The Journal of Supercomputing (2020) 76:3086–3104
https://doi.org/10.1007/s11227-019-03086-z

Perfect forward secrecy via an ECC-based authentication scheme for SIP in VoIP

Mahdi Nikooghadam[1] · **Haleh Amintoosi**[1]

Published online: 25 November 2019

Abstract

With the advent of the internet, Voice over Internet Protocol (VoIP) has obtained a considerable amount of attention due to its low cost, and ease of implementation. Similar to other emerging technologies, VoIP faces several challenges, including security in terms of confidentiality, integrity, and authenticity. Specifically, there is a need for secure and efficient authentication and key agreement scheme to address the security requirements of communications over VoIP networks. Recently, Ravanbakhsh et al. have presented an authentication and key agreement protocol for VoIP networks. Here, in this article, we first prove that Ravanbakhsh et al.'s scheme cannot provide perfect forward secrecy. Next, we present an elliptic curve cryptography-based secure two-factor authentication and key agreement scheme. We analyze the security of the proposed scheme informally, and demonstrate that the proposed scheme can provide different security features, including perfect forward secrecy, and is robust against security attacks such as the impersonation attack, the replay attack, and the stolen-verifier attack. Furthermore, we simulate the protocol and analyze its security formally using Scyther tool. The results show its robustness against different attacks, and its ability to provide perfect forward secrecy. We compare the computation cost of the proposed scheme with the related schemes. Results show that the proposed scheme achieves a satisfiable performance comparable to other ECC-based methods.

Keywords Authentication · Key agreement · VoIP · Elliptic Curve Cryptography · Session Initiation Protocol (SIP) · Cryptanalysis · Perfect forward secrecy

✉ Haleh Amintoosi
amintoosi@um.ac.ir

Mahdi Nikooghadam
mahdi.nikooghadam@mail.um.ac.ir

[1] Faculty of Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

1 Introduction

Nowadays, different applications such as voice and video calls, and instant messaging have emerged which are based on Internet communication protocols. VoIP is a technology that presents the ability to deliver voice communications over IP networks. Compared to traditional phone networks, IP-based networks have various advantages, including low cost, scalability, and practical deployment through leveraging the existing infrastructure [1].

Voice over IP (VoIP) application uses the Session Initiation Protocol (SIP) for initiating, establishment, and stopping multimedia sessions. SIP, which has initially been developed by IETF in 1999 [2], is a signaling protocol, which is used by multimedia applications to create voice/video calls, and distribute multimedia [2]. In order to benefit from SIP services, the client first registers to the server by sending a message, including his secret information (e.g., his user name and password) via a secure communication channel. Once registered, the client is authorized to log in to the server by using the previously shared secrets. Then, in order to establish a session, the SIP session procedure locates the next SIP client, and a set of messages (as described below) are exchanged between the client and server:

- REQUEST: The client sends a connection request to the server.
- CHALLENGE: Upon receiving the request, the server sends a challenge message to client including random nonce, and the information required for the verification of the server validity.
- RESPONSE: Once the challenge message is received, the client verifies the server's legitimacy. If verified, it then sends a response message to the server. Once the server receives the response message, the server verifies the user's legitimacy, and if so, the client and the server share a session key.

Like all emerging technologies, VoIP faces challenges that need to be overcome to ensure that the technology is successfully deployed on a large scale [3]. The challenge in regard to security, in terms of confidentiality, integrity, and authenticity, is of particular importance. Authentication is the process of verifying an identity claimed by or for a user. Traditional authentication schemes usually relay on single factors (e.g., passwords) [4]. However, since VoIP is prone to various attacks including stolen-verifier, and offline password guessing attacks [5], considering a second factor such as smart card, hard to be forged, or copied, seems reasonable [6, 7].

In recent years, there exists considerable work on authentication and key agreement protocols presented for SIP-based VoIP networks [1, 8, 9]. Durlanik et al. [10] proposed an authentication scheme for SIP, using an elliptic curve cryptography (ECC)-based key exchange mechanism. Elliptic curve cryptography (ECC) is an encryption algorithm, which is leveraged by many authentication schemes, due to being effective, having shorter encryption key length, and the difficulty of elliptic curve discrete logarithm problem (ECDLP). Yoon et al. [11] demonstrated that Durlanik's authentication scheme is vulnerable against the stolen-verifier attack, offline password guessing attack, and Denning-Sacco attack, and proposed an ECC-based secure SIP authentication scheme to jointly exploit the key block size, speed, and security. Arshad and Ikram [12] claimed that the lightweight key management scheme presented by Tsai [13] is not robust against the stolen-verifier attack and the password guessing attack, and does not support perfect forward secrecy and known-key secrecy. In typical authentication and key agreement schemes, 'Perfect forward secrecy' refers to the feature which guarantees that if the server's secret key is compromised, it will not lead to the compromise of the session key. To address the challenges of Tsai's scheme, they further proposed an ECC-based mutual authentication protocol for SIP. Pu et al. [14] also proved that the scheme proposed by Arshad and Ikram [12] is vulnerable to the password guessing attack. After that, they proposed a secure protocol for authentication and key agreement in SIP, which was secure against password guessing attacks.

Zhang et al. [15] presented an efficient password-based authentication scheme for SIP supporting session key agreement, authentication, and password update, which does not require password table maintenance. Their proposed method was claimed to be secure against a series of attacks, including the stolen-verifier attack, man-in-the-middle attack, replay attack, Denning–Sacco attack, and offline dictionary attack. Later, Zhang et al. [16] showed that their scheme presented in [15] is prone to the impersonation attack, and to address the issue, they presented an extended version of their previous protocol [15]. Also, Jiang et al. [17] demonstrated that Zhang et al.'s scheme [15] is vulnerable against the malicious insider impersonation attack, and proposed an efficient scheme to address the issue. However, their proposed method was shown by Arshad and Nikooghadam [18] to be prone to the user impersonation attack. Irshad et al. [19] proposed an enhanced authentication scheme for SIP using a single round-trip to overcome the drawbacks of Zhang et al.'s protocol [15]. However, Arshad et al. [20] showed that Irshad et al.'s scheme [19] is prone to user impersonation attacks.

Tu et al. [21] showed that Zhang's scheme [15] is prone to the impersonation attack, and further proposed an enhanced scheme, and showed that the computational cost of their scheme in authentication phase is 75 % of Zhang et al.'s protocol. However, Farash [6] pointed out that the scheme proposed by Tue [21] is still prone to the impersonation attack. Farash [22] later showed that Zhang et al.'s scheme [15] is prone to the password changing attack and impersonation attack. He also proposed an enhanced authentication protocol for SIP.

Chaudhry et al. [8] demonstrated that Tu et al.'s scheme [21] is insecure in regard to denial-of-service, server impersonation, and replay attacks, and cannot guarantee user anonymity. Authors also pointed out that Farash's improvement [6] on Tu et al.'s scheme [21] is prone to the replay attack, and does not support user anonymity, and proposed a lightweight authentication and key agreement protocol, which was proved to be more secure. Mishra et al. [23] also performed cryptanalysis on Tue's scheme [21], and showed its weakness against server spoofing attack and man-in-the-middle attack. To address these drawbacks, they proposed an improved protocol, comparable to Tue et al.'s scheme regarding the communication and computational overhead.

Lu et al. [24] also showed the weaknesses of Farash's scheme [22], including lack of a pre-authentication in the smart card, and offline password guessing attack. To address Farash's scheme security issue, they then proposed an ECC-based anonymous modified scheme, and showed that their scheme is immune to the attacks mentioned for Farash's scheme. Zhang et al. [25] presented an authentication protocol for SIP networks that facilitated the authentication of the users via biometric verification. However, Irshad et al. [26] showed that Zhang's scheme [25] is prone to various attacks, including privileged insider attacks and denial-of-service attacks, and lacks forward secrecy compromise. They further presented an efficient authentication protocol countering Zhang et al. scheme [25]'s weaknesses. Zhang et al. [27] presented an efficient authentication scheme for SIP, which was shown to be secure against various attacks. However, Lu et al. [4] demonstrated that Zhang et al.'s scheme is insecure against insider attacks, and does not support mutual authentication. They also presented an enhanced authentication protocol to address the security flaws of Zhang et al. [27] protocol. Nikooghadam et al. [28] showed that the work presented by Chaudhry et al. [8] is not secure against the password guessing attack, and presented an enhanced scheme to overcome the weakness. Sureshkumar et al. [29] first showed that Lu et al.'s scheme [4] cannot support user anonymity and mutual authentication, and is vulnerable to the user/server impersonation attack. Next, they presented an improved authentication scheme, robust to identity and password guessing attacks in the random oracle model. Sourav et al. [9] showed the security limitations of Sureshkumar et al. [29] and Zhang et al. [27] schemes and then, presented an enhanced scheme to overcome Sureshkumar et al. scheme's flaws, without increasing the computational cost.

Ravanbakhsh et al. [30] crypt-analyzed the protocols presented by Chaudhry et al. [31] and Nikooghadam et al. [28], and showed their weakness in perfect forward secrecy provision. The authors also pointed out that Zhang et al.'s scheme [1] is vulnerable to replay and known-session-specific temporary information attacks, and does not provide user anonymity, and re-registration and revocation. Then, they proposed a two-factor authentication and key agreement protocol, resistant against various active and passive attacks. Although the authors claimed that their proposed scheme is secure, we demonstrate that it cannot provide perfect forward secrecy.

Our contribution is as follows:

- We carry out cryptanalysis of Ravanbakhsh et al.'s authentication scheme [30] for VoIP and show its weakness in providing perfect forward secrecy.
- We propose a secure authentication and key agreement protocol for SIP in VoIP based on elliptic curve cryptography that addresses the security flaws of the recent related schemes such as [30]. The proposed scheme can also provide perfect forward secrecy and user anonymity. We also prove that the proposed scheme is robust against various attacks including replay, Denning-Sacco, and impersonation.
- We formally analyze and prove the security of the proposed scheme using the Scyther tool [32]. We also run a performance analysis of the proposed scheme in terms of computational complexity and show that our scheme satisfies vari-

ous security features, while achieving a reasonable computational complexity, in comparison with related ECC-based authentication protocols.

The paper structure is as follows. Section 2 illustrates Ravanbakhsh et al.'s scheme [30] and discusses its security flaws. In Sect. 3, we present the details of the proposed authentication scheme. Section 4 shows the result of informal security analysis of the proposed scheme, as well as formal analysis via the Scyther tool. In Sect. 5, we analyze the performance of the proposed scheme, and compare it with the related works. The conclusion is presented in Sect. 6.

2 Review of Ravanbakhsh et al.'s scheme

In this section, we review the registration and authentication phases of Ravanbaksh et al.'s scheme [30], and discuss its security weaknesses. Ravanbakhsh et al.'s scheme [30] includes three phases: registration, authentication and key agreement, and password update. Table 1 shows the notations used in Ravanbakhsh et al.'s scheme.

2.1 Registration phase

The server and the user perform the following steps. At the end of the registration process, the server issues a smart card for the user.

- Step 1. The user selects ID_i, PW_i, and two random numbers r_i and b_i, then computes $RB = h(r_i||b_i)$ and $IP_i = h(h(ID_i||PW_i||B) \bmod \text{m})$. Then, the user sends $\{ID_i, IP_i, b_i\}$ to the server via a secure channel.
- Step 2. Once the registration request $\{ID_i, IP_i, b_i\}$ is received at the server, it calculates A_i, B_i and NID_i as $A_i = h(SID_i||x_s||ID_{sc}||ID_i)$, $B_i = A_i \oplus IP_i$

Table 1 Notations used in Ravanbakhsh et al.'s scheme [30]

Symbol	Description
U_i	User i
S	The SIP Server
ID_i	Identity of U_i
PW_i	Password of U_i
ID_{sc}	Identity of Smart Card
x_s	A high-entropy secret key of S
r_i, b_i, r_c, r_s	High-entropy random numbers
$\|\|$	Concatenation operation
$\oplus$	Bitwise (XOR) operation
SK	The shared one-time session key
T_1, T_2, T_3, T_4, T_5	The current time of user's system/server's system
$E_k(.)/D_k(.)$	The symmetric encryption/decryption with the key k
$h(.)$	A secure one-way hash function

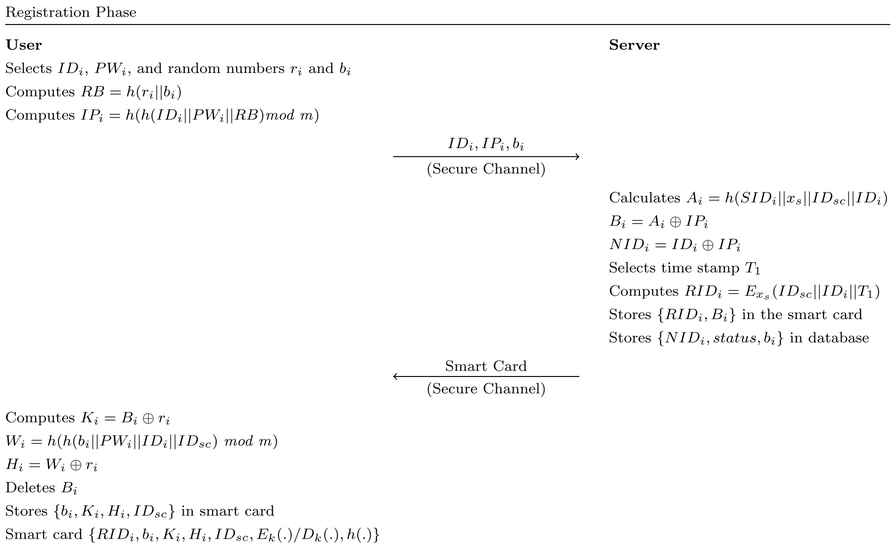
and $NID_i = ID_i \oplus IP_i$. It then selects time stamp T_1, and computes $RID_i = E_{x_s}(ID_{sc}||ID_i||T_1)$. The server stores RID_i and B_i in the smart card, and transmits it via a secure channel. The server also stores $(NID_i, status, b_i)$ in its database.

- Step 3. Upon receiving the smart card, the user computes $K_i = B_i \oplus r_i$, $W_i = h(h(b_i||PW_i||ID_i||ID_{sc})\text{mod m})$, and $H_i = W_i \oplus r_i$. Then, he deletes B_i and stores $\{b_i, K_i, H_i, ID_{sc}\}$ in the smart card. The smart card contains values $\{RID_i, b_i, K_i, H_i, ID_{sc}, E_k(.)/D_k(.), h(.)\}$.

Figure 1 shows the registration phase of Ravanbakhsh et al.'s scheme [30].

2.2 Authentication and key agreement phase

- Step 1. At first, the user inserts his smart card, and enters his ID_i and PW_i. Then, the smart card performs the following computations: $W_i^* = h(h(b_i||PW_i^*||ID_i^*||ID_{sc}^*)modm)$, $r_i^* = H_i \oplus W_i^*$, $B_i^* = K_i \oplus r_i^*$, $RB^* = h(r_i^*||b_i)$, $IP_i^* = h(h(ID_i^*||PW_i^*||RB^*)modm)$, $A_i^* = B_i^* \oplus IP_i^*$. Then, a time stamp T_2, and a random number c is chosen by the smart card. It then calculates $E_{A_i^*}(A_i^*||r_c||T_2||IP_i^*) = E_{A_i}$, and sends the request message $REQUEST\{EA_i, RID_i, T_2\}$ to the server.
- Step 2. Once the request message is received, the server first verifies the freshness of the message, and then decrypts RID_i with x_s as $D_{x_s}(RID_i) = (ID_{sc}^*||ID_i^*||T_1)$. Then, it calculates $A_i^* = h(SID_i||x_s||ID_{sc}||ID_i)$, decrypts EA_i with A_i^*, and obtains A_i, r_c, T_2', IP_i. Then, the server compares A_i with A_i^*. If equal, the server authenticates the user. The server calculates $NID_i = ID_i \oplus IP_i$, looks for NID_i^* in its database, and extracts $< NID_i, status, b_i >$. Then, the server selects a time



Registration Phase

User

Selects ID_i, PW_i, and random numbers r_i and b_i
Computes $RB = h(r_i||b_i)$
Computes $IP_i = h(h(ID_i||PW_i||RB)mod\ m)$

$\xrightarrow[\text{(Secure Channel)}]{ID_i, IP_i, b_i}$

Server

Calculates $A_i = h(SID_i||x_s||ID_{sc}||ID_i)$
$B_i = A_i \oplus IP_i$
$NID_i = ID_i \oplus IP_i$
Selects time stamp T_1
Computes $RID_i = E_{x_s}(ID_{sc}||ID_i||T_1)$
Stores $\{RID_i, B_i\}$ in the smart card
Stores $\{NID_i, status, b_i\}$ in database

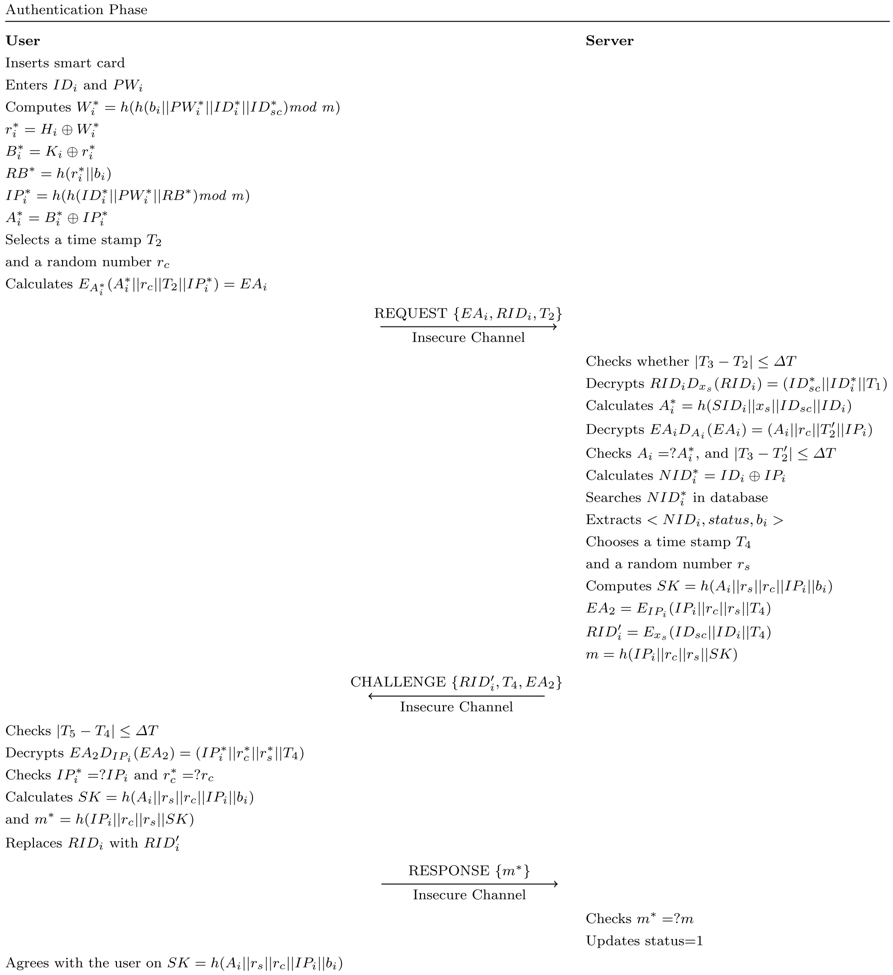
$\xleftarrow[\text{(Secure Channel)}]{\text{Smart Card}}$

Computes $K_i = B_i \oplus r_i$
$W_i = h(h(b_i||PW_i||ID_i||ID_{sc})\ mod\ m)$
$H_i = W_i \oplus r_i$
Deletes B_i
Stores $\{b_i, K_i, H_i, ID_{sc}\}$ in smart card
Smart card $\{RID_i, b_i, K_i, H_i, ID_{sc}, E_k(.)/D_k(.), h(.)\}$

Fig. 1 Registration phase of Ravanbakhsh et al. [30]

stamp T_4 and a random number r_s, and computes $SK = h(A_i||r_s||r_c||IP_i||b_i)$, $m = h(IP_i||r_c||r_s||SK)$, $EA_2 = E_{IP_i}(IP_i||r_c||r_s||T_4)$, $RID'_i = E_{x_s}(ID_{sc}||ID_i||T_4)$, and $m = h(IP_i||r_c||r_s||SK)$. It then sends the challenge message as:
$CHALLENGE$ $\{RID'_i, T_4, EA_2\}$.

- Step 3. Once the user receives the challenge message, he checks its freshness, and if so, the smart card decrypts EA_2 using IP_i, and extracts $(IP^*_i, r^*_c, r^*_s, T_4)$. Next, it checks whether $IP^*_i = IP_i$ and $r^*_c = r_c$ or not. If so, the server is authenticated, and the session key $SK = h(A_i||r_s||r_c||IP_i||b_i)$ and $m^* = h(IP_i||r_c||r_s||SK)$ are calculated. RID_i is also replaced by RID'_i. At the end, the user transmits the response message RESPONSE $\{m^*\}$ to the server via a public channel.
- Step 4. Upon receiving the response message, the server checks whether $m^* = ?m$, and if so, the server agrees with the user on the session key. Figure 2



Authentication Phase

User

Inserts smart card
Enters ID_i and PW_i
Computes $W^*_i = h(h(b_i||PW^*_i||ID^*_i||ID^*_{sc})mod\ m)$
$r^*_i = H_i \oplus W^*_i$
$B^*_i = K_i \oplus r^*_i$
$RB^* = h(r^*_i||b_i)$
$IP^*_i = h(h(ID^*_i||PW^*_i||RB^*)mod\ m)$
$A^*_i = B^*_i \oplus IP^*_i$
Selects a time stamp T_2
and a random number r_c
Calculates $E_{A^*_i}(A^*_i||r_c||T_2||IP^*_i) = EA_i$

$$\xrightarrow[\text{Insecure Channel}]{\text{REQUEST } \{EA_i, RID_i, T_2\}}$$

Server

Checks whether $|T_3 - T_2| \leq \varDelta T$
Decrypts $RID_i D_{x_s}(RID_i) = (ID^*_{sc}||ID^*_i||T_1)$
Calculates $A^*_i = h(SID_i||x_s||ID_{sc}||ID_i)$
Decrypts $EA_i D_{A_i}(EA_i) = (A_i||r_c||T'_2||IP_i)$
Checks $A_i =?A^*_i$, and $|T_3 - T'_2| \leq \varDelta T$
Calculates $NID^*_i = ID_i \oplus IP_i$
Searches NID^*_i in database
Extracts $< NID_i, status, b_i >$
Chooses a time stamp T_4
and a random number r_s
Computes $SK = h(A_i||r_s||r_c||IP_i||b_i)$
$EA_2 = E_{IP_i}(IP_i||r_c||r_s||T_4)$
$RID'_i = E_{x_s}(ID_{sc}||ID_i||T_4)$
$m = h(IP_i||r_c||r_s||SK)$

$$\xleftarrow[\text{Insecure Channel}]{\text{CHALLENGE } \{RID'_i, T_4, EA_2\}}$$

Checks $|T_5 - T_4| \leq \varDelta T$
Decrypts $EA_2 D_{IP_i}(EA_2) = (IP^*_i||r^*_c||r^*_s||T_4)$
Checks $IP^*_i =?IP_i$ and $r^*_c =?r_c$
Calculates $SK = h(A_i||r_s||r_c||IP_i||b_i)$
and $m^* = h(IP_i||r_c||r_s||SK)$
Replaces RID_i with RID'_i

$$\xrightarrow[\text{Insecure Channel}]{\text{RESPONSE } \{m^*\}}$$

Checks $m^* =?m$
Updates status=1

Agrees with the user on $SK = h(A_i||r_s||r_c||IP_i||b_i)$

Fig. 2 Authentication phase of Ravanbakhsh et al. [30]

depicts the authentication and key agreement phase of Ravanbakhsh et al.'s scheme [30].

2.3 Cryptanalysis of Ravanbakhsh et al.'s scheme

In this section, we point out in two scenarios that Ravanbakhsh et al.'s scheme does not provide perfect forward secrecy. As mentioned above, perfect forward secrecy assures that the session keys will not be compromised in the case where the server's secret key is compromised.

Scenario 1: If the adversary gets access to the server's secret key, i.e., x_s, he can obtain RID_i by decrypting RID_i (which has been sent on an insecure channel) with x_s as $D_{x_s}(RID_i) = (ID^*_{sc}||ID^*_i||T_1)$, and gets access to ID^*_{sc} and ID^*_i. The adversary also has access to SID, which is the server's public ID. So, he can calculate A^*_i as $A^*_i = h(SID_i||x_s||ID_{sc}||ID_i)$. Knowing A^*_i, the adversary can decrypt EA_i, and obtain IP_i and r_c as $D_{A_i}(EA_i) = (A_i||r_c||T'_2||IP_i)$. Now, since the adversary knows ID^*_i and IP_i, he can compute NID_i as $NID^*_i = ID_i \oplus IP_i$. Knowing NID_i, he can search it in the database, and get b_i. Moreover, by knowing IP_i, he can decrypt EA_2 as $D_{IP_i}(EA_2) = (IP^*_i||r^*_c||r^*_s||T_4)$, thus obtaining IP^*_i, r^*_c, r^*_s, T_4. So, all parameters included in session key $SK = h(A_i||r_s||r_c||IP_i||b_i)$ are achievable by the adversary.

Scenario 2: Similar to the above scenario, the adversary can decrypt RID_i with x_s, and gets access to ID^*_{sc} and ID^*_i. Knowing SID, he can calculate A^*_i, thus obtaining IP_i and r_c by decrypting EA_i with A_i. Knowing IP_i, he is able to decrypt EA_2 as $D_{IP_i}(EA_2) = (IP^*_i||r^*_c||r^*_s||T_4)$, thus obtaining IP^*_i, r^*_c, r^*_s, T_4. If the adversary obtains the smart card in some way (e.g., by finding the lost card or by stealing it), he can obtain b_i. Now, he has all the parameters of the session key, and can compute SK.

3 The proposed scheme

In order to address the drawbacks of Ravanbakhsh et al.'s [30] schemes, we propose a secure and efficient ECC-based protocol for SIP in VoIP which has the registration, authentication and key agreement, and password update stages. Table 2 shows the notations used in the proposed scheme.

3.1 Registration phase

The server and the user perform the steps described below. The result will be a smart card issued by the server for the user.

- Step 1. The user chooses an identity ID_i, a password PW_i, and two random numbers a_i and b_i. Then, he computes $MPW_i = h(PW_i||a_i) \oplus PW_i$ and sends (a_i, b_i, ID_i and MPW_i) to the server.
- Step 2. Upon receiving the parameters, the server calculates z_i, w_i and xw_i as $z_i = E_{q_s}(ID_i||b_i||a_i)$, $w_i = z_i \oplus MPW_i$, and $xw_i = h(z_i||MPW_i||ID_i)$. The server then stores $(xw_i, w_i, b_i, a_i, E_k(.)/D_k(.), h(.))$ in the smart card, and transmits it to the

Table 2 Notations used in the proposed scheme

Symbol	Description
U_i	User i
S	The SIP Server
ID_i	Identity of U_i
PW_i	Password of U_i
q_s	A high-entropy secret key of S
P	The base point of the elliptic curve
$Q_s = q_s.P$	Server's public key obtained from the secret key
a_i, b_i, r, t, y	High-entropy random numbers
$\|\|$	Concatenation operation
$\oplus$	Bitwise (XOR) operation
SK	The shared one-time session key
T_1, T_2, T_3, T_4	The current time of user's system/server's system
$E_k(.)/D_k(.)$	The symmetric encryption/decryption with the key k
$h(.)$	A secure one-way hash function
ΔT	The maximum transmission delay

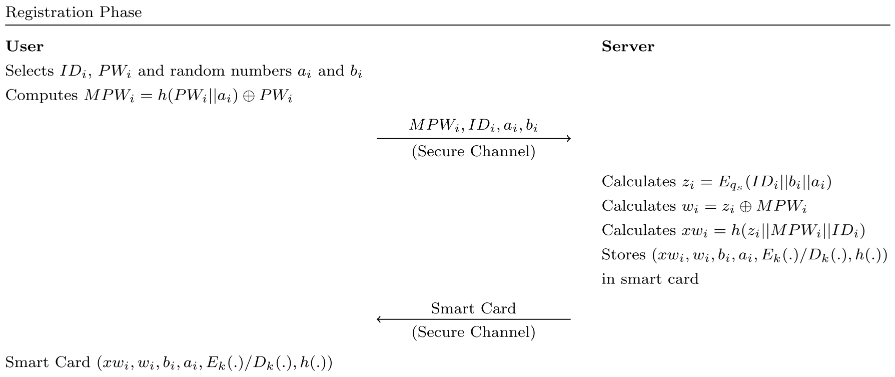

Registration Phase
User
Server
Selects ID_i, PW_i and random numbers a_i and b_i
Computes $MPW_i = h(PW_i||a_i) \oplus PW_i$
$\xrightarrow{MPW_i, ID_i, a_i, b_i}$
(Secure Channel)
Calculates $z_i = E_{q_s}(ID_i||b_i||a_i)$
Calculates $w_i = z_i \oplus MPW_i$
Calculates $xw_i = h(z_i||MPW_i||ID_i)$
Stores $(xw_i, w_i, b_i, a_i, E_k(.)/D_k(.), h(.))$
in smart card
$\xleftarrow{\text{Smart Card}}$
(Secure Channel)
Smart Card $(xw_i, w_i, b_i, a_i, E_k(.)/D_k(.), h(.))$

Fig. 3 Registration phase of the proposed scheme

user via a secure channel. Figure 3 depicts the registration phase of the proposed scheme.

3.2 Authentication phase

- Step 1. The user inserts his smart card, and enters his ID_i^* and PW_i^*. Then, the smart card performs the following computations:

$$
\begin{aligned}
&MPW_i^* = h(PW_i^*||a_i) \oplus PW_i^* \\
&\text{Chooses } r, t \in Z_n \\
&R = r.p \\
&T = t.p \\
&\text{Selects a time stamp } T_1 \\
&key_1 = r.Q_s \\
&c_i = a_i \oplus b_i \\
&z_i^* = w_i \oplus MPW_i^* \\
&xw_i^* = h(z_i^*||MPW_i^*||ID_i^*) \\
&E_1 = E_{key_1}(z_i^*||c_i||T_1||T)
\end{aligned}
$$

The user then transmits a request message $REQUEST\{E_1, R, T_1\}$ to the server through a public channel.

- Step 2. Once the server receives the request message, it first checks the freshness of the message by checking whether $|T_2 - T_1| \leq \Delta T$, and if not so, it terminates the session. Otherwise, the server creates $key_1' = R.q_s$. As mentioned above, $key_1 = r.Q_s = r.q_s.p$. On the other hand, $key_1\prime = R.q_s = r.q_s.p$. This means that $key_1' = key_1$. To authenticate the message received from the user, the server performs the following steps: (i) The server first decrypts E_1 with key_1' as $D_{key_1'}(E_1) = \{z_i^*||c_i^*||T_1^*||T^*\}$, and obtains parameters z_i^* and c_i^*. (ii) Then, having the secret key q_s, the server decrypts z_i^* as $z_i^* = E_{q_s}(ID_i^*||b_i^*||a_i^*)$ to obtain b_i^* and a_i^*, and compute $c_i' = a_i^* \oplus b_i^*$. (iii) If the computed c_i' from step (ii) equals to c_i^* obtained from decrypting E_1 in step (i), the user is authenticated. Next, the server performs the calculations stated below, and transmits a challenge message $CHALENGE(E_2, Y)$ to the user.

$$
\begin{aligned}
&\text{Chooses } y \in Z_n \\
&Y = y.p \\
&key_2 = R + T^* \\
&HID_i = h(T^*||ID_i^*||a_i^*||b_i^*) \\
&E_2 = E_{key_2}(HID_i||c_i^*)
\end{aligned}
$$

- Step 3. Upon receiving the challenge message, the smart card computes $key_2' = R + T$. Since $key_2' = key_2$, the smart card decrypts E_2 with key_2', and extracts (HID_i^*, c_i^*). On the other hand, since the smart card contains b_i, a_i, ID_i, T, it can compute HID_i as $HID_i = h(T||ID_i||a_i||b_i)$. It then checks whether $HID_i = ?HID_i^*$; if not, it terminates the session. Otherwise, the server is authenticated. The smart card then selects a time stamp T_3, computes $key_3 = t.Y$, and calculates the session key SK as $SK = h(ID_i||key_3||key_2\prime||key_1)$. It then adds two ECC points Y and R to have X $(X = Y + R)$, and encrypts X with key_3 as $E_3 = E_{key_3}(X, T_3)$. Finally, the message $RESPONSE(E_3, T_3)$ is forwarded to the server.

- Step 4. Upon receiving the response message, the server first selects the time stamp T_4, and checks the freshness of the message by checking whether $|T_4 - T_3| \leq \Delta T$. If so, it computes $key'_3 = y.T$, and decrypts E_3 with key'_3 to obtain X^*, T^*_3. On the other hand, the server has Y and R, so it can calculate $X = Y + R$. To authenticate the user, the server checks whether $X = ?X^*$, and if it holds, it creates the session key SK as $SK = h(ID_i||key'_3||key_2||key'_1)$. Figure 4 demonstrates the authentication phase of the proposed scheme.

3.3 Password update phase

In this phase, the user can securely change his password, and choose a new one. The steps are as below:

- Step 1. The user insets the smart card, and enters his current identity and password as ID^*_i and PW^*_i. Then, $MPW^*_i = h(PW^*_i||a_i) \oplus PW^*_i$ is computed.
- Step 2. The smart card first computes z^*_i and xw^*_i as $z^*_i = w_i \oplus MPW^*_i$, and $xw^*_i = h(z^*_i||MPW^*_i||ID^*_i)$, respectively, where w_i has been previously stored in the smart card. Next, the smart card compares xw^*_i with xw_i, and if equal, it verifies that the smart card belongs to the user.
- Step 3. Now, the user enters his new password PW^{**}_i. Then, the following parameters are computed:

$$\begin{aligned} MPW^{**}_i &= h(PW^{**}_i||a_i) \oplus PW^{**}_i \\ z^{**}_i &= w_i \oplus MPW^{**}_i \\ xw^{**}_i &= h(z^*_i||MPW^{**}_i||ID^*_i) \end{aligned}$$

At the end, xw^*_i is replaced with xw^{**}_i in the smart card.

4 Security analysis

In this section, we first analyze the security of the proposed scheme informally by investigating its robustness against different attacks. Then, we prove the security of the proposed protocol formally by the Scyther tool.

4.1 Informal security analysis

4.1.1 Anonymity

To preserve the anonymity of the user, his identity ID_i should not be exchanged in plain format. Moreover, if the adversary eavesdrops the $REQUEST(E_1, R, T_1)$, $CHALENGE(E_2, Y)$ or $RESPONSE(E_3, T_3)$ messages, or if he finds/steals the smart card, and gets access to its stored information as $(xw_i, w_i, b_i, a_i, E_k(.)/D_k(.), h(.))$, it should not be possible for the attacker to acquire the user's identity ID_i. As expressed in Sect. 3.1:

Authentication Phase

User

Inserts smart card

Enters ID_i^* and PW_i^*

Computes $MPW_i^* = h(PW_i^*||a_i) \oplus PW_i^*$

Chooses $r, t \in Z_n$

$R = r.p$

$T = t.p$

Selects a time stamp T_1

$key_1 = r.Q_s$

$c_i = a_i \oplus b_i$

$z_i^* = w_i \oplus MPW_i^*$

$xw_i^* = h(z_i^*||MPW_i^*||ID_i^*)$

Calculates $E_1 = E_{key_1}(z_i^*||c_i||T_1||T)$

$$\xrightarrow[\text{Insecure Channel}]{\text{REQUEST } \{E_1, R, T_1\}}$$

Server

Selects a time stamp T_2

Checks whether $|T_2 - T_1| \leq \Delta T$

$key_1' = R.q_s$

Decrypts $E_1 D_{key_1\prime}(E_1) = (z_i^*||c_i^*||T_1^*||T^*)$

Decrypts $z_i^* D_{q_s}(z_i^*) = (ID_i^*||b_i^*||a_i^*)$

Computes $c_i' = a_i^* \oplus b_i^*$

$c_i' = ?c_i^*$

$T_1^* = ?T_1$

Chooses $y \in Z_n$

Computes $Y = y.p$

Computes $key_2 = R + T^*$

Computes $HID_i = h(T^*||ID_i^*||a_i^*||b_i^*)$

Computes $E_2 = E_{key_2}(HID_i||c_i^*)$

$$\xleftarrow[\text{Insecure Channel}]{\text{CHALLENGE } \{E_2, Y\}}$$

Computes $key_2' = R + T$

Decrypts $E_2, D_{key_2'}(E_2) = \{HID_i^*, c_i^*\}$

Computes $HID_i = h(T||ID_i||a_i||b_i)$

Checks $HID_i = ?HID_i^*$

Selects a time stamp T_3

$key_3 = t.Y$

Computes $SK = h(ID_i||key_3||key_2\prime||key_1)$

Computes $X = Y + R$

Computes $E_3 = E_{key_3}(X, T_3)$

$$\xrightarrow[\text{Insecure Channel}]{\text{RESPONSE } \{E_3, T_3\}}$$

Selects a time stamp T_4

Checks whether $|T_4 - T_3| \leq \Delta T$

Calculates $key_3' = y.T$

$D_{key_3'}(E_3) = \{X^*, T_3^*\}$

Calculates $X = Y + R$

Checks $X = ?X^*$

Checks $T_3^* = ?T_3$

Computes $SK = h(ID_i||key_3'||key_2||key_1')$

Fig. 4 Authentication phase of the proposed scheme

$w_i = z_i \oplus MPW_i$
$z_i = E_{q_s}(ID_i||b_i||a_i)$
As shown above, the attacker should have access to the server's secret key q_s in order to decrypt z_i, and obtain ID_i. We have assumed that q_s has been kept secret. So, the proposed scheme preserves the anonymity of the user.

4.1.2 Known-key Secrecy

As mentioned above, the session key SK is $SK = h(ID_i||key_3||key_2||key_1)$, where $key_1 = R.q_s$, $key_2 = R + T^*$ and $key_3 = t.Y$. r, t, and y are random numbers specifically generated for each session, and are different with the ones in previous sessions. So, in case of the session key being revealed, the attacker cannot compute the session keys of other sessions.

4.1.3 Perfect forward secrecy

As mentioned earlier, perfect forward secrecy provision states that the attacker should not be able to retrieve the session key SK even if he acquires the user's password PW_i or the server's secret key q_s. In the proposed protocol, the session key is $SK = h(ID_i||key_3||key_2||key_1)$. So, the attacker cannot compute the session key, since only the user and the server know the values of key_1, key_2, and especially key_3. So, perfect forward secrecy is guaranteed.

4.1.4 Known-session-specific Temporary Information Attack

As mentioned in [30], resistance to this attack means if session random numbers a_i, b_i, r, t, y are unexpectedly disclosed to the attacker, he should not be able to retrieve the session key SK. In the proposed scheme, since the attacker cannot access the user's password PW_i and the server's secret key q_s, he cannot obtain the session key due to ID_i being included in the session key SK. So, the proposed scheme is secure against known-session-specific temporary information attacks.

4.1.5 Stolen-verifier attack

In the proposed protocol, since no parameters, including the user's password or the server's secret key, are stored in the database, the attacker cannot compute the session key even if he gets access to the database. Moreover, in case of the smart card being stolen or lost, the attacker cannot compute the session key SK, since it is dependant on the values of random numbers t, r, and y in SK, which are only in possession of the user and the server. So, the attacker cannot retrieve the verification information.

4.1.6 Offline password guessing attack

If the attacker can acquire the request, response, and challenge messages, he cannot obtain the user's password PW_i. Note that PW_i has not been exchanged anywhere in the protocol. Instead, at the registration phase, MPW_i as $MPW_i = h(PW_i||a_i) \oplus PW_i$ is calculated, and exchanged securely. Besides, PW_i cannot be obtained from MPW_i due to the characteristics of the hash function. So, the proposed scheme is secure against offline password guessing attacks.

4.1.7 Replay attack

Assume the attacker replays the old request message $REQUEST(E_1, R, T_1)$ to the server. In the proposed scheme, the server can discover that this message is repetitive and old in different ways: At first, the server verifies $|T_2 - T_1| \leq \Delta T$, and if this condition is not true, the session terminates. Even if the attacker changes T_1 with the current time T_1^{**}, and transmits $\{EA_i, RID_i, T_1^{**}\}$ to the server, the server decrypts E_1 with key_1, as $D_{key_1},(E_1) = (z_i^*||c_i^*||T_1^*||T^*)$, and compares T_1^* (obtained from decryption) with T_1^{**}. if not equal, the server understands that the timestamp has been altered. The same stands for $RESPONSE(E_3, T_3)$. So, the proposed scheme is secure against replay attacks.

4.1.8 Denning-Sacco attack

This attack refers to obtaining a long-term (e.g., the user's password or the session key), through an obtained old session key. In the proposed scheme, the session key $SK = h(ID_i||key_3||key_2||key_1)$, where key_1, key_2, and key_3 are random numbers. So, if the attacker acquires the old session key, he cannot compute the user's password or other session keys. Moreover, the proposed scheme is based on the elliptic curve discrete logarithm problem (ECDLP). So, the proposed scheme is robust against the Denning-Sacco attack.

4.1.9 User impersonation attack

To impersonate the user, the attacker has to create valid $REQUEST$ message and forward it to the server. To do so, he needs to calculate c_i^* and T_1^*. To compute a valid c_i^*, the attacker should get access to the server's secret key q_s, and the random number r. However, these values are kept secret by the user and server. So, the attacker cannot impersonate the user for the server.

4.1.10 Server impersonation attack

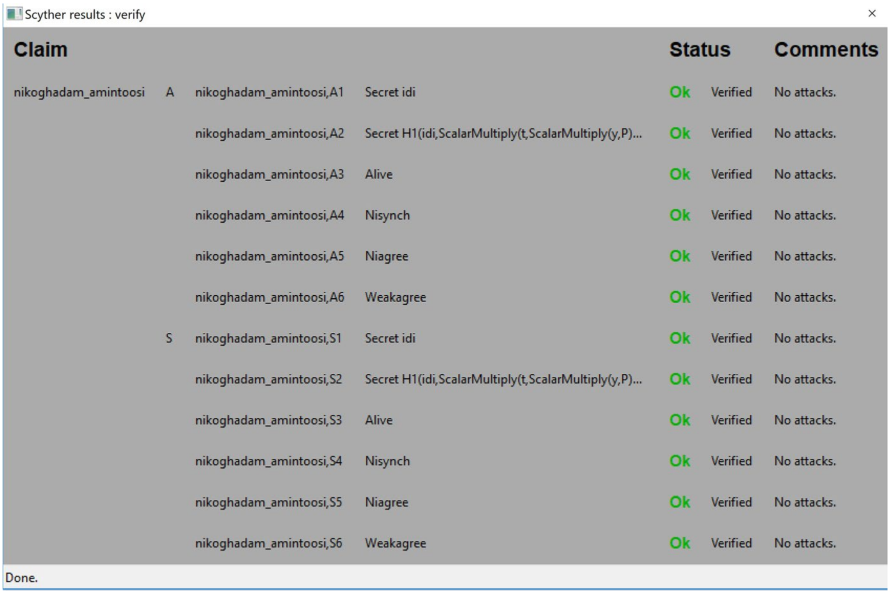
To impersonate the server, the attacker should create and send a valid challenge message as CHALLENGE $\{E_2, Y\}$ to the user. To do so, he needs to calculate E_2 as $E_2 = E_{key_2}(HID_i||c_i^*)$. To calculate E_2, he needs to know HID_i as $HID_i = h(T^*||ID_i^*||a_i^*||b_i^*)$, and to know HID_i, he should know ID_i^*. Finally, to know ID_i^*, he has to know to the server's secret key q_s as $z_i = E_{q_s}(ID_i||b_i||a_i)$, which is not accessible to the attacker due to being kept secret. Hence, the proposed scheme is secure against the server impersonation attack.

4.1.11 Insider attack

As expressed in the registration phase in Sect. 3.1, the user does not transmit his password directly to the server. Instead, MPW_i, IP_i, a_i, b_i are sent in the registration request message. So, if an insider obtains these parameters from the registration request message, he cannot obtain PW_i since $MPW_i = h(PW_i||a_i) \oplus PW_i$ has been sent instead, from which, PW_i cannot be acquired. Therefore, the proposed scheme resists the insider attacks.

4.2 Formal security analysis by Scyther tool

Scyther is a tool designed and extended to provide a means for formal analysis of the security protocols, and identification of their flaws and security requirements [32]. Scyther automatically analyzes the protocol, and investigates its behavior in regard to the potential attacks. Figure 5 demonstrates the Scyther output of the security analysis of the proposed scheme. The term *Claim* is used to specify security requirements *Alive*, *Nisynch*, *weakagree*, and *secret*. *Alive* ensures that an intended communication party *R* has executed some events. By *Nisynch*, we mean that the sender indeed has sent all sent messages, and the receiver has received all of them. *claim*(*R*; *secret*; *rt*) means that *R* claims that *rt* must be unknown to an adversary. Finally, *weakagree* ensures the robustness of the protocol against impersonation attacks. As demonstrated in Fig. 5, the proposed scheme can satisfy all the security requirements mentioned above.


Scyther results : verify

Claim				Status		Comments
nikoghadam_amintoosi	A	nikoghadam_amintoosi,A1	Secret idi	Ok	Verified	No attacks.
		nikoghadam_amintoosi,A2	Secret H1(idi,ScalarMultiply(t,ScalarMultiply(y,P)...	Ok	Verified	No attacks.
		nikoghadam_amintoosi,A3	Alive	Ok	Verified	No attacks.
		nikoghadam_amintoosi,A4	Nisynch	Ok	Verified	No attacks.
		nikoghadam_amintoosi,A5	Niagree	Ok	Verified	No attacks.
		nikoghadam_amintoosi,A6	Weakagree	Ok	Verified	No attacks.
	S	nikoghadam_amintoosi,S1	Secret idi	Ok	Verified	No attacks.
		nikoghadam_amintoosi,S2	Secret H1(idi,ScalarMultiply(t,ScalarMultiply(y,P)...	Ok	Verified	No attacks.
		nikoghadam_amintoosi,S3	Alive	Ok	Verified	No attacks.
		nikoghadam_amintoosi,S4	Nisynch	Ok	Verified	No attacks.
		nikoghadam_amintoosi,S5	Niagree	Ok	Verified	No attacks.
		nikoghadam_amintoosi,S6	Weakagree	Ok	Verified	No attacks.

Done.

Fig. 5 Security analysis of the proposed scheme using Scyther

5 Performance analysis

In this section, we present the results of the performance analysis of the proposed scheme. At first, we compare the performance of the proposed scheme with Farash [6], Tu et al. [21], Zhang et al. [1], Jiang et al. [17], Nikooghadam et al. [28], and Ravanbakhsh et al. [30], considering various security features. Then, we compute the computational complexity of the proposed scheme and compare it with the schemes mentioned above.

Table 3 demonstrates the analysis of security features for the proposed protocol in comparison with the recent related works. As can be seen, the proposed protocol is secure against all mentioned attacks, and can provide security requirements, including as perfect forward secrecy and known-key secrecy. In other words, the proposed scheme provides a high security level, compared to the related protocols.

Table 4 depicts the notations used in the computational cost comparison. We utilized the experimental results presented in [7, 33] to estimate the approximate execution timings. Specifically, the approximate execution timings of $T_{hf}, T_{mu}, T_{ad}, T_{en/d}$ are 0.0004 ms, 7.3529 ms, 0.009 ms, and 0.1303 ms, respectively. Regarding the user side of the proposed scheme, four scalar multiplication operations, three symmetric encryption operations, five hash function operations, and two point addition operations are required. So, the computational cost at the user side is $4T_{mu} + 5T_{hf} + 3T_{en/d} + 2T_{ad}$. On the other hand, three scalar multiplication operations, three hash function operations, five symmetric encryption operations, and two point addition operations are required at the server side. So, the computational cost at the serve side is $3T_{mu} + 3T_{hf} + 5T_{en/d} + 2T_{ad}$. Table 5 and Fig. 6 demonstrate the computational time of the proposed scheme, as well as Farash [6], Tu et al. [21],

Table 3 Comparison of security features

Security features	[6]	[21]	[1]	[17]	[28]	[30]	Ours
F_1	No	No	No	Yes	Yes	Yes	Yes
F_2	Yes	Yes	Yes	Yes	Yes	Yes	Yes
F_3	Yes	Yes	Yes	Yes	Yes	Yes	Yes
F_4	No	No	Yes	No	Yes	Yes	Yes
F_5	No	No	Yes	No	Yes	Yes	Yes
F_6	Yes	Yes	Yes	Yes	Yes	Yes	Yes
F_7	No	No	No	Yes	Yes	Yes	Yes
F_8	Yes	Yes	No	Yes	Yes	Yes	Yes
F_9	Yes	Yes	Yes	Yes	Yes	Yes	Yes
F_{10}	Yes	Yes	Yes	Yes	No	No	Yes
F_{11}	Yes	No	Yes	Yes	Yes	Yes	Yes

F_1: provide user anonymity and un-traceability; F_2: resists privileged insider attack; F_3: resists Denning-Sacco attack; F_4: resists user impersonation attack; F_5: resists server impersonation attack; F_6: resists off/online password guessing attack; F_7: resists replay attack; F_8: resists session-specific temporary information attack; F_9: provides known-key secrecy; F_{10}: provides perfect forward secrecy; F_{11}: provides efficient password changing

Table 4 Notations used in the performance analysis of the proposed scheme

Symbol	Description
T_{hf}	Time of performing a hash function operation
$T_{en/d}$	Time of performing symmetric encryption/decryption
T_{mu}	Time of performing the scalar multiplication operation of elliptic curve
T_{ad}	Time of performing a point addition operation of elliptic curve

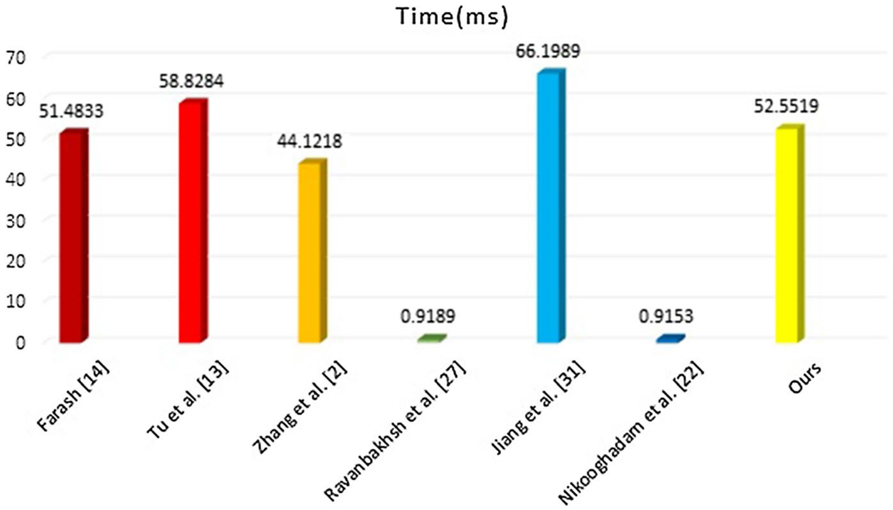


Fig. 6 Comparison of execution time in (milliseconds) between our proposed scheme and other schemes

Table 5 Computation cost comparison between the proposed protocol and related works

Scheme	User's computation	Server's computation	Total computation	Time (ms)
[6]	$4T_{mu} + 5T_{hf} + 1T_{ad}$	$3T_{mu} + 5T_{hf}$	$7T_{mu} + 10T_{hf} + 1T_{ad}$	51.4833
[21]	$4T_{mu} + 6T_{hf}$	$4T_{mu} + 7T_{hf}$	$8T_{mu} + 13T_{hf}$	58.8284
[1]	$3T_{mu} + 5T_{hf}$	$3T_{mu} + 6T_{hf}$	$6T_{mu} + 11T_{hf}$	44.1218
[30]	$12T_{hf} + T_{en/d}$	$5T_{hf} + 6T_{en/d}$	$17T_{hf} + 7T_{en/d}$	0.9189
[17]	$4T_{mu} + 6T_{hf} + T_{ad}$	$5T_{mu} + 6T_{hf} + T_{ad}$	$9T_{mu} + 12T_{hf} + 2T_{ad}$	66.1989
[28]	$5T_{hf} + 2T_{en/d}$	$3T_{hf} + 5T_{en/d}$	$8T_{hf} + 7T_{en/d}$	0.9153
Ours	$4T_{mu} + 5T_{hf} + 3T_{en/d} + 2T_{ad}$	$3T_{mu} + 3T_{hf} + 5T_{en/d} + 2T_{ad}$	$7T_{mu} + 8T_{hf} + 8T_{en/d} + 4T_{ad}$	52.5519

Zhang et al. [1], Jiang et al. [17], Nikooghadam et al. [28], and Ravanbakhsh et al. [30]. The results prove that the proposed scheme has shown a comparable performance with other ECC-based schemes. To be specific, the total computation time of the proposed protocol is 52.5519 *ms*, while it is 51.4833 *ms* for Farash [6], 58.8284 *ms* for Tu et al. [21], and 66.1989 *ms* for Jiang et al. [17]. It should be noted that the total computation time of the proposed scheme and other ECC-based schemes [6, 17, 21] is higher than non-ECC-based schemes, due to the high computation cost of point multiplication operation. However, as demonstrated in Table 3, the proposed scheme is resistant against almost all security threats, compared to other ECC-based and non-ECC-based methods.

To be brief, the proposed authentication and key agreement scheme outperforms the related protocols by achieving a delicate balance between the security and the performance, while showing a satisfiable computational performance and providing perfect forward secrecy.

6 Conclusion

In this article, we first analyzed Ravanbakhsh et al.'s authentication and key agreement scheme proposed for SIP and showed that it could not provide the perfect forward secrecy. We then present an ECC-based secure two-factor authentication and key agreement scheme for SIP. We formally analyzed the security of the proposed scheme and proved the robustness of the proposed scheme against various attacks, and its ability to provide various security features. We also demonstrated that the proposed scheme achieves a satisfying computational time compared to other ECC-based schemes.

References

1. Zhang L, Tang S, Zhu S (2016) An energy efficient authenticated key agreement protocol for SIP-based green VoIP networks. J Netw Comput Appl 59:126–133
2. Franks J, Hallam-Baker P, Hostetler J, Lawrence S, Leach P, Luotonen A, Stewart L (1999) HTTP authentication: basic and digest access authentication. Internet RFC 2617
3. Butcher D, Li X, Guo J (2007) Security challenge and defense in VoIP infrastructures. IEEE Trans Syst Man Cybern Part C (Appl Rev) 37(6):1152–1162
4. Lu Y, Li L, Peng H, Yang Y (2016) A secure and efficient mutual authentication scheme for session initiation protocol. Peer-to-Peer Netw Appl 9(2):449–459
5. Yang CC, Wang RC, Liu WT (2005) Secure authentication scheme for session initiation protocol. Comput Secur 24(5):381–386
6. Farash MS (2016) Security analysis and enhancements of an improved authentication for session initiation protocol with provable security. Peer-to-Peer Netw Appl 9(1):82–91
7. Amin R, Islam SH, Biswas GP, Khan MK, Obaidat MS (2015) Design and analysis of an enhanced patient-server mutual authentication protocol for telecare medical information system. J Med Syst 39(11):137
8. Chaudhry SA, Naqvi H, Sher M, Farash MS, Hassan MU (2017) An improved and provably secure privacy preserving authentication protocol for sip. Peer-to-Peer Netw Appl 10(1):1–15
9. Sourav S, Odelu V, Prasath R (2019) Enhanced Session Initiation Protocols for Emergency Healthcare Applications. In: Thampi S, Madria S, Wang G, Rawat D, Alcaraz Calero J (eds) Security in Computing and Communications, vol 969. SSCC 2018. Communications in Computer and Information Science
10. Durlanik A, Sogukpinar I (2005) SIP authentication scheme using ECDH. Screen 137:3367

11. Yoon EJ, Yoo KY, Kim C, Hong YS, Jo M, Chen HH (2010) A secure and efficient SIP authentication scheme for converged VoIP networks. Comput Commun 33(14):1674–1681
12. Arshad R, Ikram N (2013) Elliptic curve cryptography based mutual authentication scheme for session initiation protocol. Multimed Tools Appl 6(2):165–178
13. Tsai JL (2009) Efficient nonce-based authentication scheme for session initiation protocol. IJ Netw Secur 9(1):12–16
14. Pu Q, Wang J, Wu S (2013) Secure SIP authentication scheme supporting lawful interception. Secur Commun Netw 6(3):340–350
15. Zhang L, Tang S, Cai Z (2014) Efficient and flexible password authenticated key agreement for Voice over Internet Protocol Session Initiation Protocol using smart card. Int J Commun Syst. 7(11):2691–2702
16. Zhang L, Tang S, Cai Z (2014) Cryptanalysis and improvement of password-authenticated key agreement for session initiation protocol using smart cards. Secur Commun Netw 7(12):2405–2411
17. Jiang Q, Ma J, Tian Y (2015) Cryptanalysis of smart-card-based password authenticated key agreement protocol for session initiation protocol of Zhang et al. Int J Commun Syst 28(7):1340–1351
18. Arshad H, Nikooghadam M (2015) Security analysis and improvement of two authentication and key agreement schemes for session initiation protocol. J Supercomput 71(8):3163–3180
19. Irshad A, Sher M, Rehman E, Ch SA, Hassan MU, Ghani A (2015) A single round-trip sip authentication scheme for voice over internet protocol using smart card. Multimed Tools Appl 74(11):3967–3984
20. Arshad H, Nikooghadam M (2016) An efficient and secure authentication and key agreement scheme for session initiation protocol using ECC. Multimed Tools Appl 75(1):181–197
21. Tu H, Kumar N, Chilamkurti N, Rho S (2015) An improved authentication protocol for session initiation protocol using smart card. Peer-to-Peer Netw Appl 8(5):903–910
22. Farash MS, Attari MA (2016) An anonymous and untraceable password-based authentication scheme for session initiation protocol using smart cards. Int J Commun Syst 29(13):1956–1967
23. Mishra D, Das AK, Mukhopadhyay S (2016) A secure and efficient ECC-based user anonymity preserving session initiation authentication protocol using smart card. Peer-to-peer Netw Appl 9(1):171–192
24. Lu Y, Li L, Peng H, Yang Y (2017) An anonymous two-factor authenticated key agreement scheme for session initiation protocol using elliptic curve cryptography. Multimed Tools Appl 76(2):1801–1815
25. Zhang L, Tang S, Zhu S (2017) Privacy-preserving authenticated key agreement scheme basedon biometrics for session initiation protocol. Wirel Netw 23(6):1901–1916
26. Irshad A, Chaudhry SA, Kumari S, Usman M, Mahmood K, Faisal MS (2017) An improved lightweight multi-server authentication scheme. Int J Commun Syst 30:e3351
27. Zhang Z, Qi Q, Kumar N, Chilamkurti N, Jeong HY (2015) A secure authentication scheme with anonymity for session initiation protocol using elliptic curve cryptography. Multimed Tools Appl 74(10):3477–3488
28. Nikooghadam M, Jahantigh R, Arshad H (2017) A lightweight authentication and key agreement protocolpreserving user anonymity. Multimed Tools Appl 76(11):13401–13423
29. Sureshkumar V, Amin R, Anitha R (2018) A robust mutual authentication scheme for session initiation protocol with key establishment. Peer-toPeer Netw Appl 11(5):900–916
30. Ravanbakhsh N, Mohammadi M, Nikooghadam M (2018) Perfect forward secrecy in VoIP networks through design a lightweight and secure authenticated communication scheme. Multimed Tools Appl 78:11129–11153
31. Chaudhry SA, Farash MS, Naqvi H, Kumari S, Khan MK (2015) An enhanced privacy preserving remote user authentication scheme with provable security. Secur Commun Netw 8(18):3782–3795
32. Cremers C (2006) Scyther, Semantics and Verification of Security Protocols. Ph.D. dissertation, Eindhoven University of Technology
33. Xu L, Wu F (2015) Cryptanalysis and improvement of a user authentication scheme preserving uniqueness and anonymity for connected health care. J Med Syst 39:10