathcal{A}$ cannot impersonate user successfully. In the following three cases, we demonstrate that if any two authentication factors of user are compromised, the adversary cannot breach the other one.

Case 1. Suppose that $U_i$’s smart card and biometric are compromised. As analyzed above, the adversary cannot reveal $U_i$’s password via offline password guessing attack.

Case 2. Suppose that $U_i$’s smart card and password are compromised. As the biometric $b_i$ is protected by means of hash function, $\mathcal{A}$ is unable to retrieve $b_i$ from $V_i$.

Case 3. Suppose that $U_i$’s biometric and password are compromised. The adversary tries to reveal the parameters $\{b_i, V_i, r_i\}$ stored in the smart card, where $r_i$ is a random number, $B_i = A_i \oplus P_i$, $P_i = H_1(PW_i \| H_2(b_i) \| r_i)$, $V_i = H_1(P_i \oplus H_2(ID)) \mod n$. As $r_i, A_i$ are unavailable. The adversary is unable to reveal any data of smart card.

7. Security and Performance Comparison

We compare our scheme with other biometric-based multi-server authentication schemes using ECC [22, 25–27]. The results of comparison indicate that our scheme satisfies all the security requirements, while it requires the minimum communication and computation overhead.

Table 3 shows the results of security analysis. It indicates that only our scheme is secure against various known attacks and provides desirable security properties such as forward secrecy, user anonymity, three-factor secrecy, and efficient wrong password and biometric detection. The other schemes [22, 25–27] suffer from more or less security vulnerabilities.

Table 4 gives the computation costs of related schemes at login and authentication phase. More specifically, $T_H$ denotes computing a hash function. $T_E$ denotes one symmetric encryption. $T_D$ denotes one symmetric decryption. $T_p$ denotes one point multiplication on elliptic curve group. The computing overhead of lightweight operation “XOR” is negligible compared with other operations. Our scheme requires $3T_p + 1T_E + 7T_H$, in user end, requires $1T_p + 1T_D + 1T_E + 5T_H$ in $RC$, and requires $2T_p + 1T_D + 4T_H$ in server end. And the total computation cost of our scheme is $6T_p + 2T_D + 2T_E + 16T_H$. The total computation costs of related schemes [22, 25–27] are $8T_p + 22T_D, 8T_p + 16T_D, 8T_p + 24T_D, 7T_p + 2T_D + 2T_E + 12T_H$, respectively.

Table 5 summarizes the computing time of different cryptographic operations [34]. The hash function SHA-256 and SHA-512, the symmetric algorithm AES-128 and AES-256, the elliptic curve cryptosystem using P521, and Curve25519, respectively, are employed to estimate the running time of related schemes. We compare our scheme with related schemes for two scenarios as shown in Figure 5. The Scenario I adopts the comparably efficient algorithms, that is, SHA-256, AES-128 encryption/decryption, and elliptic curve cryptosystem using P521, and Curve25519, respectively, are employed to estimate the running time of related schemes. We compare our scheme with related schemes for two scenarios as shown in Figure 5.

Figure 6 illustrates the communication overheads of related schemes. To evaluate the communication overhead,
Table 3: Results of security analysis of related schemes.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Resist offline password guessing attack</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>User anonymity</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Resist denial of service attack</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>❌</td>
<td>✔</td>
</tr>
<tr>
<td>Resist user impersonation attack</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>❌</td>
<td>✔</td>
</tr>
<tr>
<td>Resist server impersonation attack</td>
<td>✔</td>
<td>❌</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Forward secrecy</td>
<td>✔</td>
<td>✔</td>
<td>❌</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Resist known session-specific temporary information attack</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Efficient wrong password and biometric detection</td>
<td>❌</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Resist privileged insider attack</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>❌</td>
<td>✔</td>
</tr>
<tr>
<td>Three-factor secrecy</td>
<td>✔</td>
<td>❌</td>
<td>❌</td>
<td>❌</td>
<td>✔</td>
</tr>
</tbody>
</table>
Table 4: Computation cost of related schemes.

<table>
<thead>
<tr>
<th>Schemes</th>
<th>User</th>
<th>RC</th>
<th>Server</th>
<th>Total cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>He [22]</td>
<td>$3T_p + 7T_H$</td>
<td>$2T_p + 10T_H$</td>
<td>$3T_p + 5T_H$</td>
<td>$8T_p + 22T_H$</td>
</tr>
<tr>
<td>Kumari [25]</td>
<td>$3T_p + 5T_H$</td>
<td>$2T_p + 6T_H$</td>
<td>$3T_p + 5T_H$</td>
<td>$8T_p + 16T_H$</td>
</tr>
<tr>
<td>Feng [26]</td>
<td>$3T_p + 7T_H$</td>
<td>$2T_p + 10T_H$</td>
<td>$3T_p + 7T_H$</td>
<td>$8T_p + 24T_H$</td>
</tr>
<tr>
<td>Ali [27]</td>
<td>$3T_p + 5T_H$</td>
<td>$1T_p + 1T_D + 2T_E + 3T_H$</td>
<td>$3T_p + 1T_D + 4T_H$</td>
<td>$7T_p + 2T_D + 2T_E + 12T_H$</td>
</tr>
<tr>
<td>Our protocol</td>
<td>$3T_p + 1T_E + 7T_H$</td>
<td>$1T_p + 1T_D + 1T_E + 5T_H$</td>
<td>$2T_p + 1T_D + 4T_H$</td>
<td>$6T_p + 2T_D + 2T_E + 16T_H$</td>
</tr>
</tbody>
</table>

Table 5: Computing time of cryptographic operations.

<table>
<thead>
<tr>
<th>Operations</th>
<th>Computing time</th>
</tr>
</thead>
<tbody>
<tr>
<td>SHA-256</td>
<td>13 $\mu$s</td>
</tr>
<tr>
<td>SHA-512</td>
<td>32 $\mu$s</td>
</tr>
<tr>
<td>AES-128 Encryption</td>
<td>64 $\mu$s</td>
</tr>
<tr>
<td>AES-128 Decryption</td>
<td>117 $\mu$s</td>
</tr>
<tr>
<td>AES-256 Encryption</td>
<td>90 $\mu$s</td>
</tr>
<tr>
<td>AES-256 Decryption</td>
<td>165 $\mu$s</td>
</tr>
<tr>
<td>Curve25519 Point Multiplication</td>
<td>72 ms</td>
</tr>
<tr>
<td>P521 Point Multiplication</td>
<td>1053 ms</td>
</tr>
</tbody>
</table>

Figure 5: Running time of related schemes.

![Running time of related schemes](image1.png)

Figure 6: Communication overhead of related schemes.

![Communication overhead of related schemes](image2.png)

8. Conclusions

In this paper, we prove that Ali et al.'s scheme is susceptible to various security threats, such as impersonation attack, denial of service attack, and known session-specific temporary information attack. Furthermore, we propose an efficient ECC-based three-factor authentication scheme for multiserver environment. BAN logic proof and the formal security analysis under random oracle model are used to prove the completeness and security of the proposed scheme. Besides, the informal analysis demonstrates that our scheme surmount the vulnerabilities in Ali et al.'s scheme and provides desirable attributes like forward secrecy and three-factor secrecy. In addition, the performance and security comparison shows that our scheme provides strong security, while it has minimal communication overhead and computation cost.

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 under Grant No. 2018YFB0803605 and the National Natural Science Foundation of China under Grant No. 61873069.

References


