Hindawi
Security and Communication Networks
Volume 2021, Article ID 6627956, 14 pages
https://doi.org/10.1155/2021/6627956

Research Article

Improved ECC-Based Three-Factor Multiserver Authentication Scheme

Tsu-Yang Wu,[1] **Lei Yang**,[1] **Zhiyuan Lee**,[1] **Chien-Ming Chen**,[1] **Jeng-Shyang Pan**,[1] **and SK Hafizul Islam**[2]

[1] *College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China*
[2] *Department of Computer Science and Engineering, Indian Institute of Information Technology Kalyani, Kalyani, West Bengal 741235, India*

Correspondence should be addressed to Chien-Ming Chen; chienmingchen@ieee.org

Received 20 October 2020; Revised 25 November 2020; Accepted 29 December 2020; Published 19 January 2021

Academic Editor: Stelvio Cimato

A multiserver environment can improve the efficiency of mobile network services more effectively than a single server in managing the increase in users. Because of the large number of users, the security of users' personal information and communication information is more important in a multiserver environment. Recently, Wang et al. proposed a multiserver authentication scheme based on biometrics and proved the security of their scheme. However, we first demonstrate that their scheme is insecure against a known session-specific temporary information attacks, user impersonation attacks, and server impersonation attacks. To solve the security weakness, we propose an improved scheme based on Wang et al.'s scheme. The security of our improved scheme is also validated based on the formal security analysis, Burrows–Abadi–Needham (BAN) logic, ProVerif, and informal security analysis. Security and performance comparisons prove the security and efficiency of our scheme.

1. Introduction

With the development of information technologies [1–8] and the widespread application of the Internet of Things [9–12], mobile communication has emerged in many network communication environments. The multiserver environments in mobile communication improve the efficiency of user communications; therefore, it is more popular than single-server environments for users. The multiserver environment overcomes the limited storage and computing of the single-server environment and can provide more remote services. A typical multiserver environment is shown in Figure 1.

Owing to the convenience of multiserver environments, authentication problems in the communication process cannot be disregarded. To date, three methods can be used to achieve user authentication in the environment. The first is password-based authentication [13–17]. This is the simplest method to perform authentication; however, an attacker can easily guess or steal a password from a party and impersonate as a valid user. The second is two-factor authentication, which is based on a password and a smart card [18–24]. Compared with password-based authentication, two-factor authentication improves security. However, if the smart card is stolen, then the information stored in the smart card may be recovered. This will result in well-known attacks, such as offline guessing attacks. In the past few years, Wang et al. have proposed some two-factor authentication schemes in different application scenarios. In 2014, they proposed an anonymous two-factor authentication scheme in a distributed system [19]. In the same year, they proposed an anonymous two-factor authentication scheme in a wireless sensor network [20]. In 2016, Wang et al. [25] compared and evaluated some representative two-factor authentication schemes and proposed a new evaluation standard for two-factor authentication schemes. In 2018, Wang et al. [26] proposed an evaluation framework for a two-factor authentication scheme for real-time data access in industrial wireless sensor networks and evaluated the relevant schemes. The third is three-factor authentication,

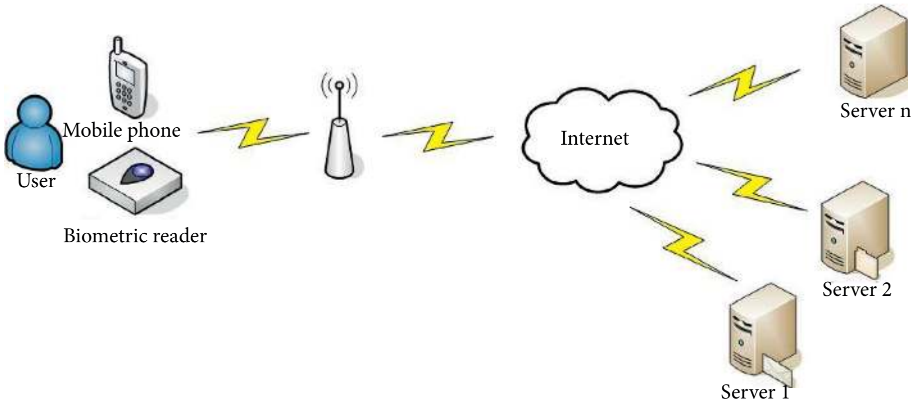

Figure 1: Typical multiserver environment.

which is based on passwords, smart cards, and biometrics [27–39]. In a public channel, an attacker may eavesdrop, modify, or replay transmitted messages. This poses a significant threat to the security of users. Because only the password- or smart card-based authentication scheme exhibits low security, applying biometrics to authentication schemes can overcome the insecurity of password- or smart card-based schemes. Therefore, a secure and efficient authentication scheme based on biometrics must be designed.

Compared with Rivest–Shamir-Adleman (RSA) or ElGaml cryptosystems, elliptic curve cryptography (ECC) provides a small key size and computation efficiency under the same security level. In recent years, several biometric-based authentication schemes based on ECC have been proposed. In 2013, Pippal et al. [27] proposed a three-factor authentication scheme in a multiserver environment and claimed that their scheme can overcome all types of network attacks. In 2014, He and Wang [28] proposed a multiserver environment authentication scheme based on robust biometrics, claiming that their scheme was the first three-factor authentication scheme applicable to multiserver environments. In 2015, Odelu et al. [30] reported that the scheme proposed in [28] was vulnerable to a known session-specific temporary information attack and an impersonation attack and hence did not provide strong user anonymity; therefore, they proposed a secure multiserver authentication protocol based on biometric technology using smart cards. In the same year, Li et al. [31] discovered that Pippal et al.' s [27] scheme can provide incorrect authentication but could not overcome impersonation, stolen smart card, and internal attacks. Therefore, Li et al. [31] proposed an improved scheme to overcome the problems above. In 2017, Kumari et al. [32] proposed a provable secure multicloud server authentication scheme based on biometrics. However, in 2018, Feng et al. [33] discovered that the scheme presented in [32] could not guarantee user anonymity, three-factor security, perfect forward security, etc.; hence, they proposed a multiserver environment authentication scheme based on anonymous biometrics. In the same year, Ali and Pal [34] analyzed Li et al.'s [31] scheme and discovered that it could not overcome password-guessing, user impersonation, insider, and smart card theft attacks nor could they guarantee user anonymity. Ali and Pal [34] proposed a three-factor multiserver authentication scheme based on an elliptic curve cryptosystem to solve the abovementioned issues. Unfortunately, Wang et al. [36] discovered that the scheme presented in [34] was vulnerable to user impersonation, server impersonation, privileged insider, and denial-of-service attacks, among others, and could not provide both forward and three-factor confidentiality. Therefore, Wang et al. proposed an improved multiserver authentication scheme based on biometrics and claimed that their scheme can overcome offline password-guessing, user impersonation, server impersonation, known specific session temporary information, three-factor security, user anonymity, and privileged internal attacks. Some important related works are summarized in Table 1.

In this study, we investigated Wang et al.'s scheme subject to known session-specific temporary information, user impersonation, and server impersonation attacks. To overcome the abovementioned attacks, we refer to Wang et al.'s scheme and propose an improved authentication scheme. Finally, we demonstrate that our scheme is semantically secure in the ROR model and overcome known attacks using the ProVerif tool and the BAN logic.

The remainder of this paper is organized as follows. A simple review and cryptanalysis of the scheme proposed by Wang et al. is discussed in Sections 2 and 3, respectively. Section 4 elaborates the proposed scheme in detail. Section 5 demonstrates the security analysis of the proposed scheme. Section 6 presents a comparison of performance and security. Section 7 summarizes the paper.

2. Review of Wang et al.'s Scheme

Wang et al.'s scheme includes initialization, server and user registration, and login authentication phases. Their scheme involves three types of entities: users, servers, and a registration center. The notations used in the scheme and their descriptions are shown in Table 2.

2.1. Initialization. In this phase, the registration center (RC) selects an elliptic curve E_q, and the basic point P of E_q defines two hash functions $H(\cdot)$ and $h(\cdot)$. Subsequently, the

Table 1: The summary of authentication schemes.

Scheme	Cryptographic techniques	Limitations
Pippal et al. [27]	(1) Utilized one-way hash function (2) Based on Diffie–Hellman problem (3) Based on smart card	(1) Does not resist impersonation attacks (2) Does not resist internal attacks
Li et al. [31]	(1) Utilized one-way hash function (2) Based on Diffie–Hellman problem (3) Based on smart card	(1) Does not resist password-guessing attacks (2) Does not resist impersonation attacks (3) Does not resist internal attacks (4) Does not resist smart card theft attacks (5) Does not support user anonymity
Kumari et al. [32]	(1) Based on biometrics (2) Utilized one-way hash function (3) Based on anonymous authentication	(1) Does not support user anonymity (2) Does not resist man-in-the-middle attacks
Feng et al. [33]	(1) Utilized ECC (2) Based on smart card (3) Based on biometrics	(1) Does not provide three-factor secrecy (2) Does not resist known session-specific temporary information attack
Ali and Pal [34]	(1) Utilized ECC (2) Three-factor security (3) Based on data encryption scheme	(1) Does not resist impersonation attacks (2) Does not resist internal attacks (3) Does not provide forward secrecy (4) Does not provide three-factor secrecy (5) Does not resist known session-specific temporary information attack
Wang et al. [36]	(1) Utilized ECC (2) Based on biometrics (3) Based on data encryption scheme	(1) Does not resist impersonation attacks (2) Does not resist known session-specific temporary information attack

Table 2: Notations and descriptions.

Notations	Descriptions
RC	The registration center
Server_j	The j-th server
User_i	The i-th user
A	The attacker
P_{pub}	The public key of RC
$H(\cdot)$	Hash function
$h(\cdot)$	Biohash function
ID_i	User identity
PW_i	User's password
b_i	User's biometrics
SC	Smart card
SK	Session key
	Concatenation
$\oplus$	Bitwise XOR
$E_{\text{key}}(\cdot)/D_{\text{key}}(\cdot)$	Symmetric encryption/decryption algorithm with key

RC selects a random number x and computes the public key $P_{\text{pub}} = xP$, where x is the RC's secret key and publishes $\{E_q, P, P_{\text{pub}}, H(\cdot), h(\cdot)\}$.

2.2. Server and User Registration. The server Server_j selects its identity SID_j and sends its identity to the RC through a secure channel. The RC receives this message, computes $\text{SM}_j = H(\text{SID}_j \| x)$, and sends SM_j to Server_j. When Server_j receives SM_j, it stores it as the secret key.

The user User_i selects his ID_i and PW_i and imprints b_i. Subsequently, User_i selects a random number r_i, computes $P_i = H(\text{PW}_i \| h(b_i) \| r_i)$, and sends $\{\text{ID}_i, P_i\}$ to the RC. The RC receives this message and calculates the following:

$$\begin{aligned} A_i &= H(x \| \text{ID}_i), \\ B_i &= A_i \oplus P_i, \\ V_i &= H(P_i \oplus H(\text{ID}_i)) \bmod n, \end{aligned} \tag{1}$$

where $2^4 \le n \le 2^8$. Note that $H(P_i \oplus H(\text{ID}_i))$ is the technique of fuzzy-verifier [40]. The RC stores $\{B_i, V_i, E_{\text{key}}(\cdot), P, P_{\text{pub}}, n\}$ in the smart card (SC) and then sends the SC to User_i in a secure channel. Subsequently, User_i stores r_i in the SC.

2.3. Login and Authentication. In this phase, User_i and Server_j complete a mutual authentication and establish a session key (SK) with the aid of the RC.

Step 1 User_i enters ID_i and PW_i, imprints b_i, and logins the SC. Subsequently, the SC computes

$$\begin{aligned} P_i' &= H(\text{PW}_i \| h(b_i) \| r_i), \\ V_i' &= H(P_i' \oplus H(\text{ID}_i)) \bmod n, \end{aligned} \tag{2}$$

and verifies if $V_i' = V_i$. If they are equal, then User_i generates a random number N_1 and computes

$$\begin{aligned} A_i' &= B_i \oplus P_i', \\ R_i &= N_1 P, \\ C_i &= H(N_1 P_{\text{pub}}), \\ L_i &= E_{C_i}(\text{ID}_i \| A_i' \| \text{SID}_j). \end{aligned} \tag{3}$$

Next, User_i sends $\{R_i, L_i\}$ to the RC in the public channel.

Step 2 After the *RC* receives $\{R_i, L_i\}$, it computes

$$\begin{aligned} C_i' &= H(xR_i), \\ (\mathrm{ID}_i \| A_i' \| \mathrm{SID}_j) &= D_{C_i'}(L_i), \\ A_i &= H(x \| \mathrm{ID}_i), \end{aligned} \tag{4}$$

and verifies if $A_i' = A_i$. If they are equal, then the RC computes

$$\begin{aligned} \mathrm{SM}_j &= H(\mathrm{SID}_j \| x), \\ Y_i &= H(\mathrm{SID}_j \| \mathrm{SM}_j), \\ M_i &= E_{\mathrm{SM}_j}(\mathrm{ID}_i \| R_i \| Y_i \| H(A_i \| C_i')). \end{aligned} \tag{5}$$

Next, the RC sends $\{M_i\}$ to Server$_j$ in the public channel.

Step 3 After Server$_j$ receives $\{M_i\}$, it computes

$$\begin{aligned} (ID_i \| R_i \| Y_i \| H(A_i \| C_i')) &= D_{\mathrm{SM}_j}(M_i), \\ Y_i' &= H(\mathrm{SID}_j \| \mathrm{SM}_j), \end{aligned} \tag{6}$$

and verifies if $Y_i' = Y_i$. If they are equal, then Server$_j$ generates a random number N_2 and computes

$$\begin{aligned} R_S &= N_2 P, \\ E_i &= N_2 R_i, \\ \mathrm{SK}_j &= H(E_i \| Ht(A_i \| C_i')), \\ F_i &= H(\mathrm{ID}_i \| \mathrm{SK}_j \| tR_S \| n\mathrm{SID}_j). \end{aligned} \tag{7}$$

Subsequently, Server$_j$ sends $\{R_s, F_i\}$ to User$_i$ in the public channel.

Step 4 After User$_i$ receives $\{R_s, F_i\}$, he computes

$$\begin{aligned} E_i' &= N_1 R_S, \\ \mathrm{SK}_i &= H(E_i' \| Ht(A_i \| C_i)), \\ F_i' &= H(\mathrm{ID}_i \| \mathrm{SK}_i \| tR_S \| n\mathrm{SID}_j), \end{aligned} \tag{8}$$

and verifies if $F_i' = F_i$. If they are equal, then User$_i$ computes

$$Q_i = H(\mathrm{SK}_i \| R_S). \tag{9}$$

Next, User$_i$ sends $\{Q_i\}$ to Server$_j$ in the public channel.

Step 5 After Server$_j$ receives $\{Q_i\}$, it computes

$$Q_i' = H(\mathrm{SK}_j \| R_S), \tag{10}$$

and verifies if $Q_i' = Q_i$. If they are equal, then $\mathrm{SK}_i = \mathrm{SK}_j$ is the session key for User$_i$ and Server$_j$.

3. Cryptanalysis of Wang et al.'s Scheme

In this section, we demonstrate Wang et al.'s scheme subject to three security attacks. In our proposed attacks, we assumed that the attacker A is a legitimate user and has already registered with the RC.

3.1. Known Session-Specific Temporary Information Attack.

A known session-specific temporary information attack refers to a security attack in which an attacker attempts to obtain the current SK when temporary secret values such as random numbers are disclosed [41].

In this attack, we assume that the attacker A obtains temporary information N_1 and captures $\{R_i, L_i\}$ and $\{R_s, F_i\}$, which are transmitted over the public channel. Based on the above, A can compute

$$\begin{aligned} C_i &= H(N_1 P_{\mathrm{pub}}), \\ (\mathrm{ID}_i \| A_i' \| \mathrm{SID}_j) &= D_{C_i}(L_i), \\ E_i' &= N_1 R_S. \end{aligned} \tag{11}$$

Subsequently, A obtains C_i, A_i', and E_i'; hence, it can determine $\mathrm{SK} = H(E_i' \| Ht(A_i' \| C_i))$. Furthermore, based on the formulas above, A can obtain the user's ID_i; in other words, the user's anonymity is not protected.

3.2. User Impersonation Attack

Step 1 Based on Section 3.1, A can obtain ID_i,A_i', and SID_j. Subsequently, A generates a random number N_A and computes

$$\begin{aligned} R_A &= N_A P, \\ C_A &= H(N_A P_{\mathrm{pub}}), \\ L_A &= E_{C_A}(\mathrm{ID}_i \| A_i' \| \mathrm{SID}_j). \end{aligned} \tag{12}$$

A fakes User$_i$ to send $\{R_A, L_A\}$ to the RC.

Step 2 Upon receiving $\{R_A, L_A\}$, the RC computes

$$\begin{aligned} C_A' &= H(xR_A), \\ (\mathrm{ID}_i \| A_i' \| \mathrm{SID}_j) &= D_{C_A'}(L_A), \\ A_i &= H(x \| \mathrm{ID}_i). \end{aligned} \tag{13}$$

It is clear that $A_i' = A_i$. Next, the RC computes

$$\begin{aligned} \mathrm{SM}_j &= H(\mathrm{SID}_j \| x), \\ Y_i &= H(\mathrm{SID}_j \| \mathrm{SM}_j), \\ M_A &= E_{\mathrm{SM}_j}(\mathrm{ID}_i \| R_A \| Y_i \| H(A_i \| C_A')), \end{aligned} \tag{14}$$

and sends $\{M_A\}$ to Server$_j$.

Step 3 After receiving $\{M_A\}$, Server$_j$ computes

$$\begin{aligned} (\mathrm{ID}_i \| R_A \| Y_i \| H(A_i \| C_A')) &= D_{\mathrm{SM}_j}(M_A), \\ Y_i' &= H(\mathrm{SID}_j \| \mathrm{SM}_j). \end{aligned} \tag{15}$$

It is clear that $Y_i' = Y_i$. Next, Server$_j$ generates a random number N_2 and computes

$$\begin{aligned} R_S &= N_2 P, \\ E_A &= N_2 R_A, \\ \mathrm{SK}_{jA} &= H(E_A \| H(A_i \| C_A')), \\ F_A &= H\left(\mathrm{ID}_i \| \mathrm{SK}_{jA} \| tR_S \| n\mathrm{SID}_j\right), \end{aligned} \tag{16}$$

and sends $\{R_S, F_A\}$ to User$_i$.

Step 4 A intercepts the massage $\{R_S, F_A\}$ and computes

$$\begin{aligned} E_A' &= N_A R_S, \\ \mathrm{SK}_A &= H(E_A' \| Ht(A_i \| C_A)), \\ F_A' &= H\left(\mathrm{ID}_i \| \mathrm{SK}_A \| tR_S \| n\mathrm{SID}_j\right). \end{aligned} \tag{17}$$

It is clear that $F_A' = F_A$. Next, A computes

$$Q_A = H(\mathrm{SK}_A \| R_S), \tag{18}$$

and sends $\{Q_A\}$ to Server$_j$.

Step 5 Upon receiving $\{Q_A\}$, Server$_j$ computes $Q_A' = H(\mathrm{SK}_{jA} \| R_S)$. It is clear that $Q_A' = Q_A$. During this process, the server regards A as User$_i$.

3.3. Server Impersonation Attack. This attack is also based on C_i, ID_i, A_i', and SID_j in Section 3.1. When User$_i$ sends $\{R_i, L_i\}$ to the RC, A eavesdrops the message. Subsequently, when RC sends $\{M_i\}$ to Server$_j$, A intercepts the message. A generates a random number N_A and computes

$$\begin{aligned} R_A &= N_A P, \\ E_A &= N_A R_i, \\ \mathrm{SK}_{Ai} &= H(E_A \| Ht(A_i' \| C_i)), \\ F_A &= H\left(\mathrm{ID}_i \| \mathrm{SK}_{Ai} \| tR_A \| n\mathrm{SID}_j\right), \end{aligned} \tag{19}$$

and sends $\{R_A, F_A\}$ to User$_i$.

Upon receiving $\{R_A, F_A\}$, User$_i$ computes

$$\begin{aligned} E_A' &= N_1 R_A, \\ \mathrm{SK}_i &= H(E_A' \| Ht(A_i \| C_i)), \\ F_A' &= H\left(\mathrm{ID}_i \| t\mathrm{SK}_i \| nR_A \| q\mathrm{SID}_j\right). \end{aligned} \tag{20}$$

It is clear that $F_A' = F_A$. Next, User$_i$ computes

$$Q_A = H(\mathrm{SK}_i \| R_A), \tag{21}$$

and sends $\{Q_A\}$ to Server$_j$. At this point, A intercepts the message and computes

$$Q_A' = H(\mathrm{SK}_A \| R_A). \tag{22}$$

It is clear that $Q_A' = Q_A$. During the entire process, the user regards A as Server$_j$.

4. Improved Scheme

To overcome the attacks, we proposed an improved scheme based on Wang et al.'s scheme in this section. Our scheme still operates in a multiserver environment, including the initialization, modified server and user registration, and modified login and authentication phases. It is noteworthy that the initialization phase in our scheme is the same as that in Wang et al.'s scheme, and we used a rectangle to denote our modifications.

4.1. Modified Server and User Registration. The server Server$_j$ selects its identity SID_j and sends its identity to the RC through a secure channel. The RC receives this message and selects a random number e_j. Subsequently, the RC computes $\mathrm{SM}_j = H(\mathrm{SID}_j \| x \| e_j)$, stores $\{\mathrm{SID}_j, e_j\}$, and sends SM_j to Server$_j$. When Server$_j$ receives SM_j, it stores it in the database.

The user User$_i$ selects his ID_i and PW_i and imprints b_i. Subsequently, User$_i$ selects a random number r_i and computes

$$\begin{aligned} P_i &= H(\mathrm{PW}_i \| h(b_i) \| r_i), \\ \mathrm{HID}_i &= H(\mathrm{ID}_i \oplus r_i), \end{aligned} \tag{23}$$

and sends $\{\mathrm{HID}_i, \mathrm{ID}_i, P_i\}$ to the RC. The RC receives this message, selects a random number d_i, and computes

$$\begin{aligned} A_i &= H(x \| \mathrm{HID}_i \| t\mathrm{ID}_i \| nd_i), \\ B_i &= A_i \oplus P_i, \\ V_i &= H(P_i \oplus H(\mathrm{HID}_i)) \bmod n, \end{aligned} \tag{24}$$

where $2^4 \le n \le 2^8$. The RC stores $\{\mathrm{HID}_i, \mathrm{ID}_i, d_i\}$ in the database, stores $\{B_i, V_i, E_{\mathrm{key}}(\cdot), P, P_{\mathrm{pub}}, n\}$ in the SC, and sends the SC to User$_i$ in a secure channel. Next, User$_i$ stores r_i in the SC. The complete registration process is shown in Figure 2.

4.2. Modified Login and Authentication. In this phase, User$_i$ and Server$_j$ complete a mutual authentication and use the RC as an information center to establish an SK. The complete login and authentication processes are shown in Figure 3.

Step 1 User$_i$ enters ID_i and PW_i, imprints b_i, and logins the SC. Next, the SC computes

$$\begin{aligned} P_i' &= H(\mathrm{PW}_i \| h(b_i) \| r_i), \\ \mathrm{HID}_i' &= H(\mathrm{ID}_i \oplus r_i), \\ V_i' &= H(P_i' \oplus H(\mathrm{HID}_i')) \bmod n, \end{aligned} \tag{25}$$

and verifies if $V_i' = V_i$. If they are equal, User$_i$ generates a random number N_1 and computes

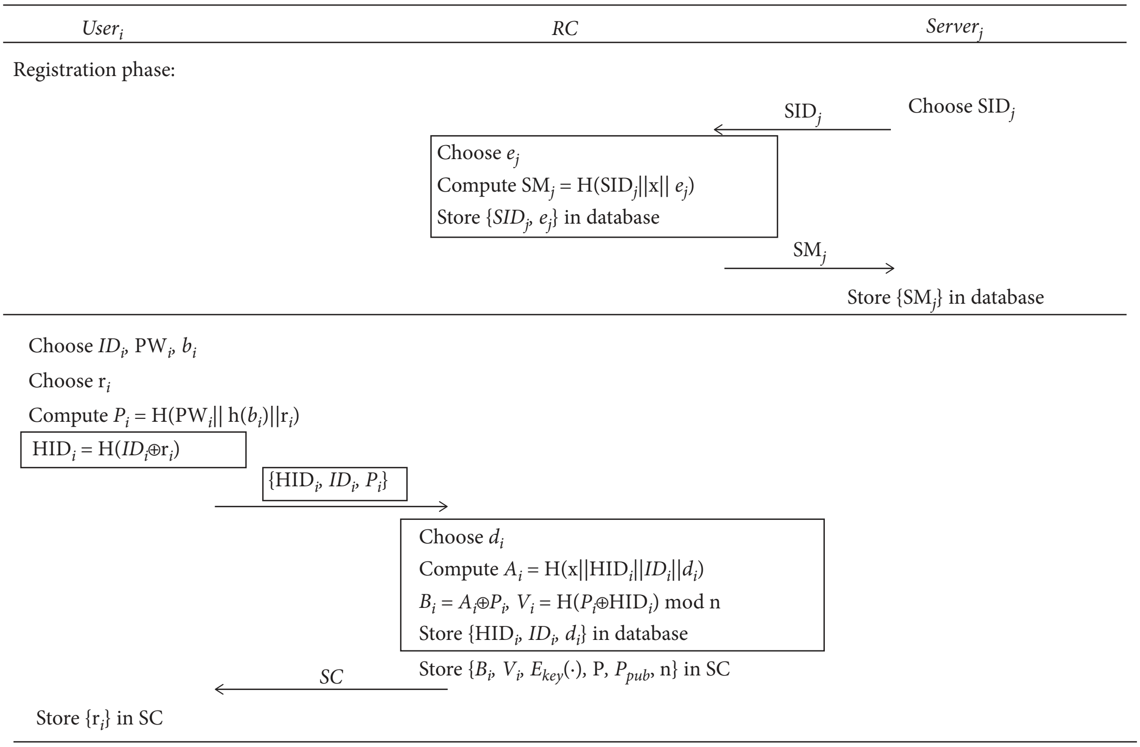
$User_i$
RC
$Server_j$
Registration phase:
SID_j
Choose SID_j
Choose e_j
Compute $\mathrm{SM}_j = \mathrm{H}(\mathrm{SID}_j||\mathrm{x}||\, e_j)$
Store $\{SID_j, e_j\}$ in database
SM_j
Store $\{\mathrm{SM}_j\}$ in database
Choose ID_i, PW_i, b_i
Choose r_i
Compute $P_i = \mathrm{H}(\mathrm{PW}_i||\, \mathrm{h}(b_i)||\mathrm{r}_i)$
$\mathrm{HID}_i = \mathrm{H}(ID_i \oplus \mathrm{r}_i)$
$\{\mathrm{HID}_i, ID_i, P_i\}$
Choose d_i
Compute $A_i = \mathrm{H}(\mathrm{x}||\mathrm{HID}_i||ID_i||d_i)$
$B_i = A_i \oplus P_i$, $V_i = \mathrm{H}(P_i \oplus \mathrm{HID}_i)$ mod n
Store $\{\mathrm{HID}_i, ID_i, d_i\}$ in database
Store $\{B_i, V_i, E_{key}(\cdot), \mathrm{P}, P_{pub}, \mathrm{n}\}$ in SC
SC
Store $\{\mathrm{r}_i\}$ in SC

Figure 2: Registration phase.

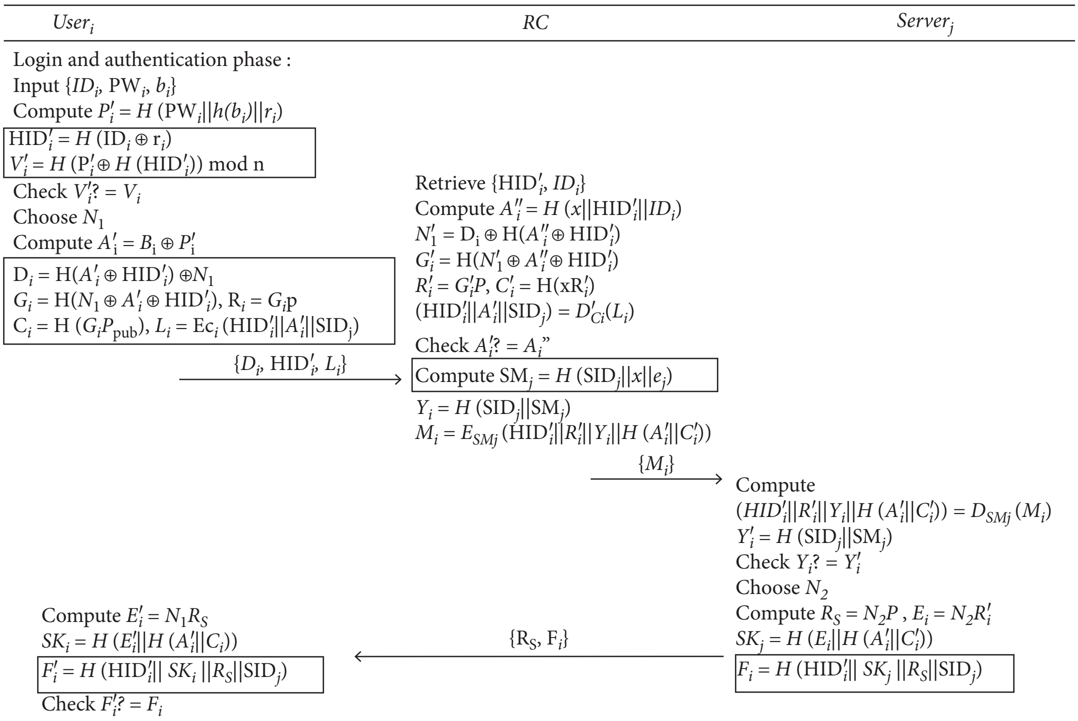
$User_i$
RC
$Server_j$
Login and authentication phase :
Input $\{ID_i, \mathrm{PW}_i, b_i\}$
Compute $P_i' = H\,(\mathrm{PW}_i||h(b_i)||r_i)$
$\mathrm{HID}_i' = H\,(\mathrm{ID}_i \oplus \mathrm{r}_i)$
$V_i' = H\,(\mathrm{P}_i' \oplus H\,(\mathrm{HID}_i'))$ mod n
Check $V_i'? = V_i$
Choose N_1
Compute $A_\mathrm{i}' = B_\mathrm{i} \oplus P_\mathrm{i}'$
$\mathrm{D}_i = \mathrm{H}(A_i' \oplus \mathrm{HID}_i') \oplus N_1$
$G_i = \mathrm{H}(N_1 \oplus A_i' \oplus \mathrm{HID}_i')$, $\mathrm{R}_i = G_i\mathrm{p}$
$\mathrm{C}_i = \mathrm{H}\,(G_iP_{\mathrm{pub}})$, $L_i = \mathrm{Ec}_i\,(\mathrm{HID}_i'||A_i'||\mathrm{SID}_\mathrm{j})$
$\{D_i, \mathrm{HID}_i', L_i\}$
Retrieve $\{\mathrm{HID}_i', ID_i\}$
Compute $A_i'' = H\,(x||\mathrm{HID}_i'||ID_i)$
$N_1' = \mathrm{D}_\mathrm{i} \oplus \mathrm{H}(A_i'' \oplus \mathrm{HID}_i')$
$G_i' = \mathrm{H}(N_1' \oplus A_i'' \oplus \mathrm{HID}_i')$
$R_i' = G_i'P$, $C_i' = \mathrm{H}(\mathrm{x}R_i')$
$(\mathrm{HID}_i'||A_i'||\mathrm{SID}_j) = D_{Ci}'(L_i)$
Check $A_i'? = A_i$”
Compute $\mathrm{SM}_j = H\,(\mathrm{SID}_j||x||e_j)$
$Y_i = H\,(\mathrm{SID}_j||\mathrm{SM}_j)$
$M_i = E_{SMj}\,(\mathrm{HID}_i'||R_i'||Y_i||H\,(A_i'||C_i'))$
$\{M_i\}$
Compute
$(HID_i'||R_i'||Y_i||H\,(A_i'||C_i')) = D_{SMj}\,(M_i)$
$Y_i' = H\,(\mathrm{SID}_j||\mathrm{SM}_j)$
Check $Y_i? = Y_i'$
Choose N_2
Compute $R_S = N_2P$, $E_i = N_2R_i'$
$SK_j = H\,(E_i||H\,(A_i'||C_i'))$
$F_i = H\,(\mathrm{HID}_i'||\, SK_j\, ||R_S||\mathrm{SID}_j)$
$\{\mathrm{R_S}, \mathrm{F}_i\}$
Compute $E_i' = N_1R_S$
$SK_i = H\,(E_i'||H\,(A_i'||C_i))$
$F_i' = H\,(\mathrm{HID}_i'||\, SK_i\, ||R_S||\mathrm{SID}_j)$
Check $F_i'? = F_i$

Figure 3: Login and authentication phase.

$$\begin{aligned}
A_i' &= B_i \oplus P_i',\\
D_i &= H\left(A_i' \oplus \mathrm{HID}_i'\right) \oplus N_1,\\
G_i &= H\left(N_1 \oplus A_i' \oplus \mathrm{HID}_i'\right),\\
R_i &= G_i P,\\
C_i &= H\left(G_i P_{\mathrm{pub}}\right),\\
L_i &= E_{C_i}\left(\mathrm{HID}_i' \| A_i' \| \mathrm{SID}_j\right).
\end{aligned} \tag{26}$$

Subsequently, User_i sends $M_1 = \{D_i, \mathrm{HID}_i', L_i\}$ to the RC in the public channel.

Step 2 After the RC receives M_1, it retrieves $\{\mathrm{HID}_i', \mathrm{ID}_i, d_i\}$ in the database and computes

$$\begin{aligned}
A_i'' &= H\left(x \| \mathrm{HID}_i' \| t\mathrm{ID}_i \| nd_i\right),\\
N_1' &= D_i \oplus H\left(A_i'' \oplus \mathrm{HID}_i'\right),\\
G_i' &= H\left(N_1' \oplus A_i'' \oplus \mathrm{HID}_i'\right),\\
R_i' &= G_i' P,\\
C_i' &= H\left(x R_i'\right),\\
\left(\mathrm{HID}_i' \| A_i' \| \mathrm{SID}_j\right) &= D_{C_i'}\left(L_i\right),
\end{aligned} \tag{27}$$

and verifies if $A_i' = A_i''$. If they are equal, the RC computes

$$\begin{aligned}
\mathrm{SM}_j &= H\left(\mathrm{SID}_j \| x \| te_j\right),\\
Y_i &= H\left(\mathrm{SID}_j \| \mathrm{SM}_j\right),\\
M_i &= E_{\mathrm{SM}_j}\left(\mathrm{HID}_i' \| R_i' \| Y_i \| H\left(A_i' \| C_i'\right)\right).
\end{aligned} \tag{28}$$

Next, the RC sends $M_2 = \{M_i\}$ to Server_j in the public channel.

Step 3 After Server_j receives M_2, it computes

$$\begin{aligned}
\left(\mathrm{HID}_i' \| R_i' \| Y_i \| H\left(A_i' \| C_i'\right)\right) &= D_{\mathrm{SM}_j}\left(M_i\right),\\
Y_i' &= H\left(\mathrm{SID}_j \| \mathrm{SM}_j\right),
\end{aligned} \tag{29}$$

and verifies if $Y_i' = Y_i$. If they are equal, Server_j generates a random number N_2 and computes

$$\begin{aligned}
R_S &= N_2 P,\\
E_i &= N_2 R_i',\\
\mathrm{SK}_j &= H\left(E_i \| H\left(A_i' \| C_i' \right),\right.\\
F_i &= H\left(\mathrm{HID}_i \| \mathrm{SK}_j \| tR_S \| n\mathrm{SID}_j\right).
\end{aligned} \tag{30}$$

Subsequently, Server_j sends $M_3 = \{R_s, F_i\}$ to User_i in the public channel.

Step 4 After User_i receives M_3, it computes

$$\begin{aligned}
E_i' &= N_1 R_S,\\
\mathrm{SK}_i &= H\left(E_i' \| H\left(A_i \| C_i \right),\right.\\
F_i' &= H\left(\mathrm{HID}_i \| \mathrm{SK}_i \| tR_S \| n\mathrm{SID}_j\right),
\end{aligned} \tag{31}$$

and verifies if $F_i' = F_i$. If they are equal, SK is the session key for User_i and Server_j.

5. Security Analysis

5.1. Formal Security Analysis. In this section, we show the security analysis of our improved scheme in the random oracle model [42]. First, we define the adversarial model [25, 26, 43–47] and simulate the adversary capabilities in a real attack. In the proposed scheme, three participants, User_i, Server_j, and RC, are involved. We use $\prod_U^x$, $\prod_S^y$, and $\prod_{\mathrm{RC}}^z$ to represent the xth communication of User_i, the y th communication of Server_j, and the z th communication of RC, respectively. To perform a formal security analysis, we defined the following query model for the attacker A.

Execute $(\prod_U^x, \prod_S^y, \prod_{\mathrm{RC}}^z)$: A performs this query to eavesdrop and record the messages transmitted on the public channel, such as the messages between the U and the RC, the messages between the RC and the S, and the messages between the S and the U

Hash $(\prod_E^K, M)$: based on this query, A can get the hash value if each item in the hash function is known, where $E = \{U, S, \mathrm{RC}\}$

Send $(\prod_E^K, M)$: A executes this query with message M and then receives the response message from the entity E

Reveal $(\prod_E^K)$: A executes this query to obtain the return result of current session key SK generated by E

Corrupt $(\prod_E^K)$: A executes this query to obtain information $\left\{B_i, V_i, r_i, E_{\mathrm{key}}(\cdot), P, P_{\mathrm{pub}}, n\right\}$ in the smart card

Test $(\prod_E^K)$: based on this query, an unbiased coin c begins to be flipped. If $c = 0$, A returns SK to a random string, and if $c = 1$, A returns SK to a session key

In the ROR model, the following theorem describes the security of our proposed scheme P.

Theorem 1. *If A runs P in an ROR model against a scheme in polynomial time, l represents the total number of bits of the biometric. The Adv_A^{AKE} that A's advantage breaks the security of SK in AKE scheme, and then $Adv_A^P \leq (q_h^2/|Hash|) + 2\max\{C' \cdot q_{send}^s, \ (q_{send}/2^l)\} + 2 \ Adv_\Omega^P(k)$, where q_{send} and q_h are the number of Send $(\prod_E^K, M)$ and Hash $(\prod_E^K, M)$; $|Hash|$ is the range space of $h(\cdot)$; C' and s are parameters of Zipf's law [48]; $Adv_\Omega^P(k)$ is the advantage of A breaking the symmetric cipher Ω.*

Proof. We define a sequence of five games, namely, GM_i $(i = 0, 1, 2, 3, 4)$. Let $\mathrm{Succ}_A^{\mathrm{GM}_i}$ represent the event that U_A wins GM_i. The $\mathrm{Adv}_{A,\mathrm{GM}_i}^P = \Pr[\mathrm{Succ}_A^{\mathrm{GM}_i}]$ represents the advantage of A winning GM_i, where $\Pr[E]$ is the probability of event E. The Adv_A^P represents the advantage of U_A that breaks the security of SK in the proposed scheme. The detailed description of GM_i is as follows.

Game GM_0: GM_0 is the first game that represents a real attack on the ROR model. At this point, select coin c to start GM_0. From semantic security, we can get $\mathrm{Adv}^P_A = |2 \cdot \mathrm{Adv}^P_{A,\mathrm{GM}_0} - 1|$.

Game GM_1: GM_1 means that A can perform the Execute query and get the message $\{D_i, \mathrm{HID}_i, L_i\}$, $\{M_i\}$ and $\{R_S, F_i\}$ transmitted in the scheme. At the end of the game, A will perform Reveal and Test queries to determine whether $\mathrm{SK} = H(E_i \| H(A_i \| C_i)$ can be obtained. But A cannot derive E_i, A_i, and C_i, so the probability of GM_1 is the same as that of GM_0, that is, $\mathrm{Adv}^P_{A,\mathrm{GM}_1} = \mathrm{Adv}^P_{A,\mathrm{GM}_0}$.

Game GM_2: GM_2 has added Hash and Send queries, E_i, A_i, and C_i, which are all protected by $h\ (\cdot)$. But E_i, A_i, and C_i are not directly obtained in the transmission channel, and according to the birthday paradox, we can get $|\mathrm{Adv}^P_{A,\mathrm{GM}_1} - \mathrm{Adv}^P_{A,\mathrm{GM}_2}| \le (q_h^2/2|\mathrm{Hash}|)$.

Game GM_3: Corrupt query is added in GM_3 and A can get the information $\{B_i, V_i, r_i, E_{\mathrm{key}}(\cdot), P, P_{\mathrm{pub}}, n\}$ in the smart card. The User_i uses the password and biometric information to register, and A wants to guess $P_i = H(\mathrm{PW}_i \| h(b_i) \| r_i)$, but the probability of guessing the biometrics is $1/2^l$ [49], which is almost negligible. Using Zipf's law [48], we can get $|\mathrm{Adv}^P_{A,\mathrm{GM}_2} - \mathrm{Adv}^P_{A,\mathrm{GM}_3}| \le \max\{C' \cdot q^s_{\mathrm{send}}, (q_{\mathrm{send}}/2^l)\}$.

Game GM_4: GM_4 is the last part of the game. At this time, A attempts to decrypt the information $\{L_i, M_i\}$ and uses the obtained information $\{B_i, V_i, r_i, E_{\mathrm{key}}(\cdot), P, P_{\mathrm{pub}}, n\}$ to infer SK. Without the master key x of RC, A cannot compute $A_i = H(x \| \mathrm{HID}_i \| t\mathrm{ID}_i \| nd_i)$ and $C_i = H(G_i P_{\mathrm{pub}}) = H(H(N_1 \oplus A_i' \oplus \mathrm{HID}_i')P_{\mathrm{pub}})$. According to the security of Ω symmetric encryption algorithm, we can obtain $|\mathrm{Adv}^P_{A,\mathrm{GM}_3} - \mathrm{Adv}^P_{A,\mathrm{GM}_4}| \le \mathrm{Adv}^P_{\Omega}(k)$.

All queries are performed by A. After querying the test query, only the coin c of GM_4 is left. Thus, the probability of guessing coin c is $\mathrm{Adv}^P_{A,\mathrm{GM}_4} = 1/2$. In summary, we can deduce

$$\begin{aligned}\frac{1}{2}\mathrm{Adv}^P_A &= \left|\mathrm{Adv}^P_{A,\mathrm{GM}_0} - \frac{1}{2}\right| = \left|\mathrm{Adv}^P_{A,\mathrm{GM}_1} - \frac{1}{2}\right| = \left|\mathrm{Adv}^P_{A,\mathrm{GM}_1} - \mathrm{Adv}^P_{A,\mathrm{GM}_4}\right| \\ &\le \left|\mathrm{Adv}^P_{A,\mathrm{GM}_1} - \mathrm{Adv}^P_{A,\mathrm{GM}_2}\right| + \left|\mathrm{Adv}^P_{A,\mathrm{GM}_2} - \mathrm{Adv}^P_{A,\mathrm{GM}_3}\right| \\ &\quad + \left|\mathrm{Adv}^P_{A,\mathrm{GM}_3} - \mathrm{Adv}^P_{A,\mathrm{GM}_4}\right| \\ &\le \frac{q_h^2}{2|\mathrm{Hash}|} + \max\left\{C' \cdot q^s_{\mathrm{send}}, \frac{q_{\mathrm{send}}}{2^l}\right\} + \mathrm{Adv}^P_{\Omega}(k). \end{aligned} \tag{32}$$

Therefore, the advantage of A breaking the scheme is $\mathrm{Adv}^P_A \le (q_h^2/|\mathrm{Hash}|) + 2\max\{C' \cdot q^s_{\mathrm{send}}, (q_{\mathrm{send}}/2^l)\} + 2\ \mathrm{Adv}^P_{\Omega}(k)$. □

5.2. Formal Security Analysis by BAN Logic. In this subsection, we demonstrate through the BAN logic that after our scheme verifies the authenticity of each other's identity and that the determined SK will not be obtained by others. In fact, the BAN logic is a rule used to define and analyze the communication process between two parties. Specifically, the conclusions obtained by the BAN logic are through rigorous logic analysis, which further explains the confidentiality and credibility of the communication information. The notations and rules of the BAN logic used in the BAN logic calculation performed in this study are cited in [24, 27, 28, 30, 31, 36, 50, 51]. The proof of our scheme is as follows:

5.2.1. Rules

rule (1) Nonce verification rule: $(P| \equiv \#(X), P| \equiv Q| \sim X)/(P| \equiv Q| \equiv X)$

rule (2) Message meaning rule: $(P| \equiv P \overset{K}{\leftrightarrow} Q, P \triangleleft X_K)/(P| \equiv Q| \sim X)$

rule (3) Jurisdiction rule: $(P| \equiv Q| \equiv X, P| \equiv Q| \Longrightarrow X)/(P| \equiv X)$

rule (4) Jurisdiction rule: $(P| \equiv \#(X))/(P| \equiv \#(X, Y))$

5.2.2. Goals

Goal 1. $U| \equiv S| \equiv U \overset{\mathrm{SK}_j}{\leftrightarrow} S$

Goal 2. $U| \equiv U \overset{\mathrm{SK}_j}{\leftrightarrow} S$

Goal 3. $S| \equiv U| \equiv U \overset{\mathrm{SK}_i}{\leftrightarrow} S$

Goal 4. $S| \equiv U \overset{\mathrm{SK}_i}{\leftrightarrow} S$

5.2.3. Idealize the Communication Messages

M_1: $U \longrightarrow \mathrm{RC}$: $\left\{ D_i, \mathrm{HID}_i', \left\{\mathrm{HID}_i', U \overset{A_i}{\leftrightarrow} \mathrm{RC}, \mathrm{SID}_j\right\}_{C_i} \right\}$

M_2: $\mathrm{RC} \longrightarrow S$: $\left\{\mathrm{HID}_i', R_i', Y_i, U \overset{H(A_i' \| C_i')}{\leftrightarrow} S\right\}_{\mathrm{SM}_j}$

M_3: $S \longrightarrow U$: $\left\{ R_s, \left\{\mathrm{HID}_i', U \overset{\mathrm{SK}_j}{\leftrightarrow} S, R_s, \mathrm{SID}_j\right\}_{H(A_i' \| C_i)} \right\}$

5.2.4. Initial Assumptions. A_1: $S| \equiv S \overset{\mathrm{SM}_j}{\leftrightarrow} \mathrm{RC}$

A_2: $S| \equiv \#(N_1)$

A_3: $S| \equiv \mathrm{RC}| \Rightarrow U \overset{H(A_i' \| C_i')}{\leftrightarrow} S$

A_4: $U| \equiv U \overset{H(A_i' \| C_i')}{\leftrightarrow} S$

A_5: $U| \equiv \#(N_2)$

A_6: $U| \equiv S| \Rightarrow U \overset{\mathrm{SK}_j}{\leftrightarrow} S$

A_7: $S| \equiv \#(N_2)$

A_8: $S| \equiv U| \Rightarrow U \overset{\mathrm{SK}_i}{\leftrightarrow} S$

5.2.5. The Proof of Our Proposed Scheme. For **Goal** 1 By M_2, we have S_1: $S \triangleleft \left\{\mathrm{HID}_i', R_i', Y_i, U \overset{H(A_i' \| C_i')}{\leftrightarrow} S\right\}_{\mathrm{SM}_j}$. Based on A_1, S_1, and rule (2), we have S_2: $S| \equiv \mathrm{RC}| \sim \left\{\mathrm{HID}_i', R_i', Y_i, U \overset{H(A_i' \| C_i')}{\leftrightarrow} S\right\}$. Based on A_2 and rule (4), we obtain S_3: $S| \equiv \#\left\{\mathrm{HID}_i', R_i', Y_i, U \overset{H(A_i' \| C_i')}{\leftrightarrow} S\right\}$. Using S_3, S_2, and rule (1), S_4: $S| \equiv \mathrm{RC}| \equiv \left\{\mathrm{HID}_i', R_i', Y_i, U \overset{H(A_i' \| C_i')}{\leftrightarrow} S\right\}$. Subsequently, we have S_5: $S| \equiv \mathrm{RC}| \equiv U \overset{H(A_i' \| C_i')}{\leftrightarrow} S$. Based on A_3,

S_5, and rule (3), we have S_6: $S| \equiv U \overset{H(A_i' \| C_i')}{\leftrightarrow} S$. By M_3, we have S_7: $U \lhd \left\{ \text{HID}_i', U \overset{\text{SK}_j}{\leftrightarrow} S, R_s, \text{SID}_j \right\}_{H(A_i' \| C_i')}$. Based on A_4, S_7, and rule (2), we have S_8: $U| \equiv S| \sim \left\{ \text{HID}_i', U \overset{\text{SK}_j}{\leftrightarrow} S, R_s, \text{SID}_j \right\}$. Based on A_5, S_8, rule (4), and rule (1), we obtain S_9: $U| \equiv S| \equiv \left\{ \text{HID}_i', U \overset{\text{SK}_j}{\leftrightarrow} S, R_s, \text{SID}_j \right\}$ and the following: S_{10}: $U| \equiv S| \equiv U \overset{\text{SK}_j}{\leftrightarrow} S$. (**Goal** 1).

For **Goal** 2, based on A_6, S_{10}, and rule (3), we have S_{11}: $U| \equiv U \overset{\text{SK}_j}{\leftrightarrow} S$. (**Goal** 2).

For **Goal** 3, based on M_3, we have S_{12}: $S \lhd \left\{ U \overset{\text{SK}_i}{\leftrightarrow} S, R_s \right\}_{H(A_i' \| C_i')}$. Subsequently, based on S_6, S_{12}, and rule (2), we have S_{13}: $S| \equiv U| \sim \left\{ U \overset{\text{SK}_i}{\leftrightarrow} S, R_s \right\}$. Based on A_7, S_{13}, rule (4), and rule (1), we obtain S_{14}: $S| \equiv U| \equiv \left\{ U \overset{\text{SK}_i}{\leftrightarrow} S, R_s \right\}$. Therefore, we have S_{15}: $S| \equiv U| \equiv U \overset{\text{SK}_i}{\leftrightarrow} S$. (**Goal** 3).

For **Goal** 4, based on A_8, S_{15}, and rule (3), we have S_{16}: $S| \equiv U \overset{\text{SK}_i}{\leftrightarrow} S$. (**Goal** 4).

5.3. Security Verification by ProVerif. We used the verification tool ProVerif to test the security of our proposed scheme. ProVerif is an important verification tool for verifying security fundamentals such as authentication, confidentiality, anonymity, and privacy [11, 24, 51, 52]. Furthermore, ProVerif can automatically verify the security of a scheme. It handles basic elements such as public key cryptography and the Diffie–Hellman mechanism.

The definition of the ProVerif code is shown in Figure 4. Our scheme comprised three entities: User$_i$, RC, and Server$_j$. Figures 5–7 show the user, RC, and server processes in our code, respectively. Five events were involved: UserAuthed, UserStarted, RCAcUser, ServerAcRC, and UserAcServer. Event UserAuthed means that User$_i$ has been successfully authenticated. Event UserStarted means that User$_i$ has started authentication. Event RCAcUser means that the RC has successfully authenticated the User$_i$. Event ServerAcRC means that Server$_j$ has successfully authenticated the RC. Event UserAcServer means that User$_i$ has successfully authenticated Server$_j$.

Next, we used ProVerif to query whether the attacker can obtain the identities of User$_i$ and Server$_j$ as well as the SK and whether the events above were executed in sequence. Figure 8 shows the events and queries in the code.

Finally, we executed the code to perform authentication, and the results are shown in Figure 9. The result shows that ProVerif confirmed the security of our scheme. Therefore, the attacker cannot obtain parameters $\left\{ \text{SK}_i, \text{SK}_j, \text{ID}_i, \text{SID}_j \right\}$,

```
(* channel* )
free ch :channel.(*  public channel * )
free sch: channel [private].(*  secure channel, used for registering * )
(*  shared keys * )
free SKi : bitstring  [private].
free Skj : bitstring  [private].
free IDi : bitstring  [private].
free SIDj : bitstring  [private].
(*  constants  * )
free  x:bitstring [private]. (*  the RC′s secret key* )
free  Ppub:bitstring [private].(*  the RC′s public key * )
free  Eq:bitstring.
free  P:bitsting.
(*  functions  &  reductions & equations * )
fun H1(bitstring) :bitstring. (*  hash function * )
fun H2(bitsring) :bitstring. (*  hash function * )
fun mult(bitstring,bitstring) :bitstring. (*  scalar multiplication operation * )
fun mod(bitstring,bitstring) :bitstring. (*  modulus operation * )
fun addone(bitstring) :bitstring. (*  add one * )
fun senc(bitstring,bitstring) :bitstring. (*  symmetric encryption * )
reduc forall m:bitstring, key:bitstring; sdec(senc(m,key),key) = m.
fun con(bitstring,bitstring):bitstring. (*  concatenation operation * )
reduc forall m:bitstring, n:bitstring; getmess(con(m, n)) = m.
fun xor(bitstring,bitstring);bitstring. (*  XOR operation * )
equation forall m:bitstring, n:bitstring; xor(xor(m, n),n) = m.
fun gen(bitstring):bitstring. (* Generator operation * )
fun rep(bitstring,bitstring):bitstring.
```

Figure 4: The terms definition in the ProVerif tool.

```
(* -------------User's process------------- * )
let ProcessUser =
    new IDi : bitstring;(*  the user′s ID *)
    new PWi : bitstring; (* the user′s password *)
    new bi : bitstring; (* the user′s biometric *)
    new ri : bitstring;
    let Pi = H1(con(con(PWi,H2(bi)),ri)) in
    let HIDi = H1(xor(IDi,ri) in
    out(sch,(HIDi,IDi,Pi)); (* ------------registration------------ *)
    in(sch,(xBi:bitstring, xVi:bitstring,xEkey:bitstrig,xn:bitstring));
    !
    (
    event UserStarted();
    let Pi′ = H1(con(con(PWi,H2(bi)),ri)) in
    let HIDi′ = H1(xor(IDi,ri)) in
    let Vi′ = mod(H1(xor(Pi′,H1(HIDi’)))xn) in
    if Vi′ = xVi  then
    new N1:bitstring;
    new SIDj:bitsting;
    let Ai′ = xor(xBi,Pi′) in
    let Di = xor(H1(xor(Ai′,HIDi,))N1) in
    let Gi = H1(xor(N1,xor(Ai’HIDi’))) in
    let Ri = mult(Gi,P) in
    let Ci = H1(mult(Gi,Ppub)) in
    let Li = senc(con(con(HIDi′,Ai′),SIDj),Ci) in
    out(ch,(Di,HIDi′,Li)); (*-----------authentication-----------*)
    in(ch,(xRs:bitstring,xFi:bitstring));
    let Ei′ = mult(N1,xRs) in
    let SKi = H1(con(Ei′H1(con(Ai′,Ci)))) in
    let Fi′ = H1(con(con(con(HIDi′SKi),xRs),SIDj)) in
    if Fi′ = xFi then event UserAcServer();
    event UserAuthed();
    0
    ).
```

Figure 5: Process of User$_i$ in ProVerif tool.

```
(*------RC′s process------ *)
let ServerReg=
    in(sch,(rSIDj:bitstring));
    new ej:bitstring;
    let SMj = H1(con(con(rSIDj,x),ej)) in
    out(sch,(SMj));
    0.
let UserReg=
    in(sch,(rHIDi:bitstring,rIDi:bitstring,rPi:bitstring));
    new di:bitstring;
    let Ai=H1(con(con(con(x,rHIDi),rIDi),di)) in
    let Bi=xor(Ai,rPi) in
    new n :bitstring;
    new Ekey:bitstring;
    let Vi=mod(H1(xor(rPi,H1(rHIDi))),n) in
    out(sch,(Bi,Vi,Ekey,n));
    0.
let RCAuth=
    in(ch,(yDI:bitstring,yHIDi:bitstring,yLi:bitstring))
    new IDi:bitstring;
    new yej:bitsting;
    new yej:bitsting;
    let Ai″ = H1(con(con(con(x,yHIDi).IDi,ydi)) in
    let N1′ = xor (yDi,H1(xor(Ai″,yHIDi))) in
    let Gi′ = H1(xor(N1′,xor(Ai″,yHIDi))) in
    let Ri′ = mult(Gi′,P) in
    let Ci′ = H1(mult(x,Ri’)) in
    let DR = sdec(yLi,Ci′) in
    let HIDi = getmess(DR) in
    let SIDj = getmess(DR) in
    let Ai′=getmess(DR) in
    if Ai′=Ai″ then event RCAcUser();
    let SMj = H1(con(con(SIDj,x),yej)) in
    let Yi = H1(con(SIDj,SMj)) in
    let AC = H1con(Ai′,Ci′)) in
    let Mi = senc(con(con(con(HIDi,Ri′),Yi),AC),SMj) in
    out(ch, (Mi));
    0.
let ProcessRC = ServerReg |  UserReg | RCAuth.
```

Figure 6: Process of RC in ProVerif tool.

and all events are executed normally. Note that Figures 4–9 are shown in Appendix.

5.4. Informal Security Analysis

5.4.1. Known Session-Specific Temporary Information Attacks. Upon completing the login and authentication phase, if N_1 or N_2 is compromised, then A intercepts information $\{R_s, F_i\}$ and computes $E_i = N_1R_S$, but it cannot compute $A_i = H(x\| \mathrm{HID}_i\| t\mathrm{ID}_i\| nd_i)$ and $C_i = H(G_iP_{\mathrm{pub}}) = H(H(N_1\oplus A'_i\oplus HID'_i)P_{\mathrm{pub}})$. Therefore, A cannot compute the SK, and the scheme successfully overcomes known session-specific temporary information attacks.

```
(* -----Server′s process----- *)
let ProcessServer =
    new SIDj:bitstring;
    out(sch,(SIDj));
    in(sch,(zSMJ:bitstring));
    !
    (
    in(ch,(zMi:bitstring));
    let DS = sdec(zMi,zSMj) in
    let HIDi′ = getmess(DS) in
    let Ri′ = getmess(DS) in
    let Yi′ = getmess(DS) in
    let AC = getmess(DS) in
    let Yi″ = H1(con(SIDj,zSMj)) in
    if Yi″ = Yi′ then event ServerAcRC();
    new N2:bitstring;
    let Rs = mult(N2,P) in
    let Ei = mult(N2,Ri′) in
    let SKj = H1(con(Ei,AC)) in
    let Fi = H1(con(con(con(HIDi′,SKj),Rs),SIDj)) in
    out (ch,Rs,Fi));
    0
    ).
(* -----main----- *)
process
    (!ProcessUser | !ProcessRC | !ProcessServer)
```

Figure 7: Process of Server$_j$ in ProVerif tool.

```
(* queries *)
query attacker(SKi).
query attacker(SKj).
query attacker(IDi).
query attacker(SIDj).
query inj-event(UserAuthed()) ==> inj-event(UserStarted()).
query inj-event(ServerAcRC()) ==> inj-event(RCAcUser()).
query inj-event(UserAcServer()) ==> inj-event(ServerAcRc()).
(* event *)
event UserStarted().
event UserAuthed().
event RCAcUser().
event ServerAcRC().
event UserAcServer().
```

Figure 8: Queries and events in ProVerif tool.

5.4.2. User Impersonation Attacks. Assume that the A pretends to be a user and forges a message $M_1 = \{D_i, \mathrm{HID}_i, L_i\}$. Even if A forges a random number N'_1, it cannot compute A_i to forge D_i and L_i. A cannot obtain A_i for two reasons. First, upon completing the login and entering the authentication phase, A_i is encrypted by C_i, and A cannot compute C_i to decrypt A_i; therefore, A_i cannot be obtained. Second, in the registration phase, if the SC is stolen by a malicious user, then A can obtain B_i. However, because $A_i = B_i\oplus P_i$, A requires $\{\mathrm{PW}_i, b_i, r_i\}$ to compute P_i, which is impossible. Therefore, the scheme successfully overcomes user impersonation attacks.

5.4.3. Server Impersonation Attacks. Assume that A pretends to be the server and forges a message $M_3 = \{R_s, F_i\}$. Therefore, A must generate a random number N_A and compute $\{R_A = N_AP, E_A = N_2R_i, \mathrm{SK}_A = H(E_A\| Ht(A_i\| C_i))\}$. However, A cannot obtain $\{R_i, A_i, C_i\}$. Even if A can obtain temporary information N_1, it cannot compute $\{R_i, A_i, C_i\}$, nor can it obtain sensitive information by

```
-- Query not attacker(SKi[])
Completing...
Starting query not attacker(SKi[])
RESULT not attacker(SKi[] is true.
-- Query not attacker(SKj[])
Completing...
Starting query not attacker(SKj[])
RESULT not attacker(SKj[] is true.
-- Query not attacker(IDi[])
Completing...
Starting query not attacker(IDi[])
RESULT not attacker(IDi[] is true.
-- Query not attacker(SIDj[])
Completing...
Starting query not attacker(SIDj[])
RESULT not attacker(SIDj[] is true.
-- Query inj-event(UserAuthed) ==> inj-event(UserStarted)
Completing...
200 ruels inserted. The rule base contains 198 rules. 9 ruels in the queue.
Starting query inj-event(UserAuthed) ==> inj-event(UserStarted)
RESULT inj-event(UserAuthed) ==> inj-event(UserStarted) is true.
--Query inj-event(ServerAcRC) ==> inj-event(RCAcUser)
Completing...
Starting query inj-event(ServerAcRC) ==> inj-event(RCAcUser)
RESULT inj-event(ServerAcRC) ==> inj-event(RCAcUser) is true.
-- Query inj-event(UserAcServer) ==> inj-event(ServerAcRC)
Completing...
200 rules inserted. The rule base contains 188 rules. 2 rules in the queue.
Starting query inj-event(UserAcServer) ==> inj-event(ServerAcRC)
RESULT inj-event(UserAcServer) ==> inj-event(ServerAcRC) is true.
```

Figure 9: Results obtained.

decrypting L_i. Therefore, the scheme can overcome server impersonation attacks.

5.4.4. Man-in-the-Middle Attacks. Upon completing the login and authentication phase, the A intercepts the messages transmitted between User$_i$ and Server$_j$ to impersonate the user or server. The A may intercept M_3 to impersonate Server$_j$. However, A cannot compute $F_i = H(\text{HID}_i \| \text{SK}_j \| tR_S \| n\text{SID}_j)$; therefore, the session is terminated. In another case, A may intercept $\{M_1, M_2\}$ to impersonate User$_i$. However, A cannot compute A_i; therefore, it cannot pass the RC verification. Therefore, the scheme can overcome man-in-the-middle attacks.

5.4.5. Replay Attacks. Suppose that message M_1, M_2, or M_3 is replayed by A. However, our scheme overcomes this attack by refreshing random numbers $\{N_1, N_2\}$. By replaying one of the messages $\{M_1, M_2, M_3\}$, the mutual authentication values F_i for the user will not pass, and the session will be terminated. Therefore, this scheme can overcome replay attacks.

5.4.6. Stolen **SC** *Attacks.* Suppose that the SC is stolen by a malicious user A who will obtain $\{B_i, V_i, E_{\text{key}}(\cdot), P, P_{\text{pub}}, n, r_i\}$. However, based on those values, A cannot compute $A_i = H(x \| \text{HID}_i \| t\text{ID}_i \| nd_i)$. In addition, A cannot obtain N_1 to compute $C_i = H(G_i P_{\text{pub}}) = H(H(N_1 \oplus A_i' \oplus \text{HID}_i') P_{\text{pub}})$ and $E_i = N_1 R_S$. Therefore, A cannot compute $\text{SK} = H(E_i \| H(A_i \| C_i))$. Hence, it is clear that the scheme can successfully overcome stolen SC attacks.

5.4.7. Offline Password-Guessing Attacks. According to, A obtains $\{B_i, V_i, E_{\text{key}}(\cdot), P, P_{\text{pub}}, n, r_i\}$. Moreover, A can be biometric b_i by shoulder surfing. A launches an offline password-guessing attack by comparing P_i ($P_i = H(\text{PW}_i \| h(b_i) \| r_i)$). In addition, $P_i = A_i \oplus B_i = H(x \| \text{HID}_i \| t\text{ID}_i \| nd_i) \oplus B_i$. However, A cannot obtain HID$_i$, x, and d_i; therefore, the attacker cannot compute P_i. Hence, the scheme can overcome offline password-guessing attacks.

5.4.8. Privileged Insider Attacks. Assume that the privileged insider A is $\{\text{HID}_i, \text{ID}_i, d_i\}$ stored in the RC database. However, A cannot obtain x and the user's ID$_i$; therefore, it cannot compute $A_i = H(x \| \text{HID}_i \| t\text{ID}_i \| nd_i)$. Because $E_i = N_2 R_i = N_2 H(N_1 \oplus A_i' \oplus \text{HID}_i') P$ and $\text{SK} = H(E_i \| H(A_i \| C_i))$, A cannot compute the SK. Therefore, the scheme can overcome privileged insider attacks.

5.4.9. Perfect Forward Secrecy. Suppose that A obtains the RC's long-term key x and attempts to obtain the SK. If A obtains N_1 and intercepts $\{R_s, F_i\}$, then it computes $E_i = N_1 R_S$. However, A cannot compute $A_i = H(x \| \text{HID}_i \| t\text{ID}_i \| nd_i)$ and $C_i = H(G_i P_{\text{pub}}) = H(H(N_1 \oplus A_i' \oplus \text{HID}_i') P_{\text{pub}})$. In other words, A cannot compute $\text{SK} = H(E_i \| H(A_i \| C_i))$. Therefore, this scheme provides perfect forward secrecy.

5.4.10. User Anonymity. In the registration phase of the improved scheme, User$_i$ computes $\text{HID}_i = H(\text{ID}_i \oplus r_i)$ to protect the real identity of the user. In the authentication phase, the user transmits the virtual identity HID$_i$, and the attacker cannot obtain the real identity of the user. Therefore, our scheme provides user anonymity.

5.4.11. Three-Factor Secrecy. The three factors refer to the password, SC, and biometrics. Based on a previous analysis, A_i and C_i are the key parameters for launching an attack to compute the SK. A obtains two of the three factors, i.e., the password and SC. Even if A obtains the password and extracts the parameters from the SC, it cannot compute A_i and C_i to perform any attack. Passwords and biometrics: if A obtains the password and biometrics to calculate A_i, it must obtain B_i and P_i. However, B_i is stored in an SC, whereas P_i is protected by a random number. Biometrics and smart cards: if A obtains the biometrics and SC to calculate P_i, it must obtain the PW$_i$. Therefore, A cannot compute $A_i = B_i \oplus P_i$.

After analyzing the security of our improved scheme, we can conclude that our proposed scheme is "provably secure" against several well-known attacks with a higher probability. However, it not means that our scheme is a "perfectly secure" authentication scheme because many special attack approaches or tricks exist [19].

Table 3: Security comparison.

Attack methods	Ali and Pal's scheme [34]	Wang et al.'s scheme [36]	Our scheme
User anonymity	√	×	√
Offline password attacks	√	√	√
Stolen smart card attacks	√	√	√
Known session-specific temporary information attack	× [36]	×	√
User impersonation attack	× [36]	×	√
Server impersonation attack	× [36]	×	√
Replay attacks	√	√	√
Perfect forward secrecy	× [36]	√	√
Three-factor secrecy	× [36]	√	√

Note. √, able to overcome the attack, and ×, unable to overcome the attack.

Table 4: Computation cost comparison.

Scheme	User computations	Server computations	*RC*	Total
Ali et al.'s scheme [34]	$3T_m + 4T_p \approx 42.42$ ms	$4T_m + 3T_p + 1T_s \approx 55.56$ ms	$3T_m + 3T_p + 2T_s \approx 42.18$ ms	$10T_m + 10T_p + 3T_s \approx 140.16$ ms
Wang et al.'s scheme [36]	$3T_m + 1T_s \approx 40.62$ ms	$2T_m + 1T_s \approx 27.12$ ms	$1T_m + 2T_s \approx 13.74$ ms	$6T_m + 4T_s \approx 71.01$ ms
Our scheme	$3T_m + 1T_s \approx 40.62$ ms	$2T_m + 1T_s \approx 27.12$ ms	$2T_m + 2T_s \approx 27.24$ ms	$7T_m + 4T_s \approx 94.98$ ms

T_m, time for executing elliptic curve scalar point multiplication. T_p, time for performing elliptic curve point addition operation. T_s, time for executing symmetric encryption/decryption operation.

6. Performance Comparison

In this section, we compare our improved scheme with those of Ali and Pal [34] and Wang et al. [36] in terms of security and efficiency. Table 3 presents a comparison of security among the abovementioned schemes. It is evident that our scheme is secure against well-known attacks. Ali and Pal's scheme [34] could not overcome known session-specific temporary information, user impersonation, and server impersonation attacks, nor could it provide three-factor and perfect forward secrecy. Although Wang et al.'s scheme [36] guaranteed perfect forward secrecy, it could not overcome known session-specific temporary information, user impersonation, and server impersonation attacks. Hence, it is clear that only our proposed protocol successfully overcame all known attacks and achieved a certain degree of security.

A comparison of the computational costs is shown in Table 4. We used JPBC-2.0.0 (Pairing-Based Cryptography Library) [53], IntelliJ IDEA 2020.2.1 community edition, and a Windows 10 computer with a 2.3 GHz Intel (R) Core i5 processor and 16 GB of memory to simulate the computational costs. It is noteworthy that a widely accepted Type A pairing was constructed on the curve $y^2 = x^3 + x$ over F_q, where q is a prime satisfying $q = 3 \bmod 4$. In our experimental results, T_m was 13.5 ms, T_p was 0.48 ms, and T_s was 0.12 ms. As shown in Table 4, the computational cost of our scheme was lower than that of the scheme in [34], whereas it was 13.5 ms higher than that of the scheme in [36]. However, when our scheme was utilized in a practical application, the 13.5 ms difference was almost negligible. Meanwhile, the scheme in [36] was subject to known session-specific temporary information, user impersonation, and server impersonation attacks. However, our improved scheme overcame all known attacks.

Table 5: Comparison of communication and massage rounds.

Scheme	Communication cost (bits)	Massage rounds
Ali and Pal's scheme [34]	3712	4
Wang et al.'s scheme [36]	1664	4
Our scheme	1600	3

Table 5 shows a comparison of the communication costs. We assumed that the ECC points accounted for 320 bits because two 160-bit parameters form an ECC point. The hash operation was considered to be 256 bits, and the identity was 64 bits. The length of the ciphertext for a symmetric encryption was 256 bits. In Ali et al.'s scheme, the messages in the login and authentication phase were $\{DID_i, E_i, C_i, D_i\}, \{DID_i^{new}, K_i, L_i, F_i\}$, $\{DID_i^{new}, Q_i, M_i, K_i\}$, and $\{Z_i\}$, where $\{E_i, C_i, K_i, Q_i, Z_i\}$ belong to ECC, $\{D_i, L_i, M_i\}$ are hash values, and $\{DID_i, DID_i^{new}, F_i\}$ are ciphertexts. The total communication cost of Ali et al.'s scheme was 3712 bits. In Wang et al.'s scheme, the messages in the login and authentication phase were $\{R_i, L_i\}, \{M_i\}$, $\{R_S, F_i\}$, and $\{Q_i\}$, where $\{R_i, R_S\}$ belong to ECC, $\{F_i, Q_i\}$ are hash values, and $\{L_i, M_i\}$ are ciphertexts. The total communication cost of Wang et al.'s scheme was 1664 bits. In our scheme, the messages in the login and authentication phases were $\{D_i, HID_i', L_i\}, \{M_i\}$, and $\{R_S, F_i\}$, where $\{R_S\}$ belongs to ECC, $\{D_i, HID_i', F_i\}$ are hash values, and $\{L_i, M_i\}$ are ciphertexts. The total communication cost of our scheme was 1600 bits.

Through the analysis of computation cost and communication cost, the communication cost of our scheme is significantly lower than [34, 36] and the computation cost is also acceptable. Combined with the previous security analysis mentioned in Table 3, our scheme also has strong security. Hence, our scheme is worthy of being adopted in secure three-factor authentication.

7. Conclusion

In this study, we performed a security analysis of Wang et al.'s scheme and discovered that their scheme could not overcome known session-specific temporary information, user impersonation, and server impersonation attacks. Additionally, we have proven the security of our proposed scheme through formal and informal security analysis. Subsequently, the communication security of our scheme was validated by the ProVerif tool, and the BAN logic indicated that mutual authentication can be completed safely. Finally, through a comparison of performance and security, the security and efficiency of our proposed scheme was proven. However, the computational cost of our scheme is still high. It will lead us to design lightweight authentication schemes in the future.

Data Availability

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

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] J.-S. Pan, P. Hu, and S.-C. Chu, "Novel parallel heterogeneous meta-heuristic and its communication strategies for the prediction of wind power," *Processes*, vol. 7, no. 11, p. 845, 2019.

[2] T.-Y. Wu, C.-M. Chen, K.-H. Wang, C. Meng, and E. K. Wang, "A provably secure certificateless public key encryption with keyword search," *Journal of the Chinese Institute of Engineers*, vol. 42, no. 1, pp. 20–28, 2019.

[3] J. M.-T. Wu, J. C.-W. Lin, and A. Tamrakar, "High-utility itemset mining with effective pruning strategies," *ACM Transactions on Knowledge Discovery from Data*, vol. 13, no. 6, pp. 1–22, 2019.

[4] Z. Meng, J.-S. Pan, and K.-K. Tseng, "PaDE: an enhanced Differential Evolution algorithm with novel control parameter adaptation schemes for numerical optimization," *Knowledge-Based Systems*, vol. 168, pp. 80–99, 2019.

[5] A.-Q. Tian, S.-C. Chu, J.-S. Pan, H. Cui, and W.-M. Zheng, "A compact pigeon-inspired optimization for maximum short-term generation mode in cascade hydroelectric power station," *Sustainability*, vol. 12, no. 3, p. 767, 2020.

[6] S. C. Chu, X. S. Xue, J. S. Pan, and X. J. Wu, "Optimizing ontology alignment in vector space," *Journal of Internet Technology*, vol. 21, no. 1, pp. 15–22, 2020.

[7] Z. G. Du, J. S. Pan, S. C. Chu, H. J. Luo, and P. Hu, "Quasi-affine transformation evolutionary algorithm with communication schemes for application of RSSI in wireless sensor networks," *IEEE Access*, vol. 8, pp. 858–8594, 2020.

[8] J. Zhang, H. Liu, and L. Ni, "A secure energy-saving communication and encrypted storage model based on RC4 for EHR," *IEEE Access*, vol. 8, pp. 38995–39012, 2020.

[9] J. Wang, X. Gu, W. Liu, A. K. Sangaiah, and H. J. Kim, "An empower Hamilton loop based data collection algorithm with mobile agent for WSNs," *Human-centric Computing and Information Sciences*, vol. 9, no. 1, pp. 1–14, 2019.

[10] J. Wang, Y. Gao, K. Wang, A. K. Sangaiah, and S.-J. Lim, "An affinity propagation-based self-adaptive clustering method for wireless sensor networks," *Sensors*, vol. 19, no. 11, p. 2579, 2019.

[11] H. Xiong, Y. Wu, C. Jin, and S. Kumari, "Efficient and privacy-preserving authentication protocol for heterogeneous systems in IIoT," *IEEE Internet of Things Journal*, vol. 1, 2020.

[12] H. Xiong, Y. Zhao, Y. Hou et al., "Heterogeneous signcryption with equality test for IIoT environment," *IEEE Internet of Things Journal*, vol. 1, 2020.

[13] J. H. Yang and C. C. Chang, "An id-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem," *Computers & Security*, vol. 28, no. 3-4, pp. 138–143, 2009.

[14] S. H. Islam and G. P. Biswas, "A more efficient and secure ID-based remote mutual authentication with key agreement scheme for mobile devices on elliptic curve cryptosystem," *Journal of Systems and Software*, vol. 84, no. 11, pp. 1892–1898, 2011.

[15] T. T. Truong, M. T. Tran, and A. D. Duong, "Improvement of the more efficient and secure id-based remote mutual authentication with key agreement scheme for mobile devices on ecc," in *Proceedings of the 2012 26th international Conference on Advanced information Networking and Applications Workshops*, pp. 698–703, IEEE, Fukuoka, Japan, March 2012.

[16] E. Erdem and M. T. Sandıkkaya, "OTPaaS—one time password as a service," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 3, pp. 743–756, 2018.

[17] S. A. Chaudhry, "Correcting "PALK: password-based anonymous lightweight key agreement framework for smart grid," *International Journal of Electrical Power & Energy Systems*, vol. 125, p. 106529.

[18] M. Karuppiah and R. Saravanan, "A secure remote user mutual authentication scheme using smart cards," *Journal of Information Security and Applications*, vol. 19, no. 4, pp. 282–294, 2014.

[19] D. Wang, D. B. He, P. Wang, and C. H. Chu, "Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 4, pp. 428–442, 2014.

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

[21] R. Amin, S. H. Islam, G. P. Biswas, M. K. Khan, and N. Kumar, "An efficient and practical smart card based anonymity preserving user authentication scheme for TMIS using elliptic curve cryptography," *Journal of Medical Systems*, vol. 39, no. 11, pp. 1–18, 2015.

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

[23] C.-M. Chen, B. Xiang, Y. Liu, and K.-H. Wang, "A secure authentication protocol for Internet of vehicles," *IEEE Access*, vol. 7, no. 1, pp. 12047–12057, 2019.

[24] T.-Y. Wu, Z. Lee, M. S. Obaidat, S. Kumar, C.-M. Sachin, and C. M. Chen, "An authenticated key exchange protocol for

multi-server architecture in 5G networks," *IEEE Access*, vol. 8, pp. 28096–28108, 2020.

[25] D. Wang, Q. Gu, H. Cheng, and P. Wang, "The request for better measurement: a comparative evaluation of two-factor authentication schemes," in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, pp. 475–486, Xi an, China, June 2016.

[26] D. Wang, W. Li, and P. Wang, "Measuring two-factor authentication schemes for real-time data access in industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4081–4092, 2018.

[27] R. S. Pippal, C. D. Jaidhar, and S. Tapaswi, "Robust smart card authentication scheme for multi-server architecture," *Wireless Personal Communications*, vol. 72, no. 1, pp. 729–745, 2013.

[28] D. He and D. Wang, "Robust biometrics-based authentication scheme for multiserver environment," *IEEE Systems Journal*, vol. 9, no. 3, pp. 816–823, 2014.

[29] X. Huang, Y. Xiang, E. Bertino, J. Zhou, and L. Xu, "Robust multi-factor authentication for fragile communications," *IEEE Transactions on Dependable and Secure Computing*, vol. 11, no. 6, pp. 568–581, 2014.

[30] 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.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] S. Hussain and S. A. Chaudhry, "Comments on "Biometrics-Based privacy-preserving user authentication scheme for cloud-based industrial Internet of Things deployment"" *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10936–10940, 2019.

[36] F. Wang, G. Xu, C. Wang, and J. Peng, "A provably secure biometrics-based authentication scheme for multiserver environment," *Security and Communication Networks*, vol. 2019, pp. 1–15, Article ID 2838615, 2019.

[37] C. ., M. Chen, Y. Y. Huang, K. H. Wang, S. Kumari, and M. E. Wu, "A secure authenticated and key exchange scheme for fog computing," *Enterprise Information Systems*, vol. 2020, pp. 1–16, 2020.

[38] H. Luo, G. Wen, and J. Su, "Lightweight three factor scheme for real-time data access in wireless sensor networks," *Wireless Networks*, vol. 26, no. 2, pp. 955–970, 2020.

[39] C. Shehzad Ashraf, S. Taeshik, A. T. Fadi, and H. A. Mohammed, "Correcting design flaws: an improved and cloud assisted key agreement scheme in cyber physical systems," *Computer Communications*, vol. 153, pp. 527–537, 2020.

[40] C.-G. Ma, D. Wang, and S.-D. Zhao, "Security flaws in two improved remote user authentication schemes using smart cards," *International Journal of Communication Systems*, vol. 27, no. 10, pp. 2215–2227, 2014.

[41] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 453–474, Springer, Austria, May 2001.

[42] O. Goldreich and S. Halevi, "The random oracle methodology, revisited," in *Proceedings of the 30th ACM Symposium. Theory of Computing (STOC)*, pp. 209–218, Chicago, IL, USA, June 1998.

[43] X. Huang, X. Chen, J. Li, Y. Xiang, and L. Xu, "Further observations on smart-card-based password-authenticated key agreement in distributed systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 7, pp. 1767–1775, 2013.

[44] D. Wang, N. Wang, P. Wang, and S. Qing, "Preserving privacy for free: efficient and provably secure two-factor authentication scheme with user anonymity," *Information Sciences*, vol. 321, pp. 162–178, 2015.

[45] 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, 2016.

[46] P. Gope, A. K. Das, N. Kumar, and Y. Cheng, "Lightweight and physically secure anonymous mutual authentication protocol for real-time data access in industrial wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 9, pp. 4957–4968, 2019.

[47] S. Qiu, D. Wang, G. Xu, and S. Kumari, "Practical and provably secure three-factor Authentication protocol based on extended chaotic-maps for mobile lightweight devices," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, 2020.

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

[49] 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.

[50] C.-M. Chen, C.-T. Li, S. Liu, T.-Y. Wu, and J.-S. Pan, "A provable secure private data delegation scheme for mountaineering events in emergency system," *IEEE Access*, vol. 5, pp. 3410–3422, 2017.

[51] C.-T. Li, T.-Y. Wu, and C.-M. Chen, "A provably secure group key agreement scheme with privacy preservation for online social networks using extended chaotic maps," *IEEE Access*, vol. 6, pp. 66742–66753, 2018.

[52] C.-M. Chen, K.-H. Wang, K.-H. Yeh, B. Xiang, and T.-Y. Wu, "Attacks and solutions on a three-party password-based authenticated key exchange protocol for wireless communications," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 8, pp. 3133–3142, 2019.

[53] A. De Caro, V. Iovino, and jP. B. C. ", "Java pairing based cryptography," in *Proceedings of the 2011 IEEE Symposium on Computers and Communications (ISCC)*, pp. 850–855, Kerkyra, Greece, July 2011.