Hindawi
Wireless Communications and Mobile Computing
Volume 2020, Article ID 8886585, 11 pages
https://doi.org/10.1155/2020/8886585

Research Article

An Anonymous and Efficient ECC-Based Authentication Scheme for SIP

Yousheng Zhou [1,2] **and Xinyun Chen** [1]

[1]*College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*
[2]*School of Cyber Security and Information Law, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*

Correspondence should be addressed to Yousheng Zhou; zhouys@cqupt.edu.cn

Received 25 March 2020; Revised 11 July 2020; Accepted 27 October 2020; Published 21 November 2020

Academic Editor: Mario Kolberg

Session initiation protocol (SIP), a widely used signal protocol for controlling multimedia communication sessions, is under numerous attacks when performing the authentication steps between the user and server. So secure authentication schemes are needed to be presented for SIP. Recently, Arshad et al. advanced novel schemes for SIP using elliptic curve cryptography (ECC) and claimed their schemes can resist various attacks. However, Lu et al. found that Arshad et al.'s scheme cannot resist trace and key-compromise impersonation attacks; hence, it cannot provide proper mutual authentication. Meanwhile, an enhanced scheme was advanced by Lu et al. and they stated that their scheme can stand up to possible known attacks. Nevertheless, in this paper, we conclude that Arshad and Nikooghadam's scheme is insecure against impersonation attack and Lu et al.'s scheme is still vulnerable to impersonation attack. To overcome these weaknesses of their schemes, we present a novel anonymous ECC-based scheme for SIP. Security analysis and performance analysis show that our proposed scheme can resist various known attacks and efficient in the meantime.

1. Introduction

SIP (session initiation protocol), a text-based application layer signaling control protocol, is used to create, modify, and release sessions between participators. These sessions will be initiated when users request Internet multimedia conferences, IP phones, and multimedia distribution. The participants of SIP can communicate with each other by multicast, unicast, or a mixture of two. SIP is widely used since 2002, the time when it was presented by the Internet Engineering Task Force (IETF) [1]. To protect the privacy of users, it is critical for SIP to provide mutual authentication between communicating parties. Therefore, many researchers devote to proposing secure and efficient schemes for SIP to prevent various attacks and provide mutual authentication between a legal user and server nowadays.

In 2009, Tsai [2] presented a scheme based on random nonce for SIP. He used one-way hash functions and exclusive or operations to encrypts/decrypts all the necessary information. So Tsai's scheme can be used in low-computation equipment because its computation cost is very low. Later, Yoon et al. [3] demonstrated that Tsai's scheme is not secure against off-line password guessing attack, Denning-Sacco attack, and stolen-verifier attack and cannot provide perfect forward secrecy. To overcome the shortcomings of Tsai's scheme, Yoon et al. proposed a scheme based on the elliptic curve discrete logarithm problem (ECDLP) for SIP and they claimed their scheme can resist various attacks while providing more efficiency than Tsai's scheme. In 2012, Xie [4] proposed an improved scheme after finding Yoon et al.'s scheme is still too weak to resist stolen-verifier attack and off-line password guessing attack. Shortly afterwards, Farash and Attari [5] demonstrated that Xie's scheme still suffers from off-line password guessing attack and impersonation attack and proposed an enhanced scheme. Later on, Zhang et al. [6] proposed an authentication scheme with anonymity for SIP based on Farash and Attari's work. However, Lu et al. [7] found that Zhang et al.'s scheme cannot provide proper security, because it is insecure against insider attack. To cover the demerits of Zhang et al.'s scheme, Lu et al. advanced a

new scheme and they demonstrate that their scheme is resistant to possible known attacks while having lower computation cost than other related schemes. In 2016, Chaudhry et al. [8] stated that Lu et al.'s scheme cannot withstand user and sever impersonation attacks, so they proposed their own enhanced scheme to correct these problems. However, Kumari et al. [9] suspected that the Chaudhry et al.'s scheme still has the disadvantages that appeared in Lu et al.'s. Meanwhile, Kumari et al. showed that Lu et al.'s [7] scheme cannot resist impersonation and identity guessing attacks.

In 2014, a smart-card-based scheme was advanced by Zhang et al. [10] to overcome the weaknesses of previous schemes. When a legal user attempts to communicate with the server, he must use the smart card as another authentication factor in addition to the password to achieve authentication. Later, Irshad et al. [11] demonstrated that Zhang et al.'s scheme is vulnerable to denial of service (DOS) attack and impersonation attack and advanced an improved scheme while optimizing the cost in their protocol. However, Irshad et al.'s scheme was suspected of being unable to resist user impersonation attack by Arshad and Nikooghadam [12], and Arshad and Nikooghadam advanced a new scheme in their paper. Unfortunately, Lu et al. in [13] found that Arshad et al.'s scheme is still insecure against some attacks, such as key-compromise impersonation attack and trace attack. In order to correct the shortcomings of Arshad and Nikooghadam's scheme, Lu et al. proposed a robust and efficient authentication scheme by using ECC and demonstrated that their scheme is resistant to possible known attacks. Recently, we find that Arshad and Nikooghadam's [12] scheme cannot resist user impersonation attack. Meanwhile, we observe that Lu et al.'s [13] scheme is insecure against server impersonation attack.

2. Motivations and Contributions

In this paper, we revisit Arshad and Nikooghadam and Lu et al.'s schemes and show that their schemes are vulnerable to impersonation attack. Meanwhile, we propose our enhanced ECC-based scheme for SIP to make up for the shortcomings of Arshad et al.'s and Lu et al.'s schemes.

The rest of the paper is organized as follows. Review and cryptanalysis of Arshad and Nikoofhadam's scheme are showed in Sections 3 and 4, separately. Review and cryptanalysis of Lu et al.'s scheme will be put in in Sections 5 and 6, separately. In Section 7, we present our scheme. Security analysis and performance analysis are showed in Sections 8 and 9, separately. Finally, conclusion of this paper is shown in Section 10.

3. Review of Arshad and Nikoofhadam's Scheme

In this section, we will review Arshad and Nikoofhadam's [12] scheme briefly. Firstly, we will list the notations that were used throughout Arshad et al.'s scheme in Figure 1. Then, we will use four parts to review Arshad et al.'s scheme, including setup phase, registration phase, authentication and key agreement phase, and password change phase.

3.1. Setup. Firstly, the server selects an elliptic curve equation $E_p(a, b)$ and a secure one-way function $h()$. Then, the server selects a base point P with order n over $E_p(a, b)$, chooses a integer k_s randomly and keeps it as a secret key, and computes public key $K_s = k_s P$. Finally, the server publishes $(E_p(a, b), n, P, h(), K_s)$.

3.2. Registration

(1) The client generates a number Nc randomly, chooses a password PW_i, computes $v_i = h(\mathrm{ID}_i \| PW_i \| N_c)$, sends (ID_i, v_i) to the server, and stores N_c in the memory device

(2) If ID_i does not exit in database, the server computes $V_i = h(ID_i \| k_s) \oplus v_i$ and stores it in his/her database

3.3. Authentication and Key Agreement. In this part, we will introduce the authentication and key agreement phase of Arshad and Nikoofhadam's scheme and the steps of this phase are also represented by Table 1.

(1) The client selects an integer d_c randomly, computes $R_c = d_c K_s = d_c k_s P$ and sends $\mathrm{REQUEST}(ID_i, R_c)$ to the server through the public channel

(2) If ID_i exits in database, the server selects a integer d_s randomly, computes $Q_s = d_s P$, $Q_{sc} = d_s k_s^{-1} R_c$, $V_s = h(ID_i \| Q_s \| Q_{sc})$, and sends $\mathrm{CHALLENGE}(\mathrm{realm}, Q_s, V_s)$ to the client. If ID_i does not exit in database, the server terminates the session

(3) The client computes $Q_{cs} = d_c Q_s$ and compares the value of $h(\mathrm{ID}_i \| Q_s \| Q_{cs})$ and V_s. If $h(\mathrm{ID}_i \| Q_s \| Q_{cs})$ and V_s are not equal, the session will be stopped by the client. Otherwise, the client computes $V_c = h(\mathrm{ID}_i \| Q_s \| \mathrm{realm} \| Q_{cs} \| v_i)$ and $\mathrm{SK} = h(\mathrm{ID}_i \| Q_s \| Q_{cs} \| \mathrm{realm})$. Then, $\mathrm{RESPONSE}(\mathrm{ID}_i \| \mathrm{realm} \| V_c)$ is sent to the server

(4) The server computes $v_i = V_i \oplus h(\mathrm{ID}_i \| k_s)$ and compares the value of $h(\mathrm{ID}_i \| Q_s \| \mathrm{realm} \| Q_{sc} \| v_i)$ and V_c. If $h(\mathrm{ID}_i \| Q_s \| \mathrm{realm} \| Q_{sc} \| v_i)$ is equal to V_c, the server authenticates the client

3.4. Password Change

(1) Firstly, a new password PW_i^* is chosen by the client. Then, the client generates a new random number N_c^*, computes $v_i^* = h(\mathrm{ID}_i \| \mathrm{PW}_i^* \| N_c^*)$, $z = h(\mathrm{ID}_i \| \mathrm{PW}_i \| N_c) \oplus v_i^*$, $Z = z \oplus h(\mathrm{ID}_i \| \mathrm{SK})$, and sends $\mathrm{CHANGEPWD}(\mathrm{ID}_i, Z, V_z)$ to server

(2) The server computes $v_i = V_i \oplus h(\mathrm{ID}_i \| k_s)$, $v_i^* = Z \oplus v_i \oplus h(\mathrm{ID}_i \| SK)$, and verify whether $h(\mathrm{ID}_i \| v_i^* \| \mathrm{SK} \| v_i)$ is equal to V_z or not. If they are equal, the server computes $V_i^* = V_i \oplus z$ and replaces V_i with V_i^* in the database. Then, the server sends $\mathrm{ACCEPT}(h(\mathrm{ID}_i \| v_i \| \mathrm{accept} \| v_i^* \| SK))$ to the client

Notations	Definitions
$E_p(a, b)$	An elliptic curve with prime order n
P	The base point of the elliptic curve
xP	The elliptic curve point multiplication defined as $xP = \underbrace{P + P + \ldots P + P}_{xtimes}$
$h()$	A one-way hash function h.
k_s	The secret key of server
K_s	The public key of server, K_s
PW_i	The password of i_{th} client
ID_i	The identity of i_{th} client
$\|\|$	The concatenation operation
$\bigoplus$	The bit-wise exclusive-or(XOR) operation

Figure 1: Notations of Arshad et al.'s scheme.

Table 1: Authentication and key agreement.

Client		Server
Chooses a random integer $d_c \in {}_R Z_p^*$		
Computes $R_c = d_c K_s = d_c k_s P$		
	$\xrightarrow{REQUEST\ (ID_i, R_c)}$	
		Chooses a random integer $d_s \in {}_R Z_p^*$
		Computes $Q_s = d_s P$
		Computes $Q_{sc} = d_k d_s^{-1} R_c = d_s d_c P$
		Computes $V_s = h(ID_i \| Q_s \| Q_{sc})$
	$\xleftarrow{CHALLENGE\ (realm, Q_s, V_s)}$	
Computes $Q_{cs} = d_c Q_s$		
Checks $h(ID_i \| Q_s \| Q_{cs}) \stackrel{?}{=} V_s$		
Computes $v_i = h(ID_i \| PW_i \| N_c)$		
Computes $V_c = h(ID_i \| Q_s \| realm \| Q_{cs} \| v_i)$		
Computes $SK = h(ID_i \| Q_s \| Q_{cs} \| realm)$		
	$\xrightarrow{RESPONSE\ (ID_i, realm, V_c)}$	
		Computes $v_i = V_i \bigoplus h(ID_i \| k_s)$
		Checks $h(ID_i \| Q_s \| realm \| Q_{sc} \| v_i) \stackrel{?}{=} V_c$
		Computes $SK = h(ID_i \| Q_s \| Q_{sc} \| realm)$

(3) The client calculates $h(\mathrm{ID}_i \| v_i \| \text{accept} \| v_i^* \| \mathrm{SK})$ and if it is equal to the ACCEPT, the client replaces N_c with N_c^* in the memory device

4. Cryptanalysis of Arshad and Nikoofhadam's Scheme

In this part, we will prove that Arshad and Nikoofhadam's scheme cannot withstand server impersonate attack. To do so, the adversary $\mathscr{A}$ performs the following steps.

Step 1. Suppose $\mathscr{A}$ obtains $\mathrm{REQUEST}(\mathrm{ID}_i, R_c)$ when a client wants to communicate with the server. Then, $\mathscr{A}$ forges $Q_s' = K_s$ and $Q_{sc}' = R_c = d_c K_s$, where K_s is the server's public key. Then, $\mathscr{A}$ computes $V_s' = h(\mathrm{ID}_i \| Q_s' \| Q_{sc}^*)$ and sends $\mathrm{CHALLENGE}(\mathrm{realm}, Q_s', V_s')$ to the client.

Step 2. After receiving CHALLENGE, the client computes $Q_{cs} = d_c Q_s'$. Since $Q_s' = K_s$, $Q_{cs} = d_c$, $K_s = Q_{sc}'$. Thus, $h(\mathrm{ID}_i \| Q_s' \| Q_{cs}) = V_s$ and the verification will hold. The client authenticates the "server." Then, the client computes $v_i = h(\mathrm{ID}_i \| \mathrm{PW}_i \| N_c)$, $V_c = h(\mathrm{ID}_i \| Q_s' \| \mathrm{realm} \| Q_{csv_i})$ and $\mathrm{SK} = h(\mathrm{ID}_i \| Q_s' \| Q_{cs} \| \mathrm{realm})$. Finally, the client sends $\mathrm{RESPONSE}(\mathrm{ID}_i \| Q_s' \| Q_{cs} \| \mathrm{realm})$ to $\mathscr{A}$.

Step 3. After receiving $\mathrm{RESPONSE}(\mathrm{ID}_i \| Q_s' \| Q_{cs} \| \mathrm{realm})$, $\mathscr{A}$ does not need to compute v_i and verify whether $V_c = h(\mathrm{ID}_i \| Q_s' \| \mathrm{realm} \| Q_{csv_i}) = V_c$ or not. $\mathscr{A}$ only need to computes $\mathrm{SK} = h(\mathrm{ID}_i \| Q_s' \| Q_{cs} \| \mathrm{realm})$ and then can make sure that he/she shares the same SK with the victim client.

From what has been discussed above, we can come to the conclusion that Arshad and Nikoofhadam's scheme cannot resist server impersonation attack.

5. Review of Lu et al.'s Scheme

In this part, Lu et al.'s [13] scheme will be reviewed. And the notations that were used throughout their scheme will be showed in Figure 2.

5.1. Registration

Notations	Definitions
U_i	User
S	Server
$h()$	A one-way hash function h
k_s	The secret key of server
k_{U_i}	The secret key of U_i
PW_i	The password of user U_i
ID_i	The identity user U_i
$\|\|$	The concatenation operation
$\oplus$	The bit-wise exclusive-or(XOR) operation

Figure 2: Notations of Lu et al.'s scheme.

Table 2: Authentication and key agreement.

U_i		S
Generates r_2		
Computes $T = h(ID_i\|\|r_1\|\|PWD)$, $R = r_2P$		
Computes $M_1 = T \cdot r_2P$		
Computes $AID = ID_i \oplus T$		
Computes $M_2 = h(R\|\|ID_i)$		
	$REQUEST\ (M_1, AID, M_2)$ $\rightarrow$	
		Computes $T = VPW \oplus h(k_s)$
		Computes $ID_i = AID \oplus T$, $R = T^{-1}M_1$
		Checks $h(R\|\|ID_i) \stackrel{?}{=} M_2$
		Generates r_3
		Computes $H = r_3P$, $M_3 = ID_i \oplus H$
		Computes $SK_s = r_3R$, $Auth_s = h(SKs\|\|T\|\|R)$
	$\leftarrow$ CHALLENGE $(realm, M3, Auth_s)$	
Computes $H = M_3 \oplus ID_i$		
Computes $SK_{U_i} = r_2H$		
Checks $h(SK_{U_i}\|\|T\|\|R) \stackrel{?}{=} Auth_s$		
Computes $Auth_{U_i} = h(SK_{U_i}\|\|T\|\|H)$		
	$RESPONSE\ (realm, Auth_{U_i})$ $\rightarrow$	
		Checks $h(SK_s\|\|T\|\|H) \stackrel{?}{=} Auth_{U_i}$

(1) U_i chooses a password PW_i, selects secret key k_{U_i}, generates a number r_1 randomly, computes $\mathrm{PWD} = h(\mathrm{PW}_i\|k_{U_i})$ and sends $S\{\mathrm{ID}_i, r_1, \mathrm{PWD}\}$

(2) S calculates $\mathrm{VPW} = h(\mathrm{ID}_i\|\mathrm{PWD}\|r_1) \oplus h(k_s)$ and stores VPW in database

5.2. Authentication and Key Agreement. In this part, we will briefly introduce the authentication and key agreement phase of Lu et al.'s scheme and the steps of this phase are also represented by Table 2.

(1) U_i generates a number r_2 randomly, then computes $T = h(\mathrm{ID}_i\|\mathrm{PWD}\|r_1)$, $R = r_2P$, $M_1 = Tr_2P$, $\mathrm{AID} = ID_i \oplus T$, and $M_2 = h(\mathrm{ID}_i\|R)$. Finally, U_i sends S $\mathrm{REQUEST}(M_1, \mathrm{AID}, M_2)$

(2) S calculates $T = \mathrm{VPW} \oplus h(k_s)$, then computes $\mathrm{ID}_i' = \mathrm{AID} \oplus T'$, $R' = T^{-1}M_1$ and checks whether $M_2' = h(\mathrm{ID}_i'\|R')$ equals to M_2 or not. If they are equal, S generates a number r_3 randomly, computes $H = r_3P$, $M_3 = \mathrm{ID}_i \oplus H$, $\mathrm{SK}_s = r_3P$, and $\mathrm{Auth}_s = h(\mathrm{SK}_s\|T\|R)$ and sends $\mathrm{CHALLENGE}(\mathrm{realm}, M_3, \mathrm{Auth}_s)$ to U_i

(3) U_i computes $H = M_3 \oplus \mathrm{ID}_i$, $\mathrm{SK}_{U_i} = r_2H$ and verifies whether $\mathrm{Auth}_s' = h(\mathrm{SK}_{U_i}\|T\|R)$ is equal to Auth_s. If they are equal, U_i computes $\mathrm{Auth}_{U_i} = h(\mathrm{SK}_{U_i}\|T\|H)$ and then sends $\mathrm{RESPONSE}(\mathrm{realm}, \mathrm{Auth}_{U_i})$ to S

(4) S checks whether $\mathrm{Auth}_{U_i}' = h(\mathrm{SK}_s\|T\|H)$ equals to Auth_{U_i}. If the equation holds, S and U_i share the session key $\mathrm{SK} = \mathrm{SK}_{U_i} = \mathrm{SK}_s$

5.3. Password Change

(1) U_i sends the message $h(h(\mathrm{ID}_i\|r_1\|h(\mathrm{PW}_i\|r_1)\|\mathrm{SK}))$ and $h(\mathrm{ID}_i\|r_1^{\mathrm{new}}\|h(\mathrm{PW}_i^{\mathrm{new}}))$ to S

(2) If $h(h(\mathrm{ID}_i\|r_1\|h(\mathrm{PW}_i\|r_1)\|\mathrm{SK}))$ is equal to the value of $h(h(\mathrm{VPW} \oplus h(k_s))\|\mathrm{SK})$ that S just calculated, S then computes $\mathrm{VPW}^{\mathrm{new}} = \mathrm{SK} \oplus h(\mathrm{ID}_i\|r_1^{\mathrm{new}}\|h(\mathrm{PW}_i^{\mathrm{new}}\|r_1^{\mathrm{new}})) \oplus \mathrm{SK} \oplus h(k_s)$ and then replaces VPW with $\mathrm{VPW}^{\mathrm{new}}$

6. Cryptanalysis of Lu et al.'s Scheme

In this part, we will analyze the security of Lu et al.'s scheme and prove that their scheme cannot resist user impersonation

Notations	Definitions
U_i	User
S	Server
s_2, s_2	The secret key of server
$h()$	A one-way hash function h.
K_P	The public key of server, $K_p = k_2P$
PW_i	The password of user U_i
ID_i	The identity user U_i
r_u, r_s	The random generated by user and server
$\|\|$	The concatenation operation
$\bigoplus$	The bit-wise exclusive-or(XOR) operation

Figure 3: Notations of our scheme.

Table 3: Steps of the registration phase.

U_i		S
Chooses a password PW_i Chooses a random number γ Computes $v_i = h(h(ID_i) \bigoplus h(PWi) \bigoplus h(\gamma))$		
	$\xrightarrow{(ID_i, v_i)}$	
		Computes $V_i = v_i \bigoplus h(s_2)$ Computes $Y = h(v_i \| s_1) S_2^{-1} P$ Computes $V_j = h(ID_i \| V_i \| s_1)$ Stores (V_i, V_j)
	$\xleftarrow{(Y)}$	
Stores (Y, γ) in memory device		

attack. To do so, suppose an adversary $\mathcal{A}$ obtains ID_i of a legal user U_i and intercepts $\mathrm{REQUEST}(M_1, \mathrm{AID}, M_2)$ when U_i wants to send it to S. $\mathcal{A}$ can obtains T by computing $T = \mathrm{AID} \oplus \mathrm{ID}_i$, then masquerades as U_i to communicate with S by following steps.

Step 1. $\mathcal{A}$ generates a number r_2^* randomly, computes $R^* = r_2^* P$, $M_1^* = T \cdot r_2^* P$, $\mathrm{AID} = \mathrm{ID}_i \oplus T$, and $M_2^* = h(R^* \| \mathrm{ID}_i)$. Then, $\mathcal{A}$ sends $\mathrm{REQUEST}(M_1^*, \mathrm{AID}, M_2^*)$ to S.

Step 2. S computes $T = \mathrm{VPW} \oplus h(k_s)$, $\mathrm{ID}_i = \mathrm{AID} \oplus T$, and $R^* = T^{-1} M_1^*$ and checks if $h(R^* \| \mathrm{ID}_i) \stackrel{?}{=} M_2^*$. Obviously, this equation is established. So S authenticates the attacker as U_i. Then, S generates r_3 and computes $H = r_3 P$, $M_3 = \mathrm{ID}_i \oplus H$, $\mathrm{SK}_s^* = r_3 R^*$, and $\mathrm{Auth}_s^* = h(\mathrm{SK}_s \| T \| R^*)$ and sends $\mathrm{CHALLENGE}(\mathrm{realm}, M_3, \mathrm{Auth}_s^*)$ to $\mathcal{A}$.

Step 3. $\mathcal{A}$ computes $H = M_3 \oplus \mathrm{ID}_i$, $\mathrm{SK}_{U_i}^* = r_2^* H$, and checks whether $h(\mathrm{SK}_{U_i}^* \| T \| R^*)$ equals to Auth_s^*. If they are equal, then $\mathcal{A}$ calculates $\mathrm{Auth}_{U_i}^* = h(\mathrm{SK}_{U_i}^* \| T \| H)$ and sends $\mathrm{RESPONSE}(\mathrm{realm}, \mathrm{Auth}_{U_i}^*)$ to S.

Step 4. S checks whether $h(\mathrm{SK}_s^* \| T \| H)$ equals to $\mathrm{Auth}_{U_i}^*$. Obviously, this equation is established. So the attacker $\mathcal{A}$ shares the same session key $\mathrm{SK} = \mathrm{Auth}_{U_i}^* = \mathrm{Auth}_s^*$ with S.

From what has been discussed above, we can come to the conclusion that Lu et al.'s scheme cannot resist user impersonation attack.

7. Our Proposed Scheme

An enhanced scheme for SIP will be advanced in this section. Our scheme is based on the schemes of Irshad et al and Lu et al. and has corrected the problem that appeared in their schemes. We will list the notations that used throughout our scheme in Figure 3. The content of our scheme will be shown as follows:

7.1. Setup Phase. Firstly, an elliptic curve equation $E_p(a, b)$ and a secure one-way function $h()$ are selected by the server. Then, S chooses a base point P with order n over $E_p(a, b)$ and two random numbers k_1,k_2, computes public key $K_p = s_2 P$. Finally, S keeps s_1, s_2 as its secret keys, publishes $(E_p(a, b), n, P, h(), K_p)$.

7.2. Registration. The registration phase will be shown in Table 3 and the steps for user registration are as follows:

Step 1. The user U_i chooses number γ randomly, chooses a password PW_i, computes $v_i = h(h(\mathrm{ID}_i) \oplus h(\mathrm{PW}_i) \oplus h(\gamma))$. Then, U_i sends (ID_i, v_i) to S through a secure channel.

Step 2. S calculates $V_i = v_i \oplus h(s_2)$, $Y = h(v_i \| s_1) s_2^{-1} P$, and $V_j = h(ID_i \| V_i \| s_1)$. Then S stores V_i, V_j and sends Y to U_i through a secure channel.

Step 3. U_i keeps (Y, γ) in the memory device.

7.3. Authentication and Key Agreement. When a legal user U_i attempts to obtain a session key shared with S, the following

Table 4: Authentication and key agreement.

U_i		S
Generates time stamp t		
Generates a random integer $r_u \in_R Z_p^*$		
Computes $v_i = h(h(ID_i) \bigoplus h(PW_i) \bigoplus h(\gamma))$		
Computes $H_j = h(v_i \| Y \| r_u P \| t)$		
Computes $X = r_u Y = r_u s_2^{-1} h(v_i \| s1) P$		
Computes $PID = ID_i \bigoplus v_i$		
	$\xrightarrow{REQUEST(PID, X, H_j, t)}$	
		Checks validity of t
		Computes $ID_i' = V_i \bigoplus h(s_2) \bigoplus PID$
		Checks $V_j' = h(ID_i' \| V_i \| s_1) \stackrel{?}{=} V_j$
		Computes $v_i' = V_i \bigoplus h(s_2)$
		Computes $Y' = h(v_i \| s_1) s_2^{-1} P$
		Computes $Q = X \cdot s_2 \cdot h^{-1}(v_i' \| s_1) = r_u P$
		Checks $H_j' = h(v_i' \| Y' \| Q \| t) = H_j$
		Computes $Q_{sc} = r_s Q = r_u r_s P$
		Computes $SK = h(ID_i \| v_i' \| Q_{sc} \| realm)$
		Computes $\mu = r_s P$
		Computes $\eta = h(SK \| Q \| realm \| v_i')$
	$\xleftarrow{RESPONSE(\mu, realm, \eta)}$	
Computes $Q' = r_u P$		
Computes $Q'_{sc} = r_u \mu = r_u r_s P$		
Computes $SK = h(ID_i \| v_i \| Q'_{sc} \| realm)$		
Checks $\eta' = h(SK \| Q' \| realm \| v_i) \stackrel{?}{=} \eta$		
Computes $Auth = h(SK \| \mu \| v_i)$		
	$\xrightarrow{Confirm(Auth, realm)}$	
		Checks $Auth' = h(SK \| r_s P \| v_i') \stackrel{?}{=} Auth$

steps will be performed. Meanwhile, the details of this phase will be shown in Table 4.

Step 1. The user U_i selects a integer $r_u \in_R Z_p^*$ randomly, generates a time stamp t, and then computes $v_i = h(h(\mathrm{ID}_i) \oplus h(\mathrm{PW}_i) \oplus h(\gamma))$, $H_j = h(v_i \| Y \| r_u P \| t)$, and $X = r_u Y = r_u s_2^{-1} h(v_i \| s_1) P$. Finally, U_i computes $\mathrm{PID} = \mathrm{ID}_i \oplus v_i$ and sends $\mathrm{REQUEST}(\mathrm{PID}, X, H_j, t)$ to S.

Step 2. S checks the validity of time stamp t by checking the validity of the predicate $(t' - t <^? \Delta t)$, and abort if the check fails. Then, S computes $\mathrm{ID}_i' = V_i \oplus h(s_2) \oplus \mathrm{PID}$, $V_j' = h(\mathrm{ID}_i' \| V_i \| s_1)$ and compares the values of V_j' and the stored V_j. If they are equal, S can make sure that the received ID_i and V_i is a pair. After that, S computes $v_i' = V_i \oplus h(s_2)$, $Y' = h(v_i' \| s_1) s_2^{-1} P$, and $Q = X \cdot s_2 \cdot h^{-1}(v_i' \| s_1) = r_u P$ and checks whether $H_j' = h(v_i' \| Y' \| Q \| t)$ equals to H_j or not. If they are equal, S authenticates the user U_i. Then, S chooses a integer $r_s \in_R Z_p^*$ randomly, computes $Q_{sc} = r_s Q = r_u r_s P$, $\mathrm{SK} = h(\mathrm{ID}_i \| v_i' \| Q_{sc} \| \mathrm{realm})$, $\mu = r_s P$, and $\eta = h(\mathrm{SK} \| Q \| \mathrm{realm} \| v_i')$. Finally, S sends $\mathrm{RESPONSE}(\mu, \mathrm{realm}, \eta)$ to U_i.

Step 3. U_i computes $Q' = r_u P$, $Q'_{sc} = r_u \mu = r_u r_s P$, and $\mathrm{SK} = h(\mathrm{ID}_i \| v_i \| Q'_{sc} \| \mathrm{realm})$ and then checks whether $\eta' = h(\mathrm{SK} \| Q' \| \mathrm{realm} \| v_i)$ equals to received η. If they are equal, U_i authenticates S. Finally, U_i computes $\mathrm{Auth} = h(\mathrm{SK} \| \mu \| v_i)$ and sends message $\mathrm{Confirm}(\mathrm{Auth}, \mathrm{realm})$ to S.

Step 4. S calculates $\mathrm{Auth}' = h(\mathrm{SK} \| r_s P \| v_i')$ and compares the values Auth' and the received Auth. S confirms that he/she shares the same session key SK with U_i if Auth' is equal to Auth.

7.4. Password Change

Step 1. U_i chooses a figure γ^* randomly, selects a new password PW_i^*, then U_i computes $v_i^* = h(h(\mathrm{ID}_i) \oplus h(\mathrm{PW}_i^*) \oplus h(\gamma^*))$, $I = Y \oplus v_i^*$ and sends $E_{\mathrm{SK}}(I \| v_i \| \mathrm{ID}_i)$ to S, where SK is the current session key and $E_{\mathrm{SK}}(m)$ means the encryption of the message m with the symmetric key SK.

Step 2. Once receiving $E_{\mathrm{SK}}(I \| v_i \| \mathrm{ID}_i)$, S computes $D_{\mathrm{SK}}(I \| v_i \| \mathrm{ID}_i)$, $v_i^* = I \oplus Y$, and then computes $Y^* = h(v_i^* \| s_1) s_2^{-1} P$, $V_i^* = v_i^* \oplus h(s_2)$, and $V_j^* = h(\mathrm{ID}_i \| V_i^* \| s_1)$. Finally, S replaces (V_i, V_j) with (V_i^*, V_j^*) and sends $E_{\mathrm{SK}}(Y \oplus Y^* \oplus v_i^*)$ to U_i, where $D_{\mathrm{SK}}(c)$ means the decryption of message c with SK.

Step 3. After receiving $E_{\mathrm{SK}}(Y \oplus Y^* \oplus v_i^*)$, U_i computes $D_{\mathrm{SK}}(Y \oplus Y^* \oplus v_i^*)$ and $Y^* = Y \oplus Y^* \oplus v_i^* \oplus Y \oplus v_i^*$. Finally, U_i replaces Y with Y^* in the memory device.

8. Security Analysis for our Proposed Scheme

In this part, we use Burrows-Abadi-Needham logic to prove the correctness of our proposed scheme at first. Then, we use informal security analysis to prove that our scheme is secure under various attacks.

8.1. Correctness Proof. In this section, we will briefly introduce the BAN logic and then prove the security of our proposed scheme by using BAN logic.

8.1.1. Brief Introduction about BAN Logic. BAN logic is a belief-based logic proposed by Burrow, Abadi, and Needham, and this logic plays a significant role in analyzing authentication protocols. When applying BAN logic to protocol analysis, it is essential to idealize the message of the protocol into a formula that BAN logic can recognize. Then, according to the reasonable initialization hypothesis, and the logical reasoning rules are used to infer whether the protocol can reach the expected goal according to the idealized protocol and initialization protocol. Figure 4 lists some of the logical symbols and inference rules for BAN logic.

8.1.2. Verifying the Proposed Scheme with BAN Logic

(1) Goals

$(g_1)\ U \mid \equiv S \mid \equiv U \leftrightarrow^{\mathrm{SK}} S$
$(g_2)\ U \mid \equiv U \leftrightarrow^{\mathrm{SK}} S$
$(g_3)\ S \mid \equiv U \mid \equiv U \leftrightarrow^{\mathrm{SK}} S$
$(g_4)\ S \mid \equiv U \leftrightarrow^{\mathrm{SK}} S$

(2) Idealized scheme

(a) $U_i \rightarrow S{:}\{r_u\}_{U \leftrightarrow^{Y} S}$, $<\mathrm{ID}_i>_{U \leftrightarrow^{v_i} S}$, $(r_u P, t)_{U \leftrightarrow^{v_i, Y} S}$, $(U \leftrightarrow^{\mathrm{SK}} S, r_s P)_{U \leftrightarrow^{v_i} S}$

(b) $S \rightarrow U_i$: $(U \leftrightarrow^{\mathrm{SK}} S, r_u P, \mathrm{realm})_{U \leftrightarrow^{v_i} S}$

(3) Initiative premises

(n1) $U \mid \equiv \#\gamma$
(n2) $U \mid \equiv \#r_u$
(n3) $S \mid \equiv \#r_s$
(n4) $U \mid \equiv U \leftrightarrow^{v_i} S$
(n5) $S \mid \equiv U \leftrightarrow^{v_i} S$
(n6) $U \mid \equiv U \leftrightarrow^{Y} S$
(n7) $S \mid \equiv U \leftrightarrow^{Y} S$
(n8) $U \mid \equiv S \mid \Rightarrow (U \leftrightarrow^{SK} S)$
(n9) $S \mid \equiv U \mid \Rightarrow (U \leftrightarrow^{SK} S)$

(4) Proof of the proposed scheme

(p1) From n4, $U_i \triangleleft (U \leftrightarrow^{\mathrm{SK}} S, r_u P, \mathrm{realm})_{U \leftrightarrow^{v_i} S}$ and by applying the message meaning rule, we deduce,

$$\frac{U \mid \equiv U \overset{v_i}{\leftrightarrow} S, U_i \triangleleft \left(U \overset{\mathrm{SK}}{\leftrightarrow} S, r_u P, \mathrm{realm}\right)_{U \overset{v_i}{\leftrightarrow} S}}{U \mid \equiv S \mid \sim \left(U \overset{\mathrm{SK}}{\leftrightarrow} S, r_u P, \mathrm{realm}\right)} \quad (1)$$

(p2) From $n2$ and by applying the fresh conjuncatenation rule, we deduce,

$$\frac{U \mid \equiv \# r_u}{U \mid \equiv \#\left(U \overset{\mathrm{SK}}{\leftrightarrow} S, r_u P, \mathrm{realm}\right)} \quad (2)$$

(p3) From $p1$, $p2$ and by applying the nonce-verification rule, we deduce,

$$\frac{U \mid \equiv \#\left(U \overset{\mathrm{SK}}{\leftrightarrow} S, r_u P, \mathrm{realm}\right), U \mid \equiv S \mid \sim \left(U \overset{\mathrm{SK}}{\leftrightarrow} S, r_u P, \mathrm{realm}\right)}{U \mid \equiv S \mid \equiv \left(U \overset{\mathrm{SK}}{\leftrightarrow} S, r_u P, \mathrm{realm}\right)} \quad (3)$$

(p4) From deduction $p3$ and by applying the belief rule, we deduce,

$$\frac{U \mid \equiv S \mid \equiv \left(U \overset{\mathrm{SK}}{\leftrightarrow} S, r_u P, \mathrm{realm}\right)}{U \mid \equiv S \mid \equiv U \overset{\mathrm{SK}}{\leftrightarrow} S}\ (\mathrm{goal\ g1}) \quad (4)$$

(p5) From $p4$, $n8$ and by applying the jurusdiction rlue, we deduce,

$$\frac{U \mid \equiv S \mid \Rightarrow \left(U \overset{\mathrm{SK}}{\leftrightarrow} S\right), U \mid \equiv S \mid \equiv U \overset{\mathrm{SK}}{\leftrightarrow} S}{U \mid \equiv U \overset{\mathrm{SK}}{\leftrightarrow} S}\ (\mathrm{goal\ g2}) \quad (5)$$

(p6) From $n5$, $S \triangleleft (U \leftrightarrow^{\mathrm{SK}} S, r_s P)_{U \leftrightarrow^{v_i} S}$ and by applying the message meaning rule, we deduce,

$$\frac{S \mid \equiv U \overset{v_i}{\leftrightarrow} S, S \triangleleft \left(U \overset{\mathrm{SK}}{\leftrightarrow} S, r_s P\right)_{U \overset{v_i}{\leftrightarrow} S}}{S \mid \equiv U \mid \sim \left(U \overset{\mathrm{SK}}{\leftrightarrow} S, r_s P\right)} \quad (6)$$

(p7) From $n3$ and by applying the fresh conjuncatenation rule, we deduce,

$$\frac{S \mid \equiv \# r_s}{S \mid \equiv \#\left(U \overset{\mathrm{SK}}{\leftrightarrow} S, r_s P\right)} \quad (7)$$

(p8) From $p6$, $p7$ and by applying the nonce-verification rule, we deduce,

$$\frac{S \mid \equiv \#\left(U \overset{\mathrm{SK}}{\leftrightarrow} S, r_s P\right), S \mid \equiv U \mid \sim \left(U \overset{\mathrm{SK}}{\leftrightarrow} S, r_s P\right)}{S \mid \equiv U \mid \equiv \left(U \overset{\mathrm{SK}}{\leftrightarrow} S, r_s P\right)} \quad (8)$$

(p9) From deduction $p8$ and by applying the belief rule, we deduce,

$$\frac{S \mid \equiv U \mid \equiv \left(U \overset{\mathrm{SK}}{\leftrightarrow} S, r_s P\right)}{S \mid \equiv U \mid \equiv U \overset{\mathrm{SK}}{\leftrightarrow} S}\ (\mathrm{goal\ g3}) \quad (9)$$

$X \mid\equiv A$	X believes a statement A
$X \overset{K}{\longleftrightarrow} Y$	X and Y share a key K
$X \triangleleft A$	X sees A
$\#A$	A is fresh
$X \mid\sim A$	X said A
$(A, B)_K$	A and B are hashed by the key k
$\{A\}_K$	Encryption of A with key K
$< X >_Y$	Combination of X and Y
$X \mid\Rightarrow A$	X has jurisdiction over A
$\mid\overset{K}{\rightarrow} P$	k is P's public key
$X \overset{Q}{=} Y$	Q is the shared secret between X and Y
$\dfrac{X\mid\equiv X \overset{K}{\longleftrightarrow} Y,\ X \triangleleft\{A\}_K}{X \mid\equiv Y\mid\sim A}$	Message-meaning rule
$\dfrac{X\mid\equiv\#A,\ X\mid\equiv Y\mid\sim A}{X \mid\equiv Y\mid\equiv A}$	Nonce-verification rule
$\dfrac{X\mid\equiv(A, B)}{X \mid\equiv A}$	Belief rule
$\dfrac{X \mid\equiv\#A}{X\mid\equiv\#(A,B)}$	Fresh conjuncatenation rule
$\dfrac{X \mid\equiv Y\mid\Rightarrow A,\ X\mid\equiv Y\mid\sim A}{X\mid\equiv A}$	Jurisdiction rule

Figure 4: BAN logic notations.

(p10) From $p9$, $n9$ and by applying the jurusdiction rlue, we deduce,

$$\frac{S\mid\equiv U\mid\Rightarrow\left(U \overset{\mathrm{SK}}{\leftrightarrow} S\right),\ S\mid\equiv U\mid\equiv U \overset{\mathrm{SK}}{\leftrightarrow} S}{S\mid\equiv U \overset{\mathrm{SK}}{\leftrightarrow} S}(\text{goal g4}) \tag{10}$$

Therefore, our proposed scheme achieves mutual authentication and key agreement between S and U_i.

8.2. Informal Security Analysis. The security of our proposed scheme will be discussed in this section. We will prove our scheme is secure in the face of various attacks. We draw on the experience of [14] and define the capabilities of the attacker $\mathscr{A}$ as follows:

(c1) $\mathscr{A}$ can off-line enumerate the Cartesian product $S_{\mathrm{id}} * S_{\mathrm{PW}}$, where S_{id}, S_{PW} means the size of the identity space and password space, separately

(c2) $\mathscr{A}$ has full control of the communication channel

(c3) $\mathscr{A}$ may either learn the victim's password via malicious card readers, or extract the secret data in the card by side-channel attacks, but cannot realize both

(c4) $\mathscr{A}$ can learn the previous session key(s)

(c5) $\mathscr{A}$ can learn the server's long-time private key(s) as well as all other data stored in the server only when evaluating the eventual failure of the server

8.2.1. Denning-Sacco Attack. In Denning-Sacco attack [15], when the client or server leaks the previous session key, then $\mathscr{A}$ tries to get other session keys or a long-term key (for example, the client's password or the server's key).

Suppose $\mathscr{A}$ has gained a session key $\mathrm{SK} = h(\mathrm{ID}_i\|v_i\|Q_{sc}\|$ realm). $\mathscr{A}$ cannot obtain use's password PW_i since PW_i is hidden in a hash function $v_i = h(h(\mathrm{ID}_i) \oplus h(\mathrm{PW}_i) \oplus h(\gamma))$. Besides, it is impossible for $\mathscr{A}$ to obtain the other session keys as he/she does not know the values of $r_u r_s P$ and v_i.

8.2.2. Man-in-the-Middle Attack (MITM). In this attack, $\mathscr{A}$ intercepts communication channels between users and the server and attempts to make them believe that they are communicating with each other directly.

In our proposed scheme, assume $\mathscr{A}$ intercepts $\mathrm{REQUEST}(\mathrm{PID}, X, H_j, t)$ and $\mathrm{RESPONSE}(\mu, \mathrm{realm}, \eta)$. However, $\mathscr{A}$ does not know Y and v_i so that he/she is not able to figure out the right PID,X and H_j. $\mathscr{A}$ will fail to cheat server as a legal user. At the same time, $\mathscr{A}$ is unaware of the server's secret keys s_2 and s_1. So $\mathscr{A}$ cannot masquerade as a legal server since he/she cannot compute the right $Q = r_u P$ and $Q_{sc} = r_u r_s P$.

8.2.3. Off-Line Password Guessing Attack. Off-line password guessing attack means $\mathscr{A}$ keeps previous authentication messages. Then, $\mathscr{A}$ selects a set of candidate passwords and uses stored messages to verify whether there is a appropriate password.

In our proposed scheme, $\mathscr{A}$ can obtain PID, X, H_j, μ, realm, and η from the communicating channels between users and server. But $\mathscr{A}$ cannot compute the values of $h(\mathrm{SK}\|r_u P\|\mathrm{realm}\|h(h(\mathrm{ID}_i) \oplus h(\mathrm{PW}_i) \oplus h(\gamma)))$ and $h(h(h(\mathrm{ID}_i) \oplus h(\mathrm{PW}_i) \oplus h(\gamma))\|Y\|r_u P\|t)$ to verify the candidate password since $\mathscr{A}$ does not know the values of γ, $r_u P$, $r_u r_s P$, s_1, and s_2.

8.2.4. Replay Attack. In this attack, suppose $\mathscr{A}$ grabs $\mathrm{REQUEST}(\mathrm{PID}, X, H_j, t)$ when a legal user U_i try to send REQUEST to server, then $\mathscr{A}$ replays it to server to impersonate U_i. However, on account of the attacker is unaware of v_i and Y, the server can easily find out if the attacker modified the time stamp t. If $\mathscr{A}$ replay REQUEST to server without any changing, the server can figure out the message is invalid since the verification of t will not hold.

8.2.5. Impersonation Attack. In this attack, the goal of $\mathscr{A}$ is impersonating a legal server or a legal user. Suppose a legal user attempts to use what he/she has got to masquerade other

Table 5: Comparison of computational cost.

Schemes	Total comparison of computational cost	Time (ms)
Our scheme	$9T_{\text{EPM}} + 21T_H + 3T_{\text{INV}} + 4T_R$	22.2551 ms
Zhang et al.'s scheme [10]	$10T_{\text{EPM}} + 19T_H + 2T_{\text{INV}} + 2T_{\text{EAM}} + 5T_R$	25.0675 ms
Irshad et al.'s scheme [11]	$9T_{\text{EPM}} + 19T_H + 2T_{\text{INV}} + 5T_R$	22.7839 ms
Arshad et al.'s scheme [12]	$4T_{\text{EPM}} + 20T_H + 1T_{\text{INV}} + 4T_R$	11.1116 ms
Lu et al.'s scheme [13]	$6T_{\text{EPM}} + 14T_H + 5T_R$	16.0832 ms

users, and all IDs are available for him/her.That means one of the legal users is $\mathscr{A}$ now. Then, $\mathscr{A}$ can intercept REQUEST and compute $v_i = \text{ID}_i \oplus \text{PID}$ obviously. However, since s_1 is unknown for the attacker and the hardness of ECDLP, $\mathscr{A}$ cannot compute the correct $Y = h(v_i \| s_1) s_2^{-1} P$. If $\mathscr{A}$ use his/her own v_i' and the victim user's ID_i to compute $\text{PID} = \text{ID}_i \oplus v_i'$, then use his/her own Y', v_i' to compute $H_j = h(v_i' \| Y' \| r_u P \| t)$ and attempt to impersonate U_i. The verification $V_j' = h(\text{ID}_i' \| V_i \| s_1) \stackrel{?}{=} V_j$ will not hold since V_i is matched with v_i and ID_i when server computes V_j. Thus, our proposed scheme can withstand user impersonation attack.

Suppose $\mathscr{A}$ tries to impersonate a legal server. Since the server's secret key S_1 and s_2 are unknown for $\mathscr{A}$, he/she cannot figure out the correct $Q = X s_2 h^{-1}(v_i' \| s_1) = r_u P$. Thus, the verification $\eta' = h(\text{SK} \| Q' \| \text{realm} \| v_i) = \eta$ will fail on the user's side.

8.2.6. Privileged inside Attack. Suppose a privileged inside user $\mathscr{A}$ of the server obtains v_i, ID_i, and PW_i of a legal user U_i, and then $\mathscr{A}$ tries to impersonate U_i to access server. Since Y is only stored in memory device which is kept by U_i and server's secret key s_1, s_2 are kept by server, the attacker cannot get the right Y. Thus, the verification $H_j' = h(v_i' \| Y' \| Q) \stackrel{?}{=} H_j$ will not hold if a privileged inside attacker want to masquerade the victim user.

8.2.7. Known Session Key Attack. If a previous session key SK has been divulged, and $\mathscr{A}$ attempts to compute a new session key. The attacker will not make it since each session key is independent from others. Only if the attacker has the newest r_u and r_s could him/her have opportunity to compute the newest $\text{SK} = (\text{ID}_i \| v_i \| Q_{sc} \| \text{realm})$.

8.2.8. Perfect Forward Secrecy. Perfect forward secrecy means that $\mathscr{A}$ cannot obtain old session keys even if the secret key of server or the user's password has leaked. In our scheme, $\text{SK} = h(\text{ID}_i \| v_i \| Q_{sc} \| \text{realm})$. And $\mathscr{A}$ cannot figure out previous session keys even if he gains user's password PW_i and server's secret key s_2 because r_u and r_s are unknown for $\mathscr{A}$. Due to the hardness of ECDLP, it is hardly impossible for $\mathscr{A}$ to obtain r_s from $\mu = r_s P$ and r_u from $X = r_u H_i$. Consequently, the proposed scheme achieves perfect forward secrecy.

8.2.9. Anonymous and Untraceable. Suppose $\mathscr{A}$ intercepts $\text{REQUEST}(\text{PID}, X, H_j, t)$ and $\text{RESPONSE}(\mu, \text{realm}, \eta)$. Since v_i is unknown for $\mathscr{A}$, he/she cannot compute $\text{ID}_i = \text{PID} \oplus v_i$. Thus, our proposed scheme is anonymous to third parties. Besides, since a legal user use a random number r_u to compute X and H_j for each session, so $\mathscr{A}$ cannot trace who is communicating with S.

8.2.10. Stolen Verifier Attack. Suppose $\mathscr{A}$ obtains V_i, V_j from server's database and tries to impersonate a legal user. Since the server's secret s_2 is unavailable for $\mathscr{A}$, $\mathscr{A}$ cannot figure out v_i of a legal user as $v_i = V_i \oplus h(s_2)$.

8.2.11. Stolen Memory Device Attack. Suppose $\mathscr{A}$ had stolen the smart card of user and can extract the secret parameters from the smart card, then $\mathscr{A}$ tries to attack the user by using impersonation attack or off-line password guessing attack. Since the password is unknown for $\mathscr{A}$, $\mathscr{A}$ will fail when he wants to impersonate the legal user. When $\mathscr{A}$ obtains previous PID from the communication channel and decides to use off-line password guessing attack to find user's $(\text{ID}_i, \text{PW}_i)$ pair. However, v_i is computed as $v_i = h(h(\text{ID}_i) \oplus h(\text{PW}_i) \oplus h(\gamma))$, and $\mathscr{A}$ will be frustrated because there are $S_{\text{id}} * S_{\text{PW}} \approx 2^{40}$ candidates of $(\text{ID}_i, \text{PW}_i)$, where $S_{\text{id}} = S_{\text{PW}} = 10^6$ [16, 17] and S_{id}, S_{PW} mean the size of the identity space and password space, separately.

9. Performance Analysis

The performance comparison of our scheme and other related schemes [10–13] will be presented in this section.

In Table 5, we compute the total computational costs of three phases (registration phase, authentication and key agreement phase and password change phase) of our scheme and make a comparison with other schemes. In order to represent each computation cost of time, we define some notations in Figure 5.

According to [18, 19], an elliptic curve point multiplication operation takes 2.226 ms, an elliptic curve point addition operation costs 0.0288 ms, a one-way hash function takes 0.0023 ms, a modular inversion operation takes 0.0056 ms, and generating a random number needs 0.539 ms.

Figure 6 shows the comparison of security attributes between our scheme and other schemes. We can notice that our scheme is resistant to various attacks. On the contrary, Zhang et al. [10] and Irshad et al.'s [11] schemes cannot withstand impersonation attack and are not anonymous and untraceable, and Arshad et al.'s [12] scheme fails to achieve anonymity and untraceability and cannot stand up to impersonation attack. Meanwhile, Lu et al.'s [13] scheme cannot resist impersonation attack. So our scheme's computational cost is lower than Irshad et al. and Zhang et al.'s schemes

Notations	Definitions
T_{EPM}	The time for performing an elliptic curve point multiplication operation.
T_{EAM}	The time for performing an elliptic curve point addition operation.
T_H	The time for performing a one-way hash function.
T_{INV}	The time for performing a modular inversion operation.
T_R	The time for generating a random number.

Figure 5: Notations for computation cost of times.

Attacks	Our scheme	Zhang et al. 's	Irshad et al. 's	Arshad et al. 's	Lu et al. 's
Replay attack	√	√	√	√	√
Impersonation attack	√	×	×	×	×
Perfect forward secrecy	√	√	√	√	√
Denning-Sacco attack	√	√	√	√	√
Anonymous and untraceable	√	×	×	×	√
Off-line password guessing attack	√	√	√	√	√
Mutual Authentication	√	√	√	×	√

Figure 6: Comparison of security attributes.

while providing better security. Despite having a little more computational cost, our scheme performance better in security attributes than Arshad et al. and Lu et al.'s schemes. From what we have discussed above, we can draw a conclusion that our proposed scheme is efficient and can withstand virous known attacks.

10. Conclusion

In this paper, we have demonstrated that Arshad et al.'s scheme cannot withstand user impersonation attack and Lu et al.'s scheme is not secure against server impersonation attack. In order to remedy the weaknesses of their schemes, we present an enhanced anonymous and efficient ECC-based authentication scheme for SIP. Our scheme inherits the merits of Arshad and Nikooghadam and Lu et al.'s schemes while standing up to user and server impersonation attacks that their schemes failed to satisfy. We use BAN logic and informal analysis to demonstrate the correctness and security of our scheme. Therefore, our proposed scheme is suitable and practical for SIP.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Our work was jointly supported by the National Natural Science Foundation of China (No. 61872051, No. 61702067), the Chongqing Natural Science Foundation of China (No. cstc2020jcyj-msxmX0343), and the Venture & Innovation Support Program for Chongqing Overseas Returnees (No. CX2018122).

References

[1] H. Schulzrinne, G. Camarillo, A. Johnston et al., *Sip: session initiation protocol*, IETF RFC, 2002.

[2] J. L. Tsai, "Efficient nonce-based authentication scheme for session initiation protocol," *IJ Network Security*, vol. 9, no. 1, pp. 12–16, 2009.

[3] E.-J. Yoon, Y.-N. Shin, I.-S. Jeon, and K.-Y. Yoo, "Robust mutual authentication with a key agreement scheme for the session initiation protocol," *IETE Technical Review*, vol. 27, no. 3, pp. 203–213, 2010.

[4] Q. Xie, "A new authenticated key agreement for session initiation protocol," *International Journal of Communication Systems*, vol. 25, no. 1, pp. 47–54, 2012.

[5] M. S. Farash and M. A. Attari, "An enhanced authenticated key agreement for session initiation protocol," *Information Technology and Control*, vol. 42, no. 4, pp. 333–342, 2013.

[6] Z. Zhang, Q. Qi, N. Kumar, N. Chilamkurti, and H.-Y. Jeong, "A secure authentication scheme with anonymity for session initiation protocol using elliptic curve cryptography," *Multimedia Tools and Applications*, vol. 74, no. 10, pp. 3477–3488, 2015.

[7] Y. Lu, L. Li, H. Peng, and Y. Yang, "A secure and efficient mutual authentication scheme for session initiation protocol," *Peer-to-Peer Networking and Applications*, vol. 9, no. 2, pp. 449–459, 2016.

[8] S. A. Chaudhry, I. Khan, A. Irshad, M. U. Ashraf, M. K. Khan, and H. F. Ahmad, "A provably secure anonymous authentication scheme for session initiation protocol," *Security and Communication Networks*, vol. 9, no. 18, pp. 5016–5027, 2016.

[9] S. Kumari, M. Karuppiah, A. K. Das, X. Li, F. Wu, and V. Gupta, "Design of a secure anonymity-preserving authentication scheme for session initiation protocol using elliptic curve cryptography," *Journal of Ambient Intelligence and Humanized Computing*, vol. 9, no. 3, pp. 643–653, 2018.

[10] L. Zhang, S. Tang, and Z. Cai, "Efficient and exible password authenticated key agreement for voice over internet protocol session initiation protocol using smart card," *International Journal of Communication Systems*, vol. 27, no. 11, pp. 2691–2702, 2013.

[11] A. Irshad, M. Sher, Eid Rehman, S. A. Ch, M. U. Hassan, and A. Ghani, "A single round-trip sip authentication scheme for voice over internet protocol using smart card," *Multimedia Tools and Applications*, vol. 74, no. 11, pp. 3967–3984, 2015.

[12] H. Arshad and M. Nikooghadam, "An efficient and secure authentication and key agreement scheme for session initiation protocol using ecc," *Multimedia Tools and Applications*, vol. 75, no. 1, pp. 181–197, 2016.

[13] L. L. Y. Lu, L. Li, and Y. Yang, "Robust and efficient authentication scheme for session initiation protocol," *Mathematical Problems in Engineering*, vol. 2015, Article ID 894549, 9 pages, 2015.

[14] 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.

[15] D. E. Denning and G. M. Sacco, "Timestamps in key distribution protocols," *Communications of the ACM*, vol. 24, no. 8, pp. 533–536, 1981.

[16] J. Ma, W. Yang, M. Luo, and N. Li, "A study of probabilistic password models," in *2014 IEEE Symposium on Security and Privacy*, pp. 689–704, San Jose, CA, USA, May 2014.

[17] J. Bonneau, "The science of guessing: analyzing an anonymized corpus of 70 million passwords," in *2012 IEEE Symposium on Security and Privacy*, pp. 538–552, San Francisco, CA, USA, May 2012.

[18] H. H. Kilinc and T. Yanik, "A survey of sip authentication and key agreement schemes," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 2, pp. 1005–1023, 2014.

[19] N. Koblitz, A. Menezes, and S. Vanstone, "The state of elliptic curve cryptography," *Designs, Codes and Cryptography*, vol. 19, no. 2/3, pp. 173–193, 2000.