Hindawi
Security and Communication Networks
Volume 2019, Article ID 2838615, 15 pages
https://doi.org/10.1155/2019/2838615

Research Article

A Provably Secure Biometrics-Based Authentication Scheme for Multiserver Environment

Feifei Wang,[1] **Guoai Xu**,[1] **Chenyu Wang**,[1] **and Junhao Peng**[2]

[1] *Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2] *Guangzhou 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

Academic Editor: Carmen Fernandez-Gago

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 threatens, 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 security attacks. However, we found that Ali et al.'s scheme is not as secure as it claimed by demonstrating their scheme is vulnerable 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 $\mathscr{A}$ are described as follows.

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

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

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

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

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

(6) When evaluating forward secrecy, $\mathscr{A}$ 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 usually is 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 $\mathscr{A}$ is able to enumerate all the values in $D_{PW} * D_{ID}$ 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 brief 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 1: Notations.

Symbols	Description
U_i	i^{th} user
S_j	Server S_j
RC	Registration center
$\mathscr{A}$	Malicious adversary
x	Master key of RC
ID_i, PW_i, b_i	Identity, password and biometric of U_i
SID_j	Identity of S_j
P	A generator of elliptic curve group E_q
$E_{Key}()/D_{Key}()$	Symmetric encryption/decryption algorithm with key Key
SK	Session key between U_i and S_j
$\|$	The string concatenation operation
$\oplus$	The bitwise XOR operation
$H_1()$	Hash function
$H_2()$	Bio-hash function, it maps the biometric b_i and a tokenised random number to a random binary string

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 $SM_j = H_1(SID_j \| x)$ 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 $DID_i = E_{H_1(x)}(ID_i \| N_1)$, $A_i = H_1(ID_i \| x)P$, $B_i = H_2(b_i)P$, $E_i = A_i + B_i$, where N_1 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 $V_i = H_1(ID_i \| PW_i \| H_2(b_i))$ 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 $V_i^* = H_1(ID_i^* \| PW_i^* \| H_2(b_i^*))$ and checks if $V_i^* = V_i$. If the equation holds, proceed to the next step.

Step 2. U_i computes $R_i = r_iP$, $B_i^* = H_2(b_i^*)P$, $C_i = R_i + B_i^*$, $D_i = H_1(ID_i^* \| R_i \| SID_j \| E_i)$, 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 $(ID_i \| N_1) = D_{H_1(x)}(DID_i)$, $A_i' = H_1(ID_i \| x)P$, $B_i' = E_i - A_i'$, $R_i' = C_i - B_i'$, $D_i' = H_1(ID_i \| R_i' \| SID_j \| E_i)$ and compares D_i' with D_i. If they are equal, proceed to the next step.

Step 4. RC computes $DID_i^{new} = E_{H_1(x)}(ID_i \| r_{RC})$, $F_i = E_{H_1(SID_j\|x)}(ID_i \| R_i' \| H_2(b_i))$, $R_{RC} = r_{RC}P$, $K_i = R_{RC} + H_2(b_i)P$, $L_i = H_1(R_{RC} \| DID_i^{new} \| ID_i \| SID_j)$, 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 $(ID_i \| R_i'' \| H_2(b_i)') = D_{SM_j}(F_i)$, $R_{RC}' = K_i - H_2(b_i)'P$, $L_i' = H_1(R_{RC}' \| DID_i^{new} \| ID_i \| SID_j)$ and checks if $L_i' = L_i$. If it holds, S_j computes $R_S = r_SP$, $Q_i = R_S + R_i''$, $M_i = H_1(DID_i^{new} \| R_S \| R_{RC}' \| H_2(b_i)'P)$, 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 $R_S' = Q_i - R_i$, $R_{RC}'' = K_i - B_i^*$, $M_i' = H_1(DID_i^{new} \| R_S' \| R_{RC}'' \| B_i^*)$ and checks if $M_i' = M_i$. If the equation holds, U_i computes $SK = H_1(R_i \| R_S' \| R_{RC}'')$, $Z_i = SK \cdot P + H_1(ID_i \| H_2(b_i^*)) \cdot P$. U_i replaces DID_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 $SK = H_1(R_i'' \| R_S \| R_{RC}')$, $Z_i' = SK \cdot P + H_1(ID_i \| H_2(b_i)') \cdot P$ 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 $\mathscr{A}$ 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 $\mathscr{A}$ is able to retrieve the session key in the following steps.

Step 1. Compute $SM_j = H_1(SID_j \| x)$, $(ID_i \| R_i' \| H_2(b_i)') = D_{SM_j}(F_i)$.

Step 2. Compute $R_S' = Q_i - R_i'$.

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

Step 4. Compute $SK = H_1(R_i' \| R_S' \| R_{RC}')$.

3.2. User Impersonation Attack. The adversary $\mathscr{A}$ gets U_i's identity ID_i by shoulder surfing and U_i's biometric b_i by

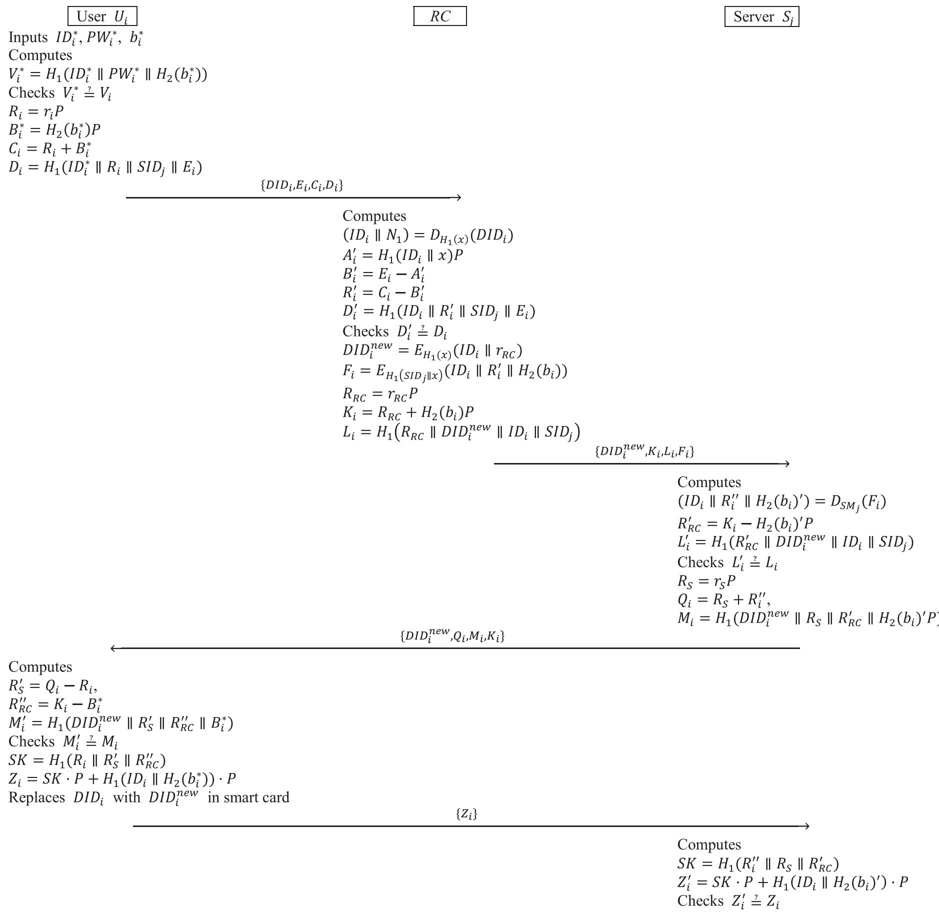

FIGURE 1: Login and authentication phase of Ali et al.'s scheme.

a malicious terminal and intercepts $\{DID_i, E_i, C_i, D_i\}$ from public channel. Then $\mathscr{A}$ performs user impersonation attack in the following steps.

Step 1. $\mathscr{A}$ computes $R_{i_a} = r_{i_a}P$, $B_i = H_2(b_i)P$, $C_a = R_{i_a} + B_i$, $D_a = H_1(ID_i \| R_{i_a} \| SID_j \| E_i)$, where r_{i_a} is a random number. $\mathscr{A}$ sends $\{DID_i, E_i, C_a, D_a\}$ to RC.

Step 2. Upon receiving $\{DID_i, E_i, C_a, D_a\}$, RC computes $(ID_i \| N_1) = D_{H_1(x)}(DID_i)$, $B_i' = E_i - H_1(ID_i \| x)P$, $R_{i_a}' = C_a - B_i'$, $D_a' = H_1(ID_i \| R_{i_a}' \| SID_j \| E_i)$, obviously $D_a' = D_a$. Then RC computes $DID_i^{new} = E_{H_1(x)}(ID_i \| r_{RC})$, $F_a = E_{H_1(SID_j \| x)}(ID_i \| R_{i_a}' \| H_2(b_i))$, $R_{RC} = r_{RC}P$, $K_i = R_{RC} + H_2(b_i)P$, $L_i = H_1(R_{RC} \| DID_i^{new} \| ID_i \| SID_j)$, where r_{RC} is a random number. RC sends $\{DID_i^{new}, K_i, L_i, F_a\}$ to S_j.

Step 3. Upon receiving $\{DID_i^{new}, K_i, L_i, F_a\}$, $S_j$computes $(ID_i \| R_{i_a}'' \| H_2(b_i)') = D_{SM_j}(F_a)$, $R_{RC}' = K_i - H_2(b_i)'P$, $L_i' = H_1(R_{RC}' \| DID_i^{new} \| ID_i \| SID_j)$, obviously $L_i' = L_i$. Then S_j computes $R_S = r_S P$, $Q_a = R_S + R_{i_a}''$, $M_i = H_1(DID_i^{new} \| R_S \| R_{RC}' \| H_2(b_i)'P)$, where r_S is a random number. S_j sends $\{DID_i^{new}, Q_a, M_i, K_i\}$ to $\mathscr{A}$.

Step 4. Upon receiving $\{DID_i^{new}, Q_a, M_a, K_i\}$, $\mathscr{A}$ computes $R_S' = Q_a - R_{i_a}$, $R_{RC}'' = K_i - B_i$, $SK = H_1(R_{i_a} \| R_S' \| R_{RC}'')$, $Z_a = SK \cdot P + H_1(ID_i \| H_2(b_i)) \cdot P$ and sends $\{Z_a\}$ to S_j.

Step 5. Upon receiving $\{Z_a\}$, S_j computes $SK = H_1(R_{i_a}'' \| R_S \| R_{RC}')$, $Z_a' = SK \cdot P + H_1(ID_i \| H_2(b_i)') \cdot P$, obviously $Z_a' = Z_a$. S_j regards $\mathscr{A}$ as legitimate user U_i.

3.3. Server Impersonation Attack. The adversary $\mathscr{A}$ obtains U_i's biometric b_i and intercepts $\{DID_i, E_i, C_i, D_i\}$ from public channel. Afterwards, $\mathscr{A}$ performs server impersonation attack in the following steps.

Step 1. $\mathscr{A}$ chooses two random numbers r_{S_a}, r_{RC_a} and computes $R_{S_a} = r_{S_a}P$, $R_{RC_a} = r_{RC_a}P$, $B_i = H_2(b_i)P$, $R_i' = C_i - B_i$, $Q_a = R_{S_a} + R_i'$, $K_a = R_{RC_a} + B_i$, $M_a = H_1(DID_i^a \| R_{S_a} \| R_{RC_a} \| B_i)$, where DID_i^a is a random binary string whose length is equal with DID_i. $\mathscr{A}$ sends $\{DID_i^a, Q_a, M_a, K_a\}$ to U_i.

Step 2. Upon receiving $\{DID_i^a, Q_a, M_a, K_a\}$, U_i computes $R_{S_a}' = Q_a - R_i$, $R_{RC_a}' = K_a - B_i^*$, $M_a' = H_1(DID_i^a \| R_{S_a}' \| R_{RC_a}' \| B_i^*)$, obviously $M_a' = M_a$; U_i regards $\mathscr{A}$ as the server S_j. U_i computes $SK = H_1(R_i \| R_{S_a}' \| R_{RC_a}')$, $Z_a = SK \cdot P + H_1(ID_i \|$

$H_2(b_i^*)) \cdot P$. U_i replaces DID_i with DID_i^a in his smart card, sends $\{Z_a\}$ to $\mathscr{A}$.

Step 3. Upon receiving $\{Z_a\}$, $\mathscr{A}$ computes $SK = H_1(R_i' \parallel R_{S_a} \parallel R_{RC_a})$. $\mathscr{A}$ establishes a session key SK with U_i successfully.

3.4. Denial of Service Attack. In the process of server impersonation attack, the adversary $\mathscr{A}$ delivers a forged dynamic identity DID_i^a to U_i. U_i believes its validity and stores it in the smart card. When U_i intends to access the server, U_i sends a login request with DID_i^a to RC. As DID_i^a is a random binary string rather than the encryption results of ID_i and a random number, RC rejects the login request. In addition, U_i cannot login any server, unless U_i reregister with RC.

3.5. Privileged Insider Attack. In authentication phase of Ali et al.'s scheme, $H_2(b_i)$ and new dynamic identity DID_i^{new} is exposed to S_j. With $H_2(b_i)$ and DID_i^{new}, S_j who acts as a privileged insider can masquerade user U_i to access server or impersonate the other server to defraud U_i. As their attack procedures are the same with aforementioned user impersonation attack and server impersonation attack, we omit it.

3.6. Known Session-Specific Temporary Information Attack. Known session-specific temporary information attack is a cryptanalysis under the circumstance the temporary secret value such as random number is leaked and the adversary tries to breach the current session key. Suppose that $\mathscr{A}$ obtains U_i's biometric b_i and intercepts $\{DID_i^{new}, Q_i, M_i, K_i\}$ from public channel. In the case that random number r_i is compromised. $\mathscr{A}$ can get the session key in the following steps.

Step 1. Compute $R_i = r_i P$.

Step 2. Compute $B_i = H_2(b_i)P$.

Step 3. Compute $R_{RC} = K_i - B_i$.

Step 4. Compute $R_S = Q_i - R_i$.

Step 5. Compute $SK = H_1(R_i \parallel R_S \parallel R_{RC})$.

3.7. Three-Factor Secrecy. In case that U_i's smart card $\{DID_i, E_i, P, E_{Key}(), V_i\}$ and biometric b_i are breached, the adversary is able to acquire U_i's password via the following steps.

Step 1. Guess the value of ID_i to be ID_i^* from identity dictionary space; guess the value of PW_i to be PW_i^* from identity dictionary space.

Step 2. Compute $V_i^* = H_1(ID_i^* \parallel PW_i^* \parallel H_2(b_i))$; check if $V_i^* = V_i$. If the equation holds, it shows that ID_i^* is U_i's real identity and PW_i^* is U_i's correct password.

Step 3. Repeat Steps 1 and 2, until $\mathscr{A}$ finds the correct ID_i and PW_i.

When the smart card and biometric of user are compromised, the attacker is able to breach the password. On the other hand, $\mathscr{A}$ 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 $SM_j = H_1(SID_j \parallel x)$ 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 his identity ID_i and password PW_i freely and imprints his biometric b_i. U_i calculates $P_i = H_1(PW_i \parallel H_2(b_i) \parallel r_i)$, 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 $A_i = H_1(x \parallel ID_i)$, $B_i = A_i \bigoplus P_i$, $V_i = H_1(P_i \bigoplus H_1(ID_i)) \bmod n$, where $2^4 \le n \le 2^8$. RC stores $\{B_i, V_i, E_{Key}(), P, P_{pub}, n\}$ 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 $P_i^* = H_1(PW_i^* \parallel H_2(b_i^*) \parallel r_i)$, $V_i^* = H_1(P_i^* \bigoplus H_1(ID_i^*)) \bmod n$ and checks if $V_i^* = V_i$. If this equation holds, the smart card computes $A_i^* = B_i \oplus P_i^*$, $R_i = N_1 P$, $C_i = H_1(N_1 P_{pub})$, $L_i = E_{C_i}(ID_i^* \parallel A_i^* \parallel SID_j)$, where N_1 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 $C_i' = H_1(xR_i)$, $(ID_i' \parallel A_i' \parallel SID_j') = D_{C_i'}(L_i)$, $A_i = H_1(x \parallel ID_i')$ and

User U_i | RC

Picks ID_i and PW_i, imprints b_i
Computes $P_i = H_1(PW_i \| H_2(b_i) \| r_i)$

$\xrightarrow{\{ID_i, P_i\}}$

Computes $A_i = H_1(x \| ID_i)$
$B_i = A_i \oplus P_i$,
$V_i = H_1(P_i \oplus H_1(ID_i)) \bmod n$
Stores $\{B_i, V_i, E_{Key}(\cdot), P, P_{pub}, n\}$ in a smart card

$\xleftarrow{\text{Smart card}}$

Stores r_i in smart card

Figure 2: User registration phase of the proposed scheme.

User U_i | RC | Server S_j

Inputs ID_i^*, PW_i^*, b_i^*
Computes
$P_i^* = H_1(PW_i^* \| H_2(b_i^*) \| r_i)$
$V_i^* = H_1(P_i^* \oplus H_1(ID_i^*)) \bmod n$
Checks $V_i^* \stackrel{?}{=} V_i$
$A_i^* = B_i \oplus P_i^*$
$R_i = N_1 P$
$C_i = H_1(N_1 P_{PUB})$
$L_i = E_{C_i}(ID_i^* \| A_i^* \| SID_j)$

$\xrightarrow{\{R_i, L_i\}}$

Computes
$C_i' = H_1(xR_i)$
$(ID_i' \| A_i' \| SID_j') = D_{C_i'}(L_i)$
$A_i = H_1(x \| ID_i')$
Checks if $A_i' = A_i$
$SM_j = H_1(SID_j' \| x)$
$Y_i = H_1(SID_j' \| SM_j)$
$M_i = E_{SM_j}(ID_i' \| R_i \| Y_i \| H_1(A_i \| C_i'))$

$\xrightarrow{\{M_i\}}$

Computes
$(ID_i'' \| R_i' \| Y_i' \| H_1(A_i \| C_i')') = D_{SM_j}(M_i)$
$Y_i'' = H_1(SID_j \| SM_j)$
Checks $Y_i'' \stackrel{?}{=} Y_i'$
$R_S = N_2 P$
$E_i = N_2 \cdot R_i$
$SK = H_1(E_i \| H_1(A_i \| C_i')')$
$F_i = H_1(ID_i'' \| SK \| R_S \| SID_j)$

$\xleftarrow{\{R_S, F_i\}}$

Computes
$E_i' = N_1 \cdot R_S$
$SK = H_1(E_i' \| H_1(A_i^* \| C_i))$
$F_i' = H_1(ID_i^* \| SK \| R_S \| SID_j)$
Checks $F_i' \stackrel{?}{=} F_i$
$Q_i = H_1(SK \| R_S)$

$\xrightarrow{\{Q_i\}}$

Computes $Q_i' = H_1(SK \| R_S)$
Checks $Q_i' \stackrel{?}{=} Q_i$

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

checks if $A_i = A_i'$. If the equation holds, RC computes $SM_j = H_1(SID_j' \| x)$, $Y_i = H_1(SID_j' \| SM_j)$, $M_i = E_{SM_j}(ID_i' \| R_i \| Y_i \| H_1(A_i \| C_i'))$. $\{M_i\}$ is transmitted to S_j.

Step 3. After receiving $\{M_i\}$, S_j computes $(ID_i'' \| R_i' \| Y_i' \| H_1(A_i \| C_i')') = D_{SM_j}(M_i)$, $Y_i'' = H_1(SID_j \| SM_j)$ and checks if Y_i'' is equal to Y_i'. If it holds, S_j computes $R_S = N_2 P$, $E_i = N_2 \cdot R_i$, $SK = H_1(E_i \| H_1(A_i \| C_i')')$, $F_i = H_1(ID_i'' \| SK \| R_S \| SID_j)$, where N_2 is a random number. $\{R_S, F_i\}$ is transmitted to U_i.

Step 4. After receiving $\{R_S, F_i\}$, U_i computes $E_i' = N_1 \cdot R_S$, $SK = H_1(E_i' \| H_1(A_i^* \| C_i))$, $F_i' = H_1(ID_i^* \| SK \| R_S \| SID_j)$ and checks if $F_i' = F_i$. If the equation holds, U_i computes $Q_i = H_1(SK \| R_S)$ and sends $\{Q_i\}$ to S_j.

Step 5. After receiving $\{Q_i\}$, S_j computes $Q_i' = H_1(SK \| R_S)$ 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 $P_i^* = H_1(PW_i^* \| H_2(b_i^*) \| r_i)$, $V_i^* = H_1(P_i^* \oplus H_1(ID_i^*)) \bmod n$

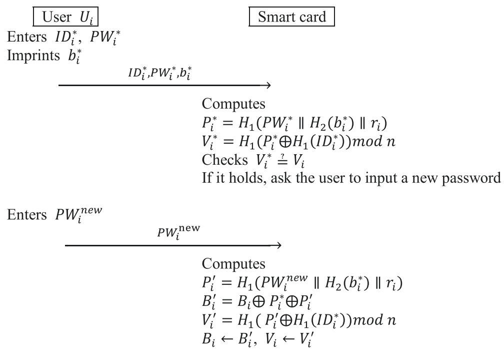

Figure 4: Password update phase.

and checks $V_i^* \stackrel{?}{=} V_i$. If it holds, the smart card asks the user to input a new password.

Step 2. U_i enters his new password PW_i^{new}. Then the smart card calculates $P_i' = H_1(PW_i^{new} \| H_2(b_i^*) \| r_i)$, $B_i' = B_i \bigoplus P_i^* \bigoplus P_i'$, $V_i' = H_1(P_i' \bigoplus H_1(ID_i^*)) \bmod n$. 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 at 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^a, S_j^a, and RC^a denote them.

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

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

Corrupt (U_i^a, 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^a to the adversary.

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

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

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

Test (U_i^a, 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^a 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^a or S_j^a is fresh, if the following conditions are satisfied.

(1) The instance is accepted and establishes a session key.

(2) The instance and its partner that belongs to the same session are never made a reveal query.

(3) The adversary never asks the Corrupt $(U_i^a, z = 1, 2, 3)$ query.

(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

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

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 $\mathscr{A}$ who tries to break the semantic security of our scheme. We assume that $\mathscr{A}$ is able to make at most q_s Send-queries, q_e Execute queries, q_h Hash queries, q_b Bio-hash queries, and q_ε Encryption/Decryption queries in polynomial time t. Then we have*

$$Adv_P^{ake}(\mathscr{A}) \le \frac{q_h^2 + 6q_s}{2^{l_1}} + \frac{q_b^2 + 2q_s}{2^{l_2}} + \frac{q_\varepsilon^2}{2^{l_3}} + \frac{(q_s + q_e)^2}{p} + \frac{2q_s}{Y} + 2q_h Adv_P^{ECDHP} \tag{2}$$

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. Adv_P^{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

$$Adv_P^{ake}(\mathscr{A}) = 2(\Pr[S_0]) - 1 \tag{3}$$

By a further transformation, we have

$$\begin{aligned} 2(\Pr[S_0]) - 1 &= 2(\Pr[S_0] - \Pr[S_6]) + 2\Pr[S_6] - 1 \\ &= 2\Pr[S_6] - 1 \\ &\quad + 2\sum_{i=0}^{5}(\Pr[S_i] - \Pr[S_{i+1}]) \end{aligned} \tag{4}$$

G_1: in this game, the hash oracle, bio-hash oracle, and encryption/decryption oracle are simulated by maintaining a hash list Λ_H, a bio-hash list Λ_{bH}, and an encryption/decryption list Λ_ε. For a hash query $H_1(\alpha)$, if there is an item (α, β) in Λ_H, the oracle returns β to the adversary. Otherwise, the oracle chooses a random number β, returns β to the adversary, and adds the item (α, β) to Λ_H. The bio-hash oracle is simulated in the same way. For an encryption query $E_k(str)$, if there is an item (k, str, γ) in Λ_ε, the oracle returns γ to the adversary. Otherwise, the oracle chooses a value γ from cipher text space, returns γ to the adversary, and adds the item (k, str, γ) to Λ_ε. For a decryption query $D_k(\gamma)$, if there is an item (k, str, γ) in Λ_ε, 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, γ) to Λ_ε. Besides, all oracles involved in security model are simulated in this game. Obviously, this game has no difference with G_0. We have

$$\Pr[S_0] - \Pr[S_1] = 0 \tag{5}$$

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 $q_b^2/2^{l_2+1}$.

(3) A collision happens in the output of symmetric encryption; the probability is less than $q_\varepsilon^2/2^{l_3+1}$.

(4) A collision happens on R_i or R_S; the probability is no more than $(q_s + q_e)^2/2p$.

So we have

$$\left|\Pr[S_1] - \Pr[S_2]\right| \le \frac{q_h^2}{2^{l_1+1}} + \frac{q_b^2}{2^{l_2+1}} + \frac{q_\varepsilon^2}{2^{l_3+1}} + \frac{(q_s + q_e)^2}{2p} \tag{6}$$

G_3: in this game, we avoid the situation that the adversary correctly guesses F_i or Q_i without making the corresponding hash query. The probability is at most $q_s/2^{l_1}$. Thus,

$$\left|\Pr[S_2] - \Pr[S_3]\right| \le \frac{q_s}{2^{l_1}}. \tag{7}$$

G_4: this game averts the execution when the adversary correctly guesses the authentication value A_i directly. The probability is at most $q_s/2^{l_1}$. We get

$$\left|\Pr[S_3] - \Pr[S_4]\right| \le \frac{q_s}{2^{l_1}}. \tag{8}$$

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_i^a, z). The following three cases are included.

Case 1. The adversary queries Corrupt $(U_i^a, 1)$ and Corrupt $(U_i^a, 2)$. To derive A_i, the adversary still needs to get the biometric. The probability that he correctly guesses the biometric is at most $q_s/2^{l_2}$

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

Case 3. The adversary queries Corrupt $(U_i^a, 1)$ and Corrupt $(U_i^a, 3)$. The probability that he correctly guesses the parameter B_i is no more than $q_s/2^{l_1}$.

The probability that the adversary gets A_i is less than $q_s * (1/Y + 1/2^{l_1} + 1/2^{l_2})$. We have

$$\left|\Pr[S_4] - \Pr[S_5]\right| \le q_s * \left(\frac{1}{Y} + \frac{1}{2^{l_1}} + \frac{1}{2^{l_2}}\right) \tag{9}$$

G_6: 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

$$\Pr[S_6] = \frac{1}{2} \tag{10}$$

G_6 has no difference with G_5, unless the adversary makes a hash query $H_1(E_i \parallel H_1(A_i \parallel C_i)$; we denote the event as Λ_1. We have

$$\left|\Pr[S_5] - \Pr[S_6]\right| \le \Pr[\Lambda_1] \tag{11}$$

G_7: we simulate the random self-reducibility of ECDHP in this game. For $R_i = N_1 P, R_S = N_2 P$, through selecting randomly in Λ_H, we can obtain the item containing $E_i = N_1 N_2 P$ with the probability $1/q_h$. Since the event Λ_1 denotes that the adversary makes a hash query $H_1(E_i \parallel H_1(A_i \parallel C_i)$. We have

$$\Pr[\Lambda_1] \le q_h Adv_P^{ECDHP} \tag{12}$$

Table 2: The notations and rules of BAN logic.

P, Q	A principal
X, Y	A statement
K	A key
$P \lhd X$	P sees X, P receives a message containing X
$P\| \sim X$	P said X, P sent a message including X
$P\| \equiv X$	P believes X is true
$P \overset{Y}{\Longleftrightarrow} Q$	P and Q share a secret Y
$\#(X)$	X is fresh
$P \overset{K}{\leftrightarrow} Q$	P and Q share a key K
$\{X\}_K$	X is encrypted under the key K
$< X >_Y$	X is combined with a secret Y
$P \Longrightarrow X$	P has jurisdiction over X
Message meaning rule	$\dfrac{P\| \equiv P \overset{K}{\longleftrightarrow} Q, P \lhd \{X\}_K}{P\| \equiv Q\| \sim X}$ or $\dfrac{P\| \equiv P \overset{Y}{\Longleftrightarrow} Q, P \lhd < X >_Y}{P\| \equiv Q\| \sim X}$
Belief rule	$\dfrac{P\| \equiv X, P\| \equiv Y}{P\| \equiv (X, Y)}$
Nonce-verification rule	$\dfrac{P\| \equiv \#(X), P\| \equiv Q\| \sim X}{P\| \equiv Q\| \equiv X}$
Jurisdiction rule	$\dfrac{P\| \equiv Q \Longrightarrow X, P\| \equiv Q\| \equiv X}{P\| \equiv X}$

Through the series of games above, we have

$$Adv_P^{ake}(\mathscr{A}) \leq \frac{q_h^2 + 6q_s}{2^{l_1}} + \frac{q_b^2 + 2q_s}{2^{l_2}} + \frac{q_\varepsilon^2}{2^{l_3}} + \frac{(q_s + q_e)^2}{p} + \frac{2q_s}{Y} + 2q_h Adv_P^{ECDHP} \quad (13)$$

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

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

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

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

We idealized the proposed scheme as follows.

M1: $U_i \longrightarrow RC {<} N_1P, U_i \overset{C_i}{\Longleftrightarrow} RC >_{U_i \overset{A_i}{\Longleftrightarrow} RC}$

M2: $RC \longrightarrow S_j \ \{N_1P, U_i \overset{H_1(A_i\|C_i)}{\Longleftrightarrow} S_j\}_{SM_j}$

M3: $S_j \longrightarrow U_i {<} N_1R_S, R_S, U_i \overset{SK}{\longleftrightarrow} S_j >_{H_1(A_i\|C_i)}$

M4: $S_j \longrightarrow U_i < N_2P, U_i \overset{SK}{\longleftrightarrow} S_j >_{H_1(A_i\|C_i)}$

The initiative assumption of our scheme is given as follows.

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

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

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

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

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

S6: $S_j| \equiv RC| \Longrightarrow (U_i \overset{H_1(A_i\|C_i)}{\Longleftrightarrow} S_j)$

S7: $U_i| \equiv U_i \overset{H_1(A_i\|C_i)}{\Longleftrightarrow} S_j$

S8: $U_i| \equiv \#(N_1)$

S9: $U_i| \equiv S_j \Longrightarrow (U_i \overset{SK}{\longleftrightarrow} S_j)$

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

S11: $S_j| \equiv U_i \Longrightarrow (U_i \overset{SK}{\longleftrightarrow} S_j)$

The proof of our scheme is performed as follows.

From M1, we have

(1) $RC \lhd \ < N_1P, \ U_i \overset{C_i}{\Longleftrightarrow} RC >_{U_i \overset{A_i}{\Longleftrightarrow} RC}$

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

(2) $RC| \equiv U_i| \sim < N_1P, \ U_i \overset{C_i}{\Longleftrightarrow} RC >$

According to S2, (2) and nonce-verification rule, we obtain

(3) $RC| \equiv U_i| \equiv < N_1P,\ U_i \overset{C_i}{\Longleftrightarrow} RC >$

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

(4) $RC| \equiv U_i \overset{C_i}{\Longleftrightarrow} RC$

From M2, we have

(5) $S_j \triangleleft \{N_1P\ ,U_i \overset{H_1(A_i\|C_i)}{\Longleftrightarrow} S_j\}_{SM_j}$

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

(6) $S_j| \equiv RC| \sim \{N_1P\ ,U_i \overset{H_1(A_i\|C_i)}{\Longleftrightarrow} S_j\}$

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

(7) $S_j| \equiv RC| \equiv \{N_1P\ ,U_i \overset{H_1(A_i\|C_i)}{\Longleftrightarrow} S_j\}$

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

(8) $S_j| \equiv U_i \overset{H_1(A_i\|C_i)}{\Longleftrightarrow} S_j$

From M3, we have

(9) $U_i \triangleleft < N_1R_S,\ R_S,\ U_i \overset{SK}{\longleftrightarrow} S_j >_{H_1(A_i\|C_i)}$

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

(10) $U_i| \equiv S_j| \sim< N_1R_S, R_S, U_i \overset{SK}{\longleftrightarrow} S_j >$

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

(11) $U_i| \equiv S_j| \equiv U_i \overset{SK}{\longleftrightarrow} S_j$(**Goal 1**)

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

(12) $U_i| \equiv U_i \overset{SK}{\longleftrightarrow} S_j$(**Goal 2**)

From M4, we have

(13) $S_j \triangleleft\ < N_2P, U_i \overset{SK}{\longleftrightarrow} S_j >_{H_1(A_i\|C_i)}$

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

(14) $S_j| \equiv U_i| \sim< N_2P, U_i \overset{SK}{\longleftrightarrow} S_j >$

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

(15) $S_j| \equiv U_i| \equiv U_i \overset{SK}{\longleftrightarrow} S_j$(**Goal 3**)

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

(16) $S_j| \equiv U_i \overset{SK}{\longleftrightarrow} S_j$(**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_j is unavailable. $\mathscr{A}$ cannot get any information about ID_i from the transmitted messages in public channel. In addition, $\mathscr{A}$ 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 $\mathscr{A}$ compromises the master key of RC and intercepts $\{R_i, L_i\}$, $\{R_S, F_i\}$ from public channel. Then $\mathscr{A}$ tries to compute the session key $SK = H_1(E_i \parallel H_1(A_i \parallel C_i))$. $\mathscr{A}$ can get C_i and A_i by computing $C_i = H_1(xR_i)$, $(ID_i' \parallel A_i' \parallel SID_j') = D_{C_i}(L_i)$. To get E_i, $\mathscr{A}$ needs to derive E_i from R_i, R_S. It means that $\mathscr{A}$ 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 $\mathscr{A}$ 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, $\mathscr{A}$ 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 $P_i^* = H_1(PW_i^* \parallel H_2(b_i^*) \parallel r_i)$, $V_i^* = H_1(P_i^* \bigoplus H_1(ID_i^*)) \bmod n$. Check $V_i^* \overset{?}{=} V_i$.

Step 3. Repeat Steps 1 and 2, until $\mathscr{A}$ finds a pair of $< ID_i^*, PW_i^* >$ satisfying $V_i^* = V_i$.

However, even if $\mathscr{A}$ finds a pair of $< ID_i^*, PW_i^* >$ satisfying $V_i^* = V_i$, 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 $V_i^* = V_i$. 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 $\mathscr{A}$ tries to impersonate user and forge a login requested message $\{R_i, L_i\}$. $\mathscr{A}$ computes $R_{i_a} = N_{1_a}P$, where N_{1_a} is a random number. To compute L_i, $\mathscr{A}$ needs to know A_i. However, $\mathscr{A}$ 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, $\mathscr{A}$ tries to retrieve A_i from B_i. As $A_i = B_i \oplus P_i$, $\mathscr{A}$ needs to get P_i at first. To compute P_i, $\mathscr{A}$ requires r_i, b_i, PW_i. That is to say, $\mathscr{A}$ cannot get A_i, unless he obtains all the three authentication factors at the same time. This is beyond the capacity of $\mathscr{A}$. The proposed scheme is secure against user impersonate attack.

6.5. Server Impersonation Attack. Suppose that $\mathscr{A}$ 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 $\{R_S, F_i\}$ to U_i. At first, $\mathscr{A}$ generates a random number N_{2_a} and computes $R_{S_a} = N_{2_a}P$, $E_i = N_{2_a} * R_i$. Next, to compute $SK = H_1(E_i \| H_1(A_i \| C_i)$, $\mathscr{A}$ still needs A_i and C_i. As analyzed above, $\mathscr{A}$ cannot obtain A_i. To derive C_i, the adversary needs to compromise the master key x or break the elliptic curve Diffie–Hellman problem. It is beyond the capacity of $\mathscr{A}$. 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 $\mathscr{A}$ intercepts $\{R_i, L_i\}$ 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 $\mathscr{A}$. Then $\mathscr{A}$ needs to generate a response message $\{Q_i\}$ and sends it to S_j. As N_1, A_i are unavailable, the adversary is unable to return a valid $\{Q_i\}$ to S_j. The protocol finally aborts.

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

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

Case 4. Assume that $\mathscr{A}$ intercepts $\{R_S, F_i\}$ from public channel and replays $\{Q_i\}$ to S_j. The server deals with this message and finds that $Q_i' \neq Q_i$. 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 $E_i = N_1 * R_S$ or $E_i = N_2 * R_i$. 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, $\mathscr{A}$ 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, $\mathscr{A}$ 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, $B_i = A_i \bigoplus P_i$, $P_i = H_1(PW_i \| H_2(b_i) \| r_i)$, $V_i = H_1(P_i \bigoplus H_2(ID_i)) \bmod n$. 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 $8T_P + 22T_H$, $8T_P + 16T_H$, $8T_P + 24T_H$, $7T_P + 2T_D + 2T_E + 12T_H$, 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 Curve-25519, 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 1 adopts the comparatively efficient algorithms, that is, SHA-256, AES-128 encryption/decryption, and elliptic curve cryptosystem using Curve25519. In this scenario, our scheme requires 432.57 ms; the related schemes [22, 25–27] require 576.286 ms, 576.208 ms, 576.312 ms, and 504.518 ms, respectively. Scenario 2 uses the comparatively time-consuming algorithms, that is, SHA-512, AES-256 encryption/decryption, and elliptic curve cryptosystem using P521. In this scenario, our scheme requires 6319.022 ms; the related schemes [22, 25–27] require 8424.704 ms, 8424.512 ms, 8424.768 ms, and 7371.894 ms, respectively. The computation overhead of our scheme is superior to other schemes in both scenarios.

Figure 6 illustrates the communication overheads of related schemes. To evaluate the communication overhead,

Table 3: Results of security analysis of related schemes.

Security properties	He [22]	Kumari [25]	Feng [26]	Ali [27]	Our scheme
Resist offline password guessing attack	√	√	√	√	√
User anonymity	√	√	√	√	√
Resist denial of service attack	√	√	√	×	√
Resist user impersonation attack	√	√	√	×	√
Resist server impersonation attack	√	×	√	×	√
Forward secrecy	√	√	√	×	√
Resist known session-specific temporary information attack	×	×	×	×	√
Efficient wrong password and biometric detection	×	√	√	√	√
Resist privileged insider attack	√	√	√	×	√
Three-factor secrecy	√	×	×	×	√

Table 4: Computation cost of related schemes.

Schemes	User	RC	Server	Total cost
He [22]	$3T_P + 7T_H$	$2T_P + 10T_H$	$3T_P + 5T_H$	$8T_P + 22T_H$
Kumari [25]	$3T_P + 5T_H$	$2T_P + 6T_H$	$3T_P + 5T_H$	$8T_P + 16T_H$
Feng [26]	$3T_P + 7T_H$	$2T_P + 10T_H$	$3T_P + 7T_H$	$8T_P + 24T_H$
Ali [27]	$3T_P + 5T_H$	$1T_P + 1T_D + 2T_E + 3T_H$	$3T_P + 1T_D + 4T_H$	$7T_P + 2T_D + 2T_E + 12T_H$
Our protocol	$3T_P + 1T_E + 7T_H$	$1T_P + 1T_D + 1T_E + 5T_H$	$2T_P + 1T_D + 4T_H$	$6T_P + 2T_D + 2T_E + 16T_H$

Table 5: Computing time of cryptographic operations.

Operations	Computing time
SHA-256	13 μs
SHA-512	32 μs
AES-128 Encryption	64 μs
AES-128 Decryption	117μs
AES-256 Encryption	90 μs
AES-256 Decryption	165 μs
Curve25519 Point Multiplication	72 ms
P521 Point Multiplication	1053 ms

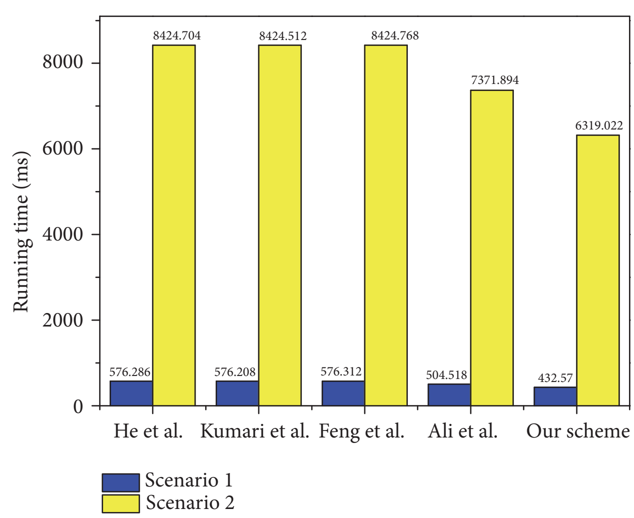

Figure 5: Running time of related schemes.

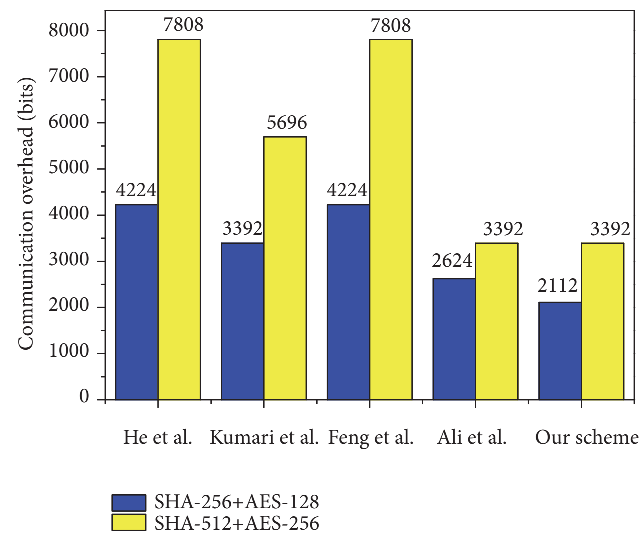

Figure 6: Communication overhead of related schemes.

we suppose that the identity of user, timestamp, random number are 64 bits; a point on E_p is 160 bits. When using the hash function SHA-256 and the symmetric algorithm AES-128, the communication overhead of our scheme is 2112 bits; the communication overheads of relevant schemes [22, 25–27] are 4224 bits, 3392 bits, 4224 bits, and 2624 bits, respectively. When adopting the hash function SHA-512 and the symmetric algorithm AES-256, the communication overhead of our scheme is 3392 bits; the communication overheads of relevant schemes [22, 25–27] are 7808 bits, 5696 bits, 7808 bits, and 3392 bits, respectively. Our scheme has the lowest communication overhead compared with other schemes.

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

[1] J.-L. Tsai and N.-W. Lo, "A chaotic map-based anonymous multi-server authenticated key agreement protocol using smart card," *International Journal of Communication Systems*, vol. 28, no. 13, pp. 1955–1963, 2015.

[2] D. He, S. Zeadally, N. Kumar, and W. Wu, "Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 9, pp. 2052–2064, 2016.

[3] W. Han and Z. Zhu, "An ID-based mutual authentication with key agreement protocol for multiserver environment on elliptic curve cryptosystem," *International Journal of Communication Systems*, vol. 27, no. 8, pp. 1173–1185, 2014.

[4] V. Odelu, A. K. Das, and A. Goswami, "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1953–1966, 2015.

[5] X. Liu, Y. Li, J. Qu, and L. Lu, "ELAKA: energy-efficient and lightweight multi-server authentication and key agreement protocol based on dynamic biometrics," *Wireless Personal Communications*, vol. 100, no. 3, pp. 767–785, 2018.

[6] A. G. Reddy, E.-J. Yoon, A. K. Das, V. Odelu, and K.-Y. Yoo, "Design of mutually authenticated key agreement protocol resistant to impersonation attacks for multi-server environment," *IEEE Access*, vol. 5, pp. 3622–3639, 2017.

[7] X. Li, K. Wang, J. Shen, S. Kumari, F. Wu, and Y. Hu, "An enhanced biometrics-based user authentication scheme for multi-server environments in critical systems," *Journal of Ambient Intelligence and Humanized Computing*, vol. 7, pp. 1–17, 2016.

[8] F. Wu, L. Xu, S. Kumari, and X. Li, "A novel and provably secure biometrics-based three-factor remote authentication scheme for mobile client-server networks," *Computers and Electrical Engineering*, vol. 45, pp. 274–285, 2015.

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

[10] J. Moon, Y. Choi, J. Jung, and D. Won, "An improvement of robust biometrics-based authentication and key agreement scheme for multi-server environments using smart cards," *PLoS ONE*, vol. 10, no. 12, Article ID e0145263, 2015.

[11] C.-T. Chen and C.-C. Lee, "A two-factor authentication scheme with anonymity for multi-server environments," *Security and Communication Networks*, vol. 8, no. 8, pp. 1608–1625, 2015.

[12] C. Wang, G. Xu, and W. Li, "A secure and anonymous two-factor authentication protocol in multiserver environment," *Security and Communication Networks*, vol. 2018, Article ID 9062675, 15 pages, 2018.

[13] D. Mishra, "Design and analysis of a provably secure multi-server authentication scheme," *Wireless Personal Communications*, vol. 86, no. 3, pp. 1095–1119, 2016.

[14] K.-H. Yeh, "A provably secure multi-server based authentication scheme," *Wireless Personal Communications*, vol. 79, no. 3, pp. 1621–1634, 2014.

[15] W.-B. Hsieh and J.-S. Leu, "An anonymous mobile user authentication protocol using self-certified public keys based on multi-server architectures," *The Journal of Supercomputing*, vol. 70, no. 1, pp. 133–148, 2014.

[16] X. Li, J. Niu, S. Kumari, J. Liao, and W. Liang, "An Enhancement of a Smart Card Authentication Scheme for Multi-server Architecture," *Wireless Personal Communications*, vol. 80, no. 1, pp. 175–192, 2015.

[17] D. Wang and P. Wang, "Offline dictionary attack on password authentication schemes using smart cards," in *Proceedings of the 16th International Conference on Information Security (ISC '13)*, pp. 221–237, Dallas, TX, USA, 2013.

[18] E.-J. Yoon and K.-Y. Yoo, "Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem," *The Journal of Supercomputing*, vol. 63, no. 1, pp. 235–255, 2013.

[19] H. Kim, W. Jeon, K. Lee, Y. Lee, and D. Won, "Cryptanalysis and improvement of a biometrics-based multi-server authentication with key agreement scheme," in *Proceedings of the 12th International Conference on Computational Science and Its Applications (ICCSA '12)*, pp. 391–406, Salvador de Bahia, Brazil, June 2012.

[20] R. Amin and G. P. Biswas, "Design and analysis of bilinear pairing based mutual authentication and key agreement protocol usable in multi-server environment," *Wireless Personal Communications*, vol. 84, no. 1, pp. 439–462, 2015.

[21] P. Chandrakar and H. Om, "Cryptanalysis and improvement of a biometric-based remote user authentication protocol usable in a multiserver environment," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 12, Article ID e3200, 2017.

[22] D. He and D. Wang, "Robust biometrics-based authentication scheme for multiserver environment," *IEEE Systems Journal*, vol. 9, no. 3, pp. 816–823, 2015.

[23] C. Wang, X. Zhang, Z. Zheng, and M. K. Khan, "Cryptanalysis and improvement of a biometric-based multi-server authentication and key agreement scheme," *PLoS ONE*, vol. 11, no. 2, Article ID e0149173, 2016.

[24] L. Yang, Z. Zheng, and M. K. Khan, "Cryptanalysis and improvement of a biometrics-based authentication and key agreement scheme for multi-server environments," *PLoS ONE*, vol. 13, no. 3, Article ID e0194093, 2018.

[25] S. Kumari, X. Li, F. Wu, A. K. Das, K.-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.

[26] 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, 2018.

[27] R. Ali and A. K. Pal, "An efficient three factor-based authentication scheme in multiserver environment using ECC," *International Journal of Communication Systems*, vol. 31, no. 4, Article ID e3484, 2018.

[28] Y. Lu, L. Li, H. Peng, and Y. Yang, "A biometrics and smart cards-based authentication scheme for multi-server environments," *Security and Communication Networks*, vol. 8, no. 17, pp. 3219–3228, 2015.

[29] S. A. Chaudhry, "A secure biometric based multi-server authentication scheme for social multimedia networks," *Multimedia Tools and Applications*, vol. 75, no. 20, pp. 12705–12725, 2016.

[30] D. Mishra, A. Das, and S. Mukhopadhyay, "A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards," *Expert Systems with Applications*, vol. 41, no. 18, pp. 8129–8143, 2014.

[31] 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.

[32] 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, article 2946, 2017.

[33] M. Burrows, M. Abadi, and R. Needham, "Logic of authentication," *ACM Transactions on Computer Systems*, vol. 8, no. 1, pp. 18–36, 1990.

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