Research Article

An Anonymous and Efficient ECC-Based Authentication Scheme for SIP

Yousheng Zhou and Xinyun Chen

1College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
2School 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

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 participants. 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 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 Nkooghadam’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 Nikooghadam’s Scheme

In this section, we will review Arshad and Nikooghadam’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 $h(\cdot)$. Then, the server selects a base point $P$ with order $n$ over $E_p(a, b)$, chooses a integer $k$, randomly and keeps it as a secret key, and computes public key $K_s = kP$. Finally, the server publishes $(E_p(a, b), n, P, h(\cdot), K_s)$.

3.2. Registration

(1) The client generates a number $N_c$ randomly, chooses a password $PW_i$, computes $v_i = h(ID_i||PW_i||N_c)$, 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 $V_i = h(ID_i||k_s||v_i)$ 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 Nikooghadam’s scheme and the steps of this phase are also represented by Table 1.

(1) The client selects an integer $d_i$ randomly, computes $R_i = d_iK_s = d_iP$, and sends REQUEST$(ID, R_i)$ to the server through the public channel

(2) If $ID_i$ exists in database, the server selects a integer $d_i$ randomly, computes $Q_s = d_iP$, $Q_{cs} = d_iK_s$.

(3) The client computes $Q_{cs} = d_iQ_s$ and compares the value of $h(ID_i||Q_s||Q_{cs})$ and $V_s$. If $h(ID_i||Q_s||Q_{cs})$ and $V_s$ are not equal, the session will be stopped by the client. Otherwise, the client computes $V_s = h(ID_i||Q_s||Q_{cs})$ and $SK = h(ID_i||Q_s||Q_{cs})$.

(4) The server computes $v_i = V_i$ and compares the value of $h(ID_i||Q_s||Q_{cs})$ and $V_i$. If $h(ID_i||Q_s||Q_{cs})$ and $V_i$ is equal to $V_i$, 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 $v^*_i = h(ID_i||PW^*_i||N^*_c)$, $z = h(ID_i||PW^*_i||N^*_c)$, $V^*_i = v_i$ and sends CHANGEPWD$(ID, Z, V^*_i)$ to server

(2) The server computes $v_i = V_i \oplus h(ID_i||K_s)$, $v^*_i = Z \oplus v_i \oplus h(ID_i||SK)$, and verify whether $h(ID_i||v^*_i||SK)$ is equal to $V_i$ 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 ACCEPT$(h(ID_i||v^*_i||accept||v^*_i||SK))$ to the client
Suppose Step 1. When a client wants to communicate with the server, then, \( \mathcal{A} \) forges \( Q' = R_0 = d_c K_s \), where \( K_s \) is the server’s public key. Then, \( \mathcal{A} \) computes \( V'_s = h(ID)\|Q'_s\|Q'_{cs} \) and sends CHALLENGE\( (\text{realm}, Q'_s, V'_s) \) to the client.

Step 2. After receiving CHALLENGE, the client computes \( Q_{cs} = d_c Q'_s \). Since \( Q' = K_s \), \( Q_{cs} = d_c K_s = Q_{cs} \). Thus, \( h(ID)\|Q'\|Q_{cs} \) and the verification will hold. The client authenticates the “server.” Then, the client computes \( v_i = h(ID)\|PW_i\|N_c \), \( V_c = h(ID)\|Q'\|Q_{cs} \), \( \text{SK} = h(ID)\|Q_{cs} \|\text{realm} \). Finally, the client sends RESPONSE\( (ID, Q'_s, Q_{cs} \|\text{realm}) \) to \( \mathcal{A} \).

Step 3. After receiving RESPONSE\( (ID, Q'_s, Q_{cs} \|\text{realm}) \), \( \mathcal{A} \) does not need to compute \( v_i \) and verify whether \( V_c = h(ID)\|Q'\|Q_{cs} \|\text{realm} \) or not. \( \mathcal{A} \) only need to computes \( \text{SK} = h(ID)\|Q'\|Q_{cs} \|\text{realm} \) 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                                                                 |
|-----------------|----------------------------------------------------------------------------|
| \( E \_b (a, b) \) | An elliptic curve with prime order \( n \)                                 |
| \( P \)          | The base point of the elliptic curve                                        |
| \( xP \)         | The elliptic curve point multiplication defined as \( xP = P + P + \ldots + P + P \) \( x \text{times} \) |
| \( h (\cdot) \)   | A one-way hash function \( h \)                                             |
| \( K_s \)        | The secret key of server                                                   |
| \( K_r \)        | The public key of server                                                   |
| \( PW_i \)       | The password of \( i_{th} \) client                                       |
| \( ID_i \)       | The identity of \( i_{th} \) client                                       |
| \( \| \)          | The concatenation operation                                                 |
| \( \oplus \)     | The bit-wise exclusive-or(XOR) operation                                    |

**Figure 1: Notations of Arshad et al.’s scheme.**

**Table 1: Authentication and key agreement.**

| Notations       | Definitions                                                                 |
|-----------------|----------------------------------------------------------------------------|
| \( N_c \)       | The secret key of client                                                   |
| \( Q' \)        | The ephemeral public key                                                   |
| \( Q_{cs} \)    | The shared secret key                                                      |
| \( \text{ID} \) | The identity of the client                                                 |

(3) The client calculates \( h(ID)\|v_i\|\text{accept}\|v'_s\|\text{SK} \) and if it is equal to the ACCEPT, the client replaces \( N_c \) with \( N'_c \) in the memory device.
Notations | Definitions
--- | ---
$U_i$ | User
$S$ | Server
$h()$ | A one-way hash function $h$
$k_s$ | The secret key of server
$PW_i$ | The password of user $U_i$
$ID_i$ | The identity user $U_i$
$\parallel$ | The concatenation operation
$\oplus$ | The bit-wise exclusive-or (XOR) operation

5.2. Authentication and Key Agreement.

Table 2: Authentication and Key Agreement.

| Notation | Description |
|----------|-------------|
| $M_1$ | The message sent by $U_i$ |
| $M_2$ | The message sent by $S$ |
| $M_3$ | The message sent by $U_i$ |
| $M_4$ | The message sent by $S$ |

Lu et al.’s scheme and the steps of this phase are also represented by Table 2.

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

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

(2) $S$ calculates $VPW = h(ID_i||PWD||r_1) \oplus h(k_s)$ and stores $VPW$ in database.

5.3. Password Change

(1) $U_i$ sends the message $h(h(ID_i||r_1||h(PW_i||r_1)||SK))$ and $h(ID_i||r_1'||h(PW_i||r_1'||SK))$ to $S$.

(2) If $h(h(ID_i||r_1||h(PW_i||r_1)||SK))$ is equal to the value of $h(h(PW_i||r_1)\oplus h(k_s)||SK)$ that $S$ just calculated, $S$ sends $VPW_{new} = SK \oplus h(ID_i||r_1'||h(PW_i||r_1'||SK))$ to $U_i$.

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.
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 $K_p = s_2P$. 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 $y$ randomly, chooses a password $PW_i$, computes $v_i = h(h(ID_i) \oplus h(PW_i) \oplus h(y))$. 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 $(Y, y)$ 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
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 $r_\alpha \in \mathbb{Z}_p^*$ randomly, generates a time stamp $t$, and then computes $v_i = h((ID_i) \oplus h(P) \oplus h(y))$, and $X = r_\alpha Y = r_\alpha s_2^{-1} h(v_i || s_1) P$. Finally, $U_i$ computes $PID = ID_i \oplus v_i$ and sends REQUEST $(PID, X, H_j, t)$ to $S$.

Step 2. $S$ checks the validity of time stamp $t$ by checking the validity of the predicate $(t - T < \Delta t)$, and abores if the check fails. Then, $S$ computes $D_1 = V_j \oplus h(s_2) \oplus PID, V'_j = h([ID] \oplus V_j \oplus s_1)$ and compares the values of $V'_j$ and the stored $V_j$. If they are equal, $S$ can make sure that the received $ID_j$ and $V_j$ is a pair. After that, $S$ computes $v'_i = V_i \oplus h(s_2), Y' = h(v'_i \oplus s_1) P$, and $X = r_\alpha X = r_\alpha s_2^{-1} h(v'_i || s_1)$ $P$ and checks whether $H'_j = h(V'_j || s_1) H_j$ equals to $H_j$ or not. If they are equal, $S$ authenticates the user $U_i$. Then, $S$ chooses a integer $r_\xi \in \mathbb{Z}_p^*$ randomly, computes $Q_{sc} = r_\xi Q = r_\xi r_\alpha P$, $SK = h([ID] \oplus v'_i || Q_{sc} || realm)$, $\mu = r_\xi P$, and $\eta = h(SK || Q_{sc} || realm || v'_i)$. Finally, $S$ sends RESPONSE $(\mu, realm, \eta)$ to $U_i$.

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

Step 4. $S$ calculates $Auth' = h(SK || r_\xi P || v'_i)$ 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 $y^*$ randomly, selects a new password $PW^*$, then $U_i$ computes $v^*_i = h((ID_i) \oplus h(PW^*_i) \oplus h(y^*))$, $I = Y \oplus v^*_i$ and sends $E_{SK}(I || v'_i || ID_i)$ to $S$, where $SK$ is the current session key and $E_{SK}(m)$ means the encryption of the message $m$ with the symmetric key $SK$.

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

Step 3. After receiving $E_{SK}(Y \oplus Y^* \oplus v^*_i)$, $U_i$ computes $D_{SK}(Y \oplus Y^* \oplus v^*_i)$ 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

\[ (g_1) \quad U \vdash S \vdash U \leftrightarrow ^{\textbf{SK}} S \]

\[ (g_2) \quad U \vdash S \vdash U \leftrightarrow ^{\textbf{SK}} S \]

\[ (g_3) \quad S \vdash U \vdash U \leftrightarrow ^{\textbf{SK}} S \]

\[ (g_4) \quad S \vdash U \vdash ^{\textbf{SK}} S \]

(2) Idealized scheme

\[ U_i \rightarrow S; (r_u)_{U \leftarrow ^{\textbf{SK}} r_u} \rightarrow (r_u, t)_{U \leftarrow ^{\textbf{SK}} r_u} \rightarrow U \rightarrow S \]

(b) \quad \rightarrow U \rightarrow S; (U \leftarrow ^{\textbf{SK}} S, r_u, P, \text{realm})_{U \leftarrow ^{\textbf{SK}} S} \]

(3) Initiative premises

\[ (n1) \quad U \vdash \#\gamma \]

\[ (n2) \quad U \vdash \#^a_r \]

\[ (n3) \quad S \vdash \#^a_r \]

\[ (n4) \quad U \vdash U \leftrightarrow ^{\textbf{SK}} S \]

\[ (n5) \quad S \vdash U \leftrightarrow ^{\textbf{SK}} S \]

\[ (n6) \quad U \vdash U \rightarrow ^{\textbf{SK}} S \]

\[ (n7) \quad S \vdash U \rightarrow ^{\textbf{SK}} S \]

\[ (n8) \quad U \vdash \#^a_r \rightarrow (U \leftrightarrow ^{\textbf{SK}} S) \]

\[ (n9) \quad S \vdash U \rightarrow (U \leftrightarrow ^{\textbf{SK}} S) \]

(4) Proof of the proposed scheme

(p1) From n4, U; s(U \leftarrow ^{\textbf{SK}} S, r_u, P, \text{realm})_{U \leftarrow ^{\textbf{SK}} S} \]

\[ U \vdash U \rightarrow U \rightarrow S; U \rightarrow S \rightarrow (U \leftarrow ^{\textbf{SK}} S, r_u, P, \text{realm})_{U \leftarrow ^{\textbf{SK}} S} \]

\[ U \vdash S \rightarrow (U \leftarrow ^{\textbf{SK}} S, r_u, P, \text{realm})_{U \leftarrow ^{\textbf{SK}} S} \]

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

\[ U \equiv \#^a_r \rightarrow U \leftarrow ^{\textbf{SK}} S, r_u, P, \text{realm} \]

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

\[ U \equiv S \rightarrow U \leftarrow ^{\textbf{SK}} S, r_u, P, \text{realm} \]

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

\[ U \equiv S \equiv U \leftarrow ^{\textbf{SK}} S \]

(p5) From p4, n8 and by applying the jurisdiction rule, we deduce,

\[ U \equiv S \equiv U \leftarrow ^{\textbf{SK}} S \]

(p6) From n5, S \a(U \leftarrow ^{\textbf{SK}} S, r_u, P)_{U \leftarrow ^{\textbf{SK}} S} \]

\[ S \equiv U \rightarrow S \rightarrow S \rightarrow (U \leftarrow ^{\textbf{SK}} S, r_u, P)_{U \leftarrow ^{\textbf{SK}} S} \]

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

\[ S \equiv \#^a_r \rightarrow U \leftarrow ^{\textbf{SK}} S, r_u, P, \text{realm} \]

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

\[ S \equiv U \leftarrow ^{\textbf{SK}} S, r_u, P, \text{realm} \rightarrow (U \leftarrow ^{\textbf{SK}} S, r_u, P)_{U \leftarrow ^{\textbf{SK}} S} \]

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

\[ S \equiv U \equiv U \leftarrow ^{\textbf{SK}} S \]

\[ S \equiv U \equiv U \leftarrow ^{\textbf{SK}} S \]
From p9, n9 and by applying the jurisdiction rule, we deduce,

\[
S \equiv U \Rightarrow \left( U \xrightarrow{SK} S \right), S \equiv U \xrightarrow{SK} S \xrightarrow{SK} S
\]

(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 \( \mathcal{A} \) as follows:

- (c1) \( \mathcal{A} \) can off-line enumerate the Cartesian product \( S_A \times S_{PW} \), where \( S_A \) and \( S_{PW} \) means the size of the identity space and password space, separately.
- (c2) \( \mathcal{A} \) has full control of the communication channel.
- (c3) \( \mathcal{A} \) 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) \( \mathcal{A} \) can learn the previous session key(s).
- (c5) \( \mathcal{A} \) 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 \( \mathcal{A} \) tries to get other session keys or a long-term key (for example, the client's password or the server's key).

Suppose \( \mathcal{A} \) has gained a session key \( SK = h(ID) \parallel v_i \parallel Q_{sk} \) realm. \( \mathcal{A} \) cannot obtain use's password \( PW_i \) since \( PW_i \) is hidden in a hash function \( v_i = h(h(ID) \oplus h(PW_i) \oplus h(y_i)) \). Besides, it is impossible for \( \mathcal{A} \) to obtain the other session keys as he/she does not know the values of \( r_u, r_P \) and \( v_i \).

8.2.2. Man-in-the-Middle Attack (MITM). In this attack, \( \mathcal{A} \) 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 \( \mathcal{A} \) intercepts REQUEST(PID, X, H_j, t) and RESPONSE(\( \mu \), realm, \( \eta \)). However, \( \mathcal{A} \) does not know Y and \( \gamma \), so he/she is not able to figure out the right PID, X and H_j. \( \mathcal{A} \) will fail to cheat server as a legal user. At the same time, \( \mathcal{A} \) is unaware of the server's secret keys \( s_2 \) and \( s_1 \). So \( \mathcal{A} \) cannot masquerade as a legal server since he/she cannot compute the right \( Q = r_1P \) and \( Q_{sk} = r_1s_1 \).

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

In our proposed scheme, \( \mathcal{A} \) can obtain PID, X, H_j, \( \mu \), realm, and \( \eta \) from the communicating channels between users and server. But \( \mathcal{A} \) cannot compute the values of \( h(SK || r_uP || \text{realm}) \oplus h(h(ID_j) \oplus h(PW_i) \oplus h(y_j)) \) and \( h(h(ID_j) \oplus h(PW_i) \oplus h(y_j)) || Y || r_1P || t \) to verify the candidate password since \( \mathcal{A} \) does not know the values of \( y, r_uP, r_uP, s_1, \) and \( s_2 \).

8.2.4. Replay Attack. In this attack, suppose \( \mathcal{A} \) grabs REQUEST(PID, X, H_j, t) when a legal user U_i tries to send REQUEST to server, then \( \mathcal{A} \) 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 \( \mathcal{A} \) 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 \( \mathcal{A} \) is impersonating a legal server or a legal user. Suppose a legal user attempts to use what he/she has got to masquerade other
users, and all IDs are available for him/her. That means one of the legal users is \( \mathcal{A} \) now. Then, \( \mathcal{A} \) 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, \( \mathcal{A} \) cannot compute the correct \( Y = h(v_i)^{s_1} P \). If \( \mathcal{A} \) use his/her own \( v_i \) and the victim user’s \( \text{ID}_i \) to compute \( \text{PID} = \text{ID}_i \oplus v_i \), then use his/her own \( Y' \), \( v_i' \) to compute \( H_j = h(Y')^v P \) and attempt to impersonate \( U_i \). The verification \( V_j' = h(\text{ID}_j)^v \) \( V_j \parallel s_1 = ? V_j \) will not hold since \( V_j \) is matched with \( v_i \) and \( \text{ID}_j \) when server computes \( V_j \). Thus, our proposed scheme can withstand user impersonation attack.

Suppose \( \mathcal{A} \) tries to impersonate a legal server. Since the server’s secret key \( s_1 \) and \( s_2 \) are unknown for \( \mathcal{A} \), he/she cannot figure out the correct \( Q = Xs_2 h^{-1}(v_i^{s_1}) = r_u P \). Thus, the verification \( \eta' = h(\text{SK}Q \parallel \text{realm} \parallel v_i) = \eta \) will fail on the user’s side.

8.2.6. Privileged inside Attack. Suppose a privileged inside user \( \mathcal{A} \) of the server obtains \( v_i \), \( \text{ID}_j \), and \( \text{PW}_i \) of a legal user \( U_j \), and then \( \mathcal{A} \) tries to impersonate \( U_j \) to access server. Since \( Y \) is only stored in memory device which is kept by \( U_j \) and server’s secret key \( s_1 \), \( s_2 \) are kept by server, the attacker cannot get the right \( Y \). Thus, the verification \( H_j' = h(Y)^v \parallel Q = ? H_j \) 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 \( \mathcal{A} \) 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 \( r_2 \), he/she could have opportunity to compute the newest SK = \( (\text{ID}_j \parallel v_i \parallel Q_{u} \parallel \text{realm}) \).

8.2.8. Perfect Forward Secrecy. Perfect forward secrecy means that \( \mathcal{A} \) cannot obtain old session keys even if the secret key of server or the user’s password has leaked. In our scheme, \( \text{SK} = h(\text{ID}_j \parallel v_i \parallel Q_{u} \parallel \text{realm}) \). And \( \mathcal{A} \) cannot figure out previous session keys even if he gains user’s password \( \text{PW}_i \) and server’s secret key \( s_2 \) because \( r_u \) and \( r_2 \) are unknown for \( \mathcal{A} \). Due to the hardness of ECDLP, it is hardly impossible for \( \mathcal{A} \) to obtain \( r_i \) from \( \mu = r_u P \) and \( r_2 \) from \( X = r_2 H_j \). Consequently, the proposed scheme achieves perfect forward secrecy.

8.2.9. Anonymous and Untraceable. Suppose \( \mathcal{A} \) intercepts REQUEST(\( \text{PID}, X, H_j, t \)) and RESPONSE(\( \mu, \text{realm}, \eta \)). Since \( v_i \) is unknown for \( \mathcal{A} \), he/she cannot compute \( \text{ID}_j = \text{PID} \oplus v_i \). 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 \( \mathcal{A} \) cannot trace who is communicating with \( S \).

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

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

Table 5: Comparison of computational cost.

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