Design of a password-based authenticated key exchange protocol for SIP

Dheerendra Mishra[1]

Received: 12 December 2014 / Revised: 15 May 2015 / Accepted: 24 August 2015 /
Published online: 29 September 2015

Abstract The Session Initiation Protocol (SIP) is a signaling communications protocol, which has been chosen for controlling multimedia communication in 3G mobile networks. In recent years, password-based authenticated key exchange protocols are designed to provide strong authentication for SIP. In this paper, we address this problem in two-party setting where the user and server try to authenticate each other, and establish a session key using a shared password. We aim to propose a secure and anonymous authenticated key exchange protocol, which can achieve security and privacy goal without increasing computation and communication overhead. Through the analysis, we show that the proposed protocol is secure, and has computational and computational overheads comparable to related authentication protocols for SIP using elliptic curve cryptography. The proposed protocol is also provably secure in the random oracle model.

Keywords Session initiation protocol (SIP) · Elliptic curve cryptography (ECC) · Authentication · Key agreement · Anonymity

1 Introduction

Session Initiation Protocol (SIP) is a signaling protocol, which operates on the User Datagram Protocol (UDP) and Transmission Control Protocol (TCP) at the application layer in order to maintain, initiate and terminate the multimedia sessions [27]. SIP supports the multimedia services (transmission of voice and video) on both wired as well as wireless networks, which has gained a wide popularity. However, there are emerging security challenges in SIP such as authentication and confidentiality by knowing the fact that an adversary can fully control the public channels. The authenticity ensures the correctness of the partici-

✉ Dheerendra Mishra
dheerendra.mishra@lnmiit.ac.in

[1] Department of Mathematics, The LNM Institute of Information Technology, Jaipur, 302031, India

pants, whereas the confidentiality is used to achieve data security and integrity. In order to ensure the authorized access of resources, the server verifies the authenticity of the user. However, one way authentication is not enough in these services as an adversary can perform server-spoofing attack [11, 24, 26, 40]. On the contrary, the mutual authentication ensures the authenticity of the sender and the receiver, where the user and the server can verify the authenticity of each other [16]. To achieve confidentiality, usually transmitted data is encrypted using the established session key between the user and server. To establish session key, key agreement mechanism is used, where the user and server compute a common session key using their shared secrets.

Security challenges in SIP are emerging with advancement in computing technology [2, 28, 31, 35]. In recent years, several SIP authentication and key agreement protocols have been designed and developed to satisfy desirable security attributes. In earlier proposed schemes for SIP, the server uses a challenge-response mechanism to verify the authenticity of the user. Unfortunately, these schemes support one-way authentication in which the server can verify user's authenticity, but the user can not verify the correctness of source. This enables an adversary to masquerade as the trusted server to achieve user's secret information. In 2005, Yang et al. [38] showed that the earlier proposed authentication schemes for SIP are vulnerable to the server-spoofing and the off-line password guessing attacks. To fix the security pitfalls of SIP authentication schemes, Yang et al. introduced a new authentication scheme for SIP, which is based on the Diffie-Hellman key exchange protocol. Their scheme supports mutual authentication and session key agreement. Although Huang et al. [17] showed that Yang et al.'s scheme does not withstand off-line password guessing attack. To enhance the security of Yang et al.'s scheme, Huang et al. also presented an improved scheme. Letter, Jo et al. [21] found that proposed scheme by Huang et al.'s does not withstand the off-line password guessing attack. Durlanik et al. [9] proposed an efficient and improved authentication scheme for SIP using the elliptic curve cryptography (ECC) in order to overcome the weaknesses in Yang et al.'s scheme. Wu et al. [34] also presented an authentication scheme for SIP using ECC. Later on, Yoon et al. [41] demonstrated that both Durlanik et al.'s scheme and Wu et al.'s scheme do not withstand the stolen-verifier attack and the off-line password guessing attack. In addition, Yoon et al. introduced an ECC-based improved authentication scheme for SIP. However, Gokhroo et al. [13] and Pu [25] pointed out that Yoon et al.'s scheme is still vulnerable to the off-line password guessing attack and does not withstand the replay attack. To enhance the security of Yoon et al.'s scheme, Tsai et al. [32] proposed an improved authentication scheme for SIP based on random nonce. Unfortunately, Yoon et al. [42] identified that Tsai et al.'s scheme cannot resist the off-line password guessing attack and the stolen-verifier attack. To overcome these weaknesses, Yoon et al. further proposed an improved scheme for SIP. Later, Xie et al. [36] found that Yoon et al.'s scheme is still vulnerable to the off-line password guessing and stolen-verifier attacks. Arshad et al. [3] also pointed out the vulnerability of Tsai's scheme, and showed that the off-line password guessing and stolen verifier attacks are possible on Tsai's scheme. To erase security flaws of Tsai's scheme, Arshad et al. proposed an ECC-based improved scheme, which is later shown to be insecure against the off-line password guessing attack by He et al. [15]. Zhang et al. [43] proposed an efficient smart card-based authentication and key agreement scheme for SIP. Tu et al. [33] pointed out that that Zhang et al.'s scheme cannot withstand the impersonation attack. To enhance the security of Zhang et al.'s scheme, Tu et al. proposed an improved authentication scheme for SIP. Their scheme is more efficient than Zhang et al.'s protocol as computational cost in the authentication phase of their scheme is about 75 % of Zhang et al.'s scheme. Recently, Farash [10] pointed out that Tu et al.'s scheme is insure against impersonation attack. He also proposed an improved scheme to withstand impersonation attack without increasing computation overhead. Jiang et al. [20] showed that Zhang et al.'s scheme is insecure against the malicious insider impersonation attack. Furthermore, they proposed an improved scheme to overcome the security weakness found in Zhang et al.'s scheme. However, Jiang et al.'s scheme does not present password change phase. Irshad et al. [18] also pointed out that Zhang et al.'s scheme is without any password-verifier database using smart card. They proposed single round authentication and key-agreement protocol which allows the involved parties to authenticate in a single round-trip of exchanged messages. Recently, Arshad and Nikooghadam [4] demonstrated that Irshad et al.'s scheme is insure against user impersonation attack. Moreover, ECC-based authentication and key agreement scheme for SIP is proposed without smart card. Yeh et al. [39] analyzed the Diffie-Hellman (DH) based authentication protocols for SIP to enhance the security in the current SIP authentication mechanism. They proposed an ECC-based authentication protocol for SIP, which is more efficient as compared to DH-based authentication protocols for SIP.

The ECC-based SIP authentication schemes [10, 20, 39] are secure against various attacks and have low computation cost compare to DH-based authentication protocols for SIP. But, these schemes do not support anonymous authentication which increases the possibility of ID-theft and other privacy concerns. However, a scalable authentication scheme for SIP should protect user's anonymity. Recently, Zhang et al. [44] proposed an anonymous authentication scheme based on Farash and Attari's work [12]. Their scheme can efficiently protect user's privacy, but it is vulnerable to insider attack, where a malicious insider can know the user's password. It increases the possibility of illegal access of user's accounts, which are protected using the same passwords. Additionally, server responses to any login request with a challenge message without identifying the validity of requester. Due to this drawback, the server may be flooded with fake requests. In this paper, we propose an improved scheme, which keeps the merits of anonymity and efficiency. Through the rigorous formal and informal security analysis, we show that our scheme is secure against various known attacks including the attacks discussed in existing authentication schemes for SIP. We also give the proof of security in the random oracle model.

The rest of the article is arranged as follows: In next Section, we recall some basic mathematical preliminaries and define some notations. In Section 3, we present password-based two party authenticated key exchange protocol (2PAKE) for SIP. Correctness of mutual authentication is demonstrated in Section 4. Security analysis is given in Section 5. We compare the security features and performance of our scheme with other related schemes in Section 6. Finally, we conclude the paper in Section 7.

2 Mathematical preliminaries

In this section, we briefly discuss some basic mathematical preliminaries. We discussed the meaning of symbols in Table 1 and abbreviations in Table 2.

2.1 BAN logic

BAN logic [7, 30] is applied to show the correctness of mutual authentication between the user and server. Using BAN logic, one can show that the user and server determine whether the exchanged information is secured and trustworthy against eavesdropping. It comprises the verification of message origin, message freshness and the origin's trustworthiness. Some notations used in BAN logic analysis are described as follows:

Table 1 List of symbols

Symbols	Descryption
U_i	User i
S	A trustworthy server
ID_i	Unique identity of user i
PW_i	Unique password of user i
T_i	Timestamp generated by user i
T_s	Timestamp generated by S
ΔT	Maximum transmission delay
mk	Master key of S
sk	Session key
p	A n-bit prime number, where n is security parameter
$E_p(a, b)$	An elliptic curve $y^2 = x^3 + ax + b \pmod p$ over a finite field Z_p with $4a^3 + 27b^2 \neq 0 \pmod p$
G	Additive group of points of $E_p(a, b)$, whose order is n
P	A generator of G
P^x	The x-coordinate of a point $P \in E_p(a, b)$
$h(\cdot)$	One-way hash functions $h : \{0, 1\}^* \to \{0, 1\}^n$
P	A point on the elliptic curve $E_p(a, b)$
$\oplus$	XOR
$\|\|$	String concatenation operation

- $P|\equiv X$: The principal P believes the statement X.
- $P \lhd X$: P sees X, means that P has received a message combine X.
- $P|\sim X$: P once said X, means that $P|\equiv X$ when P sent it.
- $P|\Rightarrow X$: P controls X, P has an authority on X (Jurisdiction over X).
- $\sharp(X)$: The message X is fresh.
- $P|\equiv Q \xleftrightarrow{k} P$: P and Q use K (shared key), to communicate with each other.
- $A \xleftrightarrow{x} B$: x is a shared secret information between A and B.
- $\{X\}_K$: The formula X is encrypted under k.
- $< X >_Y$: The formula X is combined with formula Y.
- $(X)_K$: The formula X is hashed with the key K.
- $\xrightarrow{k} P$: K is public key of P.
- $P \overset{X}{\rightleftharpoons} Q$: X is a secret formula, known only to P and Q.

Table 2 List of abbreviations

Abbreviation	Description
ECC	Elliptic curve cryptography (ECC)
SIP	Session initiation protocol
BAN logic	Burrows, Abadi and Needham logic
DC	Discrete logarithm
CDH	Computational Diffie-Hellman

In order to describe logical postulates of BAN logic in formal terms [6, 7], following rules are defined below:
Rule (1). Message meaning rule:

$$\frac{P|\equiv Q \overset{K}{\longleftrightarrow} P, P \lhd \{X\}_k}{P|\equiv Q| \sim X} \tag{1}$$

If P believes that K is shared with Q and sees X encrypted under k, then P believes that Q once said X.
Rule (2). The nonce verification rule:

$$\frac{P|\equiv \sharp(X), P|\equiv Q|\sim X}{P|\equiv Q|\equiv X} \tag{2}$$

If P believes that X has been uttered recently (freshness) and P believes that Q once said X, and then P believes that Q believes X.
Rule (3). The jurisdiction rule:

$$\frac{P|\equiv Q|\equiv X, P|\equiv Q|\Rightarrow X}{P|\equiv X} \tag{3}$$

If P believes that Q has jurisdiction over X, and P believes that Q believes a message X, then P believes X.
Rule (4). The freshness rule:

$$\frac{P|\equiv \sharp(X)}{P|\equiv \sharp(X, Y)} \tag{4}$$

If one part known to be fresh, then the entire formula must be fresh.

2.2 Collision-resistant one-way hash function

A collision-resistant one-way hash function is defined in [5, 29] as follows.

Definition 1 (Collision-resistant one-way hash function) A collision-resistant one-way hash function $h : X \to Y$, where $X = \{0, 1\}^*$ and $Y = \{0, 1\}^n$ is a deterministic algorithm that takes an input as an arbitrary length binary string $x \in \{0, 1\}^*$ and outputs a binary string $y \in \{0, 1\}^n$ of fixed-length n. If we denote $Adv_{\mathcal{A}}^{HASH}(t)$ as an adversary $\mathcal{A}$'s advantage in finding collision, we then have

$$\begin{aligned} Adv_{\mathcal{A}}^{HASH}(t) = Pr[&(x, x') \in_R \mathcal{A} : \\ &x \neq x' \; and \; h(x) = h(x')], \end{aligned}$$

where $Pr[E]$ denotes the probability of a random event E and $(x, x') \in_R \mathcal{A}$ denotes the pair (x, x') is selected randomly by $\mathcal{A}$. In this case, the adversary $\mathcal{A}$ is allowed to be probabilistic and the probability in the advantage is computed over the random choices made by the adversary $\mathcal{A}$ with the execution time t. We call such a hash function $h(\cdot)$ is collision-resistant, if $Adv_{\mathcal{A}}^{HASH}(t) \leq \epsilon_1$, for any sufficiently small $\epsilon_1 > 0$.

2.3 Elliptic curve over a prime field

A non-singular elliptic curve $y^2 = x^3 + ax + b$ over the finite field $GF(p)$ is considered as the finite set $E_p(a, b)$ of solutions $(x, y) \in Z_p \times Z_p$ to the congruence $y^2 = x^3 + ax + b \pmod p$, where $a, b \in Z_p$ are constants chosen such that the condition $4a^3 + 27b^2 \neq 0 \pmod p$ is satisfied, together with a special point $\mathcal{O}$ called the point at infinity or zero

point, where $Z_p = \{0, 1, \ldots, p-1\}$ and $p > 3$ be a prime. The total number of points on the elliptic curve $E_p(a, b)$, which is denoted by $|E|$, satisfies the inequality [22]: $p+1-2\sqrt{p} \leq |E| \leq p+1+2\sqrt{p}$. Thus, we can say that an elliptic curve $E_p(a, b)$ over Z_p has roughly p points. Furthermore, $E_p(a, b)$ forms an commutative group under addition modulo p operation.
G be a additive group of points of $E_p(a, b)$, whose order is n. Assume that $P = (x_P, y_P)$ and $Q = (x_Q, y_Q)$ are two points on elliptic curve $y^2 = x^3 + ax + b \,(\text{mod}\, p)$. Then $R = (x_R, y_R) = P + Q$ is computed as follows [23]:

$$\begin{aligned} x_R &= (\gamma^2 - x_P - x_Q)(\text{mod}\ p),\\ y_R &= (\gamma(x_P - x_R) - y_P)(\text{mod}\ p),\\ \text{where}\ \gamma &= \begin{cases} \frac{y_Q - y_P}{x_Q - x_P}\ (\text{mod}\ p), \text{if}\ P \neq Q \\ \frac{3x_P{}^2 + a}{2y_P}\ (\text{mod}\ p), \text{if}\ P = Q. \end{cases} \end{aligned}$$

In elliptic curve cryptography, multiplication is defined as the repeated additions. For example, if $P \in E_p(a, b)$, then $5P$ is computed as $5P = P + P + P + P + P\ (\text{mod}\ p)$.

Definition 2 (Elliptic curve Computational Diffie-Hellman (EC-CDH) Assumption) The problem of computing $Q = qP$ and $R = rP$ are relatively easy for given scalar $q, r \in Z_p$ and an elliptic curve point $P \in E_p(a, b)$. However, given two points qP and rP, it is a computationally hard to derive qrP. This problem is called the elliptic curve Computational Diffie-Hellman [23].
This can be defined more formally by considering an Experiment $Exp_G^{cdh}(\mathcal{A})$ where we choose two values q and r in Z_p, compute qP and rP, and then provide qP and rP to $\mathcal{A}$. $\mathcal{A}$ experiment $Exp_G^{cdh}(\mathcal{A})$ outputs 1 if $Z = qrP$ and 0 otherwise. We defined $Adv_G^{cdh}(\mathcal{A}) = Pr[Exp_G^{cdh}(\mathcal{A}) = 1]$ as the advantage of adversary in violating the CDH assumption. The advantage function of the group, $Adv_G^{cdh}(t) = max_{\mathcal{A}}\{Adv_G^{cdh}(\mathcal{A})\}$ with time complexity at most t.

2.3.1 Security model

In order to show that proposed scheme withstand the known attacks to the authentication protocols, we use the method of provable security. The security proof is based on the model of ECC-based password authentication scheme [1, 8, 14, 19, 37].

Participants We consider a distributed system, which constitutes two disjoint sets: $\mathcal{U}$, the set of users and $\mathcal{S}$, the set of trusted servers which assumed to consist of single trusted server. In distinct executions of the proposed authentication protocol Π, the participants may have several instances called oracles. Π_E^i denotes the i-th instance of participant E in a session. Each instance Π_E^i has partner ID, session ID and a session key pid_E^i, sid_E^i and sk_E^i, respectively.

Long lived keys Each user $U_i \in \mathcal{U}$ holds a password PW_i and server S holds a vector $PW_S =< PW_i >_{U \in \mathcal{U}}$ with an entry for each client.

Adversary model The interaction between the protocol participants and an adversary $\mathcal{A}$ occurs only via oracle queries that models the adversary $\mathcal{A}$ capacities in the real attack. The several instances may be active at any given time in a concurrent model, for a given intended partner, only one active user instance is allowed and password is non-concurrent model. This enables an adversary to simulates a real attack on the protocol. Let Π^i_U defines the i-th instance of participant U_i and b be a bit selected uniformly at random, then possible oracle queries are as follows:

Execute(Π^i_U, Π^j_S) This query models passive attacks against the protocol where adversary $\mathcal{A}$ eavesdrops on honest execution between user instance Π^i_U and server instance Π^i_S. It prompts an execution of the protocol between the user's instances Π^i_U and server's instances Π^j_S that outputs the exchanged messages during honest protocol execution to $\mathcal{A}$.

Reveal(Π^i_U) This query captures the notion of known key security. The instance Π^i_U, upon receiving the query and if it has accepted, provides the session key, back to $\mathcal{A}$.

Send(Π^i_U, m) This query models simulate active attacks. This query sends a message m to an instance Π^i_U, enabling $\mathcal{A}$ for active attacks. On receiving m, the instance Π^i_U continues according to the protocol specification. The message output by Π^i_U, if any, is returned to $\mathcal{A}$.

Corrupt(Π^i_U) This query returns the long-lived key PW_i of the participant U_i to the adversary.

Test(Π^i_U) This query is used for determining whether the protocol achieves authenticated key exchange or not. If Π^i_U has accepted, then a random bit $b \in \{0, 1\}$ chosen by the oracle, $\mathcal{A}$ is given either the real session key if $b = 1$, otherwise, a random key drawn from the session-key space.

Notation We say that an instance Π^i_U is said to be *open* if a query Reveal(Π^i_U) has been made by adversary and *unopened* if it is not opened. We say that an instance Π^i_U has *accepted* if it goes into an accept mode after receiving the last expected protocol message.

Definition 3 Two instances Π^i_U and Π^i_S are said to be partnered if the following conditions hold:

(1) Both Π^i_U and Π^i_S accept;
(2) Both Π^i_U and Π^i_S share the same session identifications (sid);
(3) The partner identification for Π^i_U and Π^i_S and vice-versa.

Definition 4 We say an instance Π^i_U is considered fresh if the following conditions are met:

(i) It has accepted;
(ii) Both Π^i_U and its partner Π^i_S are unopened;
(iii) They are both instances of honest clients.

Definition 5 Consider an execution of the authentication protocol Π by an adversary $\mathcal{A}$, in which the latter is given access to the Execute, Send and Test oracles and asks at most single Test query to a fresh instance of an honest clints. Let b' be his output, if $b' = b$, where b is the hidden bit selected by the Test oracle. Let D be user's password dictionary with size $|D|$. Then, the advantage of $\mathcal{A}$ in violating the semantic security of the protocol Π is defined more precisely as follows:

$$Adv_{\Pi,D}(\mathcal{A}) = |2Pr[b' = b] - 1|$$

The password authentication protocol is semantically secure if the advantage $Adv_{\Pi,D}(\mathcal{A})$ is only negligibly larger than $O(q_s)/|D|$, where where q_s is the number of Send queries.

3 Proposed authenticated key exchange protocol for sip using ECC

In this section, we discuss our proposed authenticated key exchange scheme without using smart card. Our presented scheme consists of following four phases:

- System setup phase
- Registration phase
- Authentication and key agreement phase
- password change phase

3.1 System setup phase

The server selects an elliptic curve $E_p(a, b)$ over a finite field $\mathbb{Z}_p$, where p is a large prime. A base point P with order n over the elliptic curve $E_p(a, b)$ is selected. The secret keys $mk \in Z_p^*$ is also selected. Lastly, the server computes public key $P_{pub} = mkP$ and chooses a one way hash function $h : \{0, 1\}^* \rightarrow \{0, 1\}^n$, and then publishes the system parameters $\{E_p(a, b), p, P, h(\cdot), P_{pub}\}$ and keeps mk secret.

3.2 Registration phase

To become a new user in the system, a user resisters his/her username to the server. The registration phase is summarized in Fig. 1. The description of registration phase is given below:

Step 1. User $\longrightarrow$ Server: $< ID_i, RPW_i >$
The user U_i chooses uniformly distributed password PW_i and selects a random number $b \in Z_p^*$. U_i computes $RPW_i = h(ID_i||b||PW_i)$. U_i stores b in his device and submits $< ID_i, RPW_i >$ to S via secure channel.

Step 2. On receiving registration request with ID_i and RPW_i, S checks ID_i in its database. If ID_i exists, S asks a new username. Otherwise, S computes $Y_i = h(ID_i||mk) \oplus RPW_i$ and stores (ID_i, Y_i, Y_{old}) in its database, where Y_{old} is a null value.

3.3 Authentication and key agreement phase

A legal user with valid username and password can initiate login session. Then, the user and server verify the correctness of each other. If the mutual authentication holds, both user and server compute session key. The summary of phase is given in Fig. 2.

Step 1. User $\longrightarrow$ Server: REQUEST $< DID_i, C_i, V_i, T_i >$
The user U_i chooses a random number $u \in Z_p$ and computes $C_i = uP$ and $D_i = uP_{pub} = umkP = (D_i^x, D_i^y) \in E_P(a, b)$. U_i also computes $V_i = h(ID_i||T_i||h(ID_i||b||PW_i)||D_i^x)$, where T_i is the current timestamp. U_i masks ID_i

User (U_i)	Secure channel	Server (S)
Select b, ID_i and PW_i		
Compute $RPW_i = h(ID_i\|\|b\|\|PW_i)$		
Store b into device	$\xrightarrow{< ID_i, RPW_i >}$	Compute $Y_i = h(ID_i\|\|mk) \oplus RPW_i$
		Store (ID_i, Y_i, Y_{old})

Fig. 1 Mechanism of user registration

User (U_i)	Public channel	Server (S)
Select $u \in Z_p^*$		
Compute $C_i = uP$		
$D_i = uP_{pub} = (D_i^x, D_i^y) \in E_P(a, b)$		
$DID_i = ID_i \oplus D_i^y$		
$V_i = h(ID_i\|\|T_i\|\|h(ID_i\|\|b\|\|PW_i)\|\|D_i^x)$	$\xrightarrow{< DID_i, C_i, V_i, T_i >}$	Verify $T_i' - T_i \leqslant \Delta T$
		Compute $D_s = mkC_i = (D_s^x, D_s^y) \in E_P(a, b)$
		Retrieve $ID_i = DID_i \oplus D_s^y$
		$RPW_i = Y_i \oplus h(ID_i\|\|mk)$
		Verify $V_i \overset{?}{=} h(ID_i\|\|T_i\|\|RPW_i\|\|D_i^x)$
		Compute $C_s = sP$
		$Z_s = sC_i = suP$
		$sk_s = h(ID_i\|\|T_i\|\|RPW_i\|\|T_s\|\|D_s^x\|\|Z_s^x)$
Verify $T_s' - T_s \leqslant \Delta T$	$\xleftarrow{< C_s, V_s, T_s >}$	$V_s = h(ID_i\|\|T_i\|\|sk_s\|\|T_s\|\|Z_s^y)$
Compute $Z_i = uC_s$		
$sk_i = h(ID_i\|\|T_i\|\|h(ID_i\|\|b\|\|PW_i)\|\|T_s\|\|D_i^x\|\|Z_i^x)$		
Verify $V_s \overset{?}{=} h(ID_i\|\|T_i\|\|sk_i\|\|T_s\|\|Z_i^y)$		

Fig. 2 Authentication and key exchange mechanism, where user and server mutually authenticate each other and draw a common key

as $DID_i = ID_i \oplus D_i^y$. Finally, U_i sends the login message *REQUEST* $< DID_i, C_i, V_i, T_i >$ to S.

Step 2. Upon receiving the message $< DID_i, C_i, V_i, T_i >$ at time T_i', S first verifies $T_i' - T_i \leqslant \Delta T$. If verification holds, S computes $D_s = mkC_i = mkuP = (D_s^x, D_s^y)$, and then retrieves $ID_i = DID_i \oplus D_s^y$. S computes $h(ID_i||mk)$ and retrieves RPW_i as $RPW_i = Y_i \oplus h(ID_i||mk)$. S verifies $V_i \overset{?}{=} h(ID_i||T_i||RPW_i||D_i^x)$. If verification succeeds, U_i is authenticated by S.

Step 3. Server $\longrightarrow$ User: CHALLENGE$\{C_s, V_s, T_s\}$

S selects a random number $s \in Z_p^*$ and computes $C_s = sP$ and $Z_s = sC_i = suP$. S computes the session key $sk_s = h(ID_i||T_i||RPW_i||T_s||D_s^x||Z_s^x)$ and $V_s = h(ID_i||T_i||sk_s||T_s||Z_s^y)$, where T_s is the current timestamp used by server. S sends the challenge message CHALLENGE $< C_s, V_s, T_s >$ to U_i.

Step 4. On receiving the challenge message $< C_s, V_s, T_s >$ at time T_s', U_i verifies $T_i' - T_i \leqslant \Delta T$. If verification succeeds, U_i computes $Z_i = uC_s = usP$ and the session key $sk_i = h(ID_i||T_i||h(ID_i||b||PW_i)||T_s||D_i^x||Z_i^x)$. Then, S verifies $V_s \overset{?}{=} h(ID_i||T_i||sk_i||T_s||Z_i^y)$. If the verification succeeds, S's authentication and session key verification hold.

3.4 Password change phase

When a legal user U_i wants to change his/her password, he/she selects a new password and sends the password change request to the server using established session key of current authorized session. Upon receiving the password change request, the server verifies the validity of request. For valid request, server updates the password and response with accept message. Otherwise, server responds with reject message. If a user receive acceptance of new password, he update the password. Otherwise, user again sends the password update request. The password change phase is summarized in Fig. 3. The description of password update phase is given below:

Step 1. User $\longrightarrow$ Server: CHANGEPW$< DID_i, B, M_i >$

U_i selects a new password PW_{new} and a random number b^*. U_i computes $RPW_{new} = h(ID_i||b^*||PW_{new})$, $B = RPW_{new} \oplus h(sk_i||RPW_i)$ and $M_i =$

User (U_i)	Public channel	Server (S)
	$\xleftrightarrow{\texttt{Authorized session}}$	
Select b^* and PW_{new}		
Compute $RPW_{new} = h(ID_i \Vert b^* \Vert PW_{new})$		
$B = RPW_{new} \oplus h(sk_i \Vert RPW_i)$		
$M_i = h(RPW_{new} \Vert sk_i \Vert RPW_i)$	$\xrightarrow{\texttt{CHANGEPW} < DID_i, B, M_i >}$	Retrieve $RPW_{new} = B \oplus h(sk_s \Vert RPW_i)$
		Verify $M_i \overset{?}{=} h(RPW_{new} \Vert sk_s \Vert RPW_i)$
		If verification holds,
		Compute $Y_{new} = h(ID_i \Vert mk) \oplus RPW_{new}$
		Update (ID_i, Y_i, Y_{old}) with (ID_i, Y_{new}, Y_i)
Verify	$\xleftarrow{\texttt{ACCEPT} < M_s >}$	Compute $M_s = h(ID_i \Vert RPW_{new} \Vert sk_s \Vert RPW_i)$
$M_s \overset{?}{=} h(ID_i \Vert RPW_{new} \Vert sk_i \Vert RPW_i)$		Otherwise,use Y_{old}
Update b with b^*		Verify $M_i \overset{?}{=} h(RPW_{new} \Vert sk_s \Vert RPW_{old})$
		If verification holds,
		Compute $M'_s = h(ID_i \Vert RPW_{new} \Vert sk_s \Vert RPW_{old})$
Verify	$\xleftarrow{\texttt{ACCEPT} < M'_s >}$	Update (ID_i, Y_i, Y_{old}) with (ID_i, Y_{new}, Y_{old})
$M'_s \overset{?}{=} h(ID_i \Vert RPW_{new} \Vert sk_i \Vert RPW_{old})$		Otherwise, session is terminated
Update b with b^*	$\xleftarrow{\texttt{REJECT} < M''_s >}$	Compute $M''_s = h(ID_i \Vert RPW_{new} \Vert sk_s)$

Fig. 3 Summary of password change phase

$h(RPW_{new}||sk_i||RPW_i)$. Then, U_i sends the password change request message CHANGEPW$< DID_i, B, M_i >$ to S.

Step 2. Server ⟶ User: ACCEPT$< M_s >$

Upon receiving request with message CHANGEPW$< DID_i, B, M_i >$, S retrieves $RPW_{new} = B \oplus h(sk_s||RPW_i)$, and then verifies $M_i \overset{?}{=} h(RPW_{new}||sk_s||RPW_i)$. If verification holds, S accepts the request and updates (ID_i, Y_i, Y_{old}) with (ID_i, Y_{new}, Y_i), where $Y_{new} = h(ID_i||mk) \oplus RPW_{new}$. S sends the message ACCEPT$<$ $M_s >$ to U_i, where $M_s = h(ID_i||RPW_{new}||sk_s||RPW_i)$.

Step 3. Server ⟶ User: ACCEPT$< M'_s >$

If verification does not hold, S retrieves $RPW_{old} = B \oplus h(sk_s||RPW_{old})$ using old password, where $RPW_{old} = Y_{old} \oplus h(ID_i||mk)$. S verifies $M_i \overset{?}{=} h(RPW_{new}||sk_s||RPW_{old})$. If verification holds, S accepts the request and updates (ID_i, Y_i, Y_{old}) with (ID_i, Y_{new}, Y_{old}). Then, S sends the message ACCEPT$< M'_s >$ to U_i, where $M'_s = h(ID_i||RPW_{new}||sk_s||RPW_{old})$.

Step 4. Server ⟶ User: Reject$< M''_s >$

If verification fails in Step 2 & Step 3, S rejects the password update request with the message Reject$< M''_s >$, where $M''_s = h(ID_i||RPW_{new}||sk_s)$.

Step 5. On receiving the response message, U_i can verify the correctness of response. If server rejects the request or U_i does not receive server's response, U_i again initiates the password update phase as discussed in Step 1. Otherwise, U_i can verify the response of the server as follows:

– On receiving the message ACCEPT$< M_s >$, U_i can verify $M_s \overset{?}{=} h(ID_i||RPW_{new}||sk_i||RPW_i)$ using current password. If verification succeeds, U_i replaces b with b^*.

– On receiving the message ACCEPT$< M'_s >$, U_i verifies $M_s \overset{?}{=} h(ID_i||RPW_{new}||sk_i||RPW_{old})$. The user receives $< M'_s >$ only if last session of password update failed at user side. In this case also, a user can identify the correctness of server's response and replace b with b^*.

– On receiving the message REJECT$< M''_s >$, U_i verifies $M''_s \overset{?}{=} h(ID_i||RPW_{new}||sk_s)$. Then, U_i does not update b.

Remark 1 The server keeps the backup of old password to avoid DOS attack. In case, if an adversary intercept the server's response in password update, the server will update the password, but not the user. However, the user can again sends the password update request, which correctness a server can identify using the backup password.

4 Proof of mutual authentication using BAN Logic

We apply logical postulates of BAN logic [7, 30] to show the correctness of mutual authentication between the remote user and server. Using BAN logic, we show that the user and server determine whether exchanged information is fresh and trustworthy against eavesdropping. It comprises the verification of message origin, message freshness and the origin's trustworthiness. In the proposed scheme, the generic form of the messages exchange between the user and server are as follows:

Message 1. $U_i \rightarrow S : \langle ID_i \oplus D_i^y, h(ID_i||T_i||RPW_i||D_i^x), uP, T_i\rangle$
Message 2. $S \rightarrow U_i : \langle h(ID_i||T_i||sk_s||T_s||Z_s^y), sP, T_s\rangle$

Subsequently, we translate the message 1 & 2 into idealize form as follows:
Message 1. $U \rightarrow S :< ID_i >_{umkP}, (ID_i, umP, T_i)_{RPW_i}, uP, T_i$

Message 2. $S \rightarrow U : (U_i \overset{sk}{\longleftrightarrow} S, ID_i, T_i, T_s)_{suP}, T_s$
Recall that in the proposed scheme, the user and server use fresh timestamp. We make the following assumptions about the initial state of the proposed scheme:

A_0: $U_i|\equiv \sharp(u)$;
A_1: $U_i|\equiv \sharp(T_i)$;
A_2: $S|\equiv \sharp(T_s)$;
A_3: $U_i|\equiv (U_i \overset{RPW_i}{\longleftrightarrow} S)$;
A_4: $S|\equiv (U_i \overset{RPW_i}{\longleftrightarrow} S)$;
A_5: $U_i|\equiv S|\equiv (U_i \overset{RPW_i}{\longleftrightarrow} S)$;
A_6: $S|\equiv U_i|\equiv (U_i \overset{RPW_i}{\longleftrightarrow} S)$;
A_7: $U_i|\equiv (U_i \overset{umkP}{\longleftrightarrow} S)$;
A_8: $S|\equiv (U_i \overset{umkP}{\longleftrightarrow} S)$;

Lemma 1 *The server can verify the freshness and authenticity of user's message.*

Proof User generates a login message and sends to the server in order to login to the server. With the message, the server receives the timestamp with other values which help to prove the correctness of message source as follows:

S_1: According to the message 1, we could get: $S \lhd (ID_i, umkP, T_i)_{RPW_i}, uP, T_i$.
S_2: According to the assumption A_4, we apply the message meaning rule to get: $S|\equiv U_i|\sim T_i$.
S_3: According to the assumption A_1, we apply the freshness-propagation rule to get: $S|\equiv \sharp(ID_i, umkP, T_i)_{RPW_i}$.
S_4: According to the A_8 and S_3, we apply nonce verification rule to obtain: $S|\equiv U_i|\equiv (ID_i, umkP, T_i)_{RPW_i}$.

S_5: According to the assumption A_4 and S_4, we apply the jurisdiction rule to get: $S|\equiv T_i$. The server can identify freshness of user's message using S_5 and authenticity using S_4. □

Lemma 2 *The user can verify the freshness and authenticity of server's response.*

Proof In the proposed scheme, when correctness of user's login message holds, the server responds with a message which includes the server's timestamp. The user can be able to identify the authenticity of server's message as follows:

S_6: According to the message 2, we could obtain: $U_i \lhd (U_i \overset{sk}{\longleftrightarrow} S, ID_i, T_i, T_s)_{suP}, T_s$.
S_7: According to the assumption A_3, we apply the message meaning rule to get: $U_i|\equiv S|\sim T_s$.
S_8: According to the assumption A_1 and S_7, we apply the freshness conjuncatenation rule to get: $U_i|\equiv \sharp(U_i \overset{sk}{\longleftrightarrow} S, ID_i, T_i, T_s)_{suP}$.
S_9: To compute the session key $sk(= h(ID_i||T_i||RPW_i||T_s||D_s^x||Z_s^x))$, the shared secret value RPW_i and $umkP$ are needed to get: $U_i|\equiv \sharp(ID_i, T_i, T_s, muP, suP)_{RPW_i}$.
S_{10}: According to the A_7, S_8 and S_9, we apply nonce verification rule to obtain: $U_i|\equiv S|\equiv (ID_i, T_i, T_s, muP, suP)_{RPW_i}$.
S_{11}: According to the assumption A_3, A_7 and S_{10}, we apply the jurisdiction rule to get: $U_i|\equiv T_s$.
This shows that the user can verify the freshness and authenticity of server's message with S_{10} and S_{11}. □

Theorem 1 *The user and server can mutually authenticate each other.*

Proof According to the Lemma 1, the server can identify the fastnesses of message. Then, using A_6 and A_8, we apply the BAN logic rule to get $S|\equiv U_i|\equiv (S \overset{sk}{\longleftrightarrow} U_i)$.

According to the Lemma 2, the user can identify the freshness of server's response and authenticity with A_5 and A_7. Then, we apply the BAN logic rule to get $U_i|\equiv S|\equiv (U_i \overset{sk}{\longleftrightarrow} S)$. □

5 Security analysis

5.1 Formal security analysis of the proposed scheme

Theorem 2 *let D be a uniformly distributed dictionary of possible passwords with size $|D|$, Let Π be the improved authentication protocol described in Algorithm 1 & 2. Let $\mathcal{A}$ be an adversary against the semantic security within a time bound t. Suppose that CDH assumption holds, then,*

$$Adv_{\Pi,D}(\mathcal{A}) = \frac{2q_h^2}{p} + \frac{2q_s}{p} + \frac{(q_s+q_e)^2}{p} + 2q_h Adv_G^{cdh}(\mathcal{A}) + \frac{2q_h}{p} + \frac{2q_s^2}{D}$$

where $Adv_G^{cdh}(\mathcal{A})$ is the success probability of $\mathcal{A}$ of solving the elliptic curve based computational DiffieHellman problem. q_s is the number of Send queries, q_e is the number of Execute queries, q_h is the number of random oracle queries and p is a n-bit prime number, where n is security parameter.

Proof This proof defines a sequence of hybrid games, starting at the real attack and ending up in game where the adversary has no advantage. For each game $G_i (0 \leq i \leq 5)$, we define an event $Succ_i$ corresponding to the event in which the adversary correctly guesses the bit b in the test-query.

Game G_0 This game correspond to the real attack in the random oracle model. In this game, all the instances of U_i and the server S are modeled as the real execution in the random oracle. By definition of event $Succ_i$ in which the adversary correctly guesses the bit b involved in the Test-query, we have

$$Adv_{\Pi,D}(\mathcal{A}) = 2|Pr[Succ_0] - \frac{1}{2}| \tag{5}$$

Game G_1 This game is identical to the game G_0, except that we simulate the hash oracles h by maintaining the hash lists $List_h$ with entries of the form (Inp, Out). On hash query for which there exists a record (Inp, Out) in the hash list, return Out. Otherwise, randomly choose a number $Out \in Z_p^*$, send it to $\mathcal{A}$ and store the new tuple (Inp, Out) into the hash list. The Execute, Reveal, Send, Corrupt and Test oracles are also simulated as in the real attack where the simulation of the different polynomial number of queries asked by $\mathcal{A}$. From the viewpoint of $\mathcal{A}$, we identify that the game is perfectly indistinguishable from the real attack. Thus, we have

$$Pr[Succ_1] = Pr[Succ_0] \tag{6}$$

Game G_2 In this game, the simulation of all the oracles is identical to game G_1 except that the game is terminated if the collision occurs in the simulation of the transcripts $< DID_i, C_i, V_i, T_i >$ and $< C_s, V_s, T_s >$. According to the birthday paradox, the probability of collisions of the simulation of hash oracles is at most $\frac{q_h^2}{2p}$. Similarly, the probability of collisions in the transcripts simulations is at most $\frac{(q_h+q_e)^2}{2p}$. Since C_i was selected uniformly at random. Thus, we have

$$|Pr[Succ_2] - Pr[Succ_1]| = \frac{q_h^2}{2p} + \frac{(q_s + q_e)^2}{2p} \tag{7}$$

Game G_3 The simulation of this game is similar to the previous game except the game will be aborted if $\mathcal{A}$ can correctly guessed the authentication values V_i and V_s without asking oracle h. This game and earlier game are indistinguishable unless the instances Π_U^i and Π_S^i rejects a valid authentication value. Hence, we have

$$|Pr[Succ_3] - Pr[Succ_2]| \leq \frac{q_h}{p} \tag{8}$$

Game G_4 In this game, the session key is guessed without asking the corresponding oracle h so that it become independent of password and ephemeral keys $umkP$. We change the way with earlier game unless $\mathcal{A}$ queries h on the common value $(ID_i||T_i||RPW_i||T_s||umkP||usP)$. Thus, $Adv_G^{cdh}(\mathcal{A}) \geq \frac{1}{q_h}|Pr[Succ_4] - Pr[Succ_3]| - \frac{1}{p}$, that is, the difference between the game G_4 and the game G_3 is as follows:

$$|Pr[Succ_4] - Pr[Succ_3]| \leq q_h Adv_G^{cdh}(\mathcal{A}) + \frac{q_h}{p} \tag{9}$$

Game G_5 This game is similar to the game G_4 except that in Test query, the game is aborted if $\mathcal{A}$ asks a hash function query with $(ID_i||T_i||RPW_i||T_s||umkP||usP)$.

$\mathcal{A}$ gets the session key sk by hash function query with probability at most $\frac{q_h^2}{2p}$. Hence, we have

$$|Pr[Succ_5] - Pr[Succ_4]| \leq \frac{q_h^2}{2p} \tag{10}$$

If $\mathcal{A}$ does not make any h query with the correct input, it will not have any advantage in distinguishing the real session key from the random once. Moreover, if the corrupt query Corrupt(U, 2) is made that means the password-corrupt query Corrupt(U, 1) is not made. Thus, the probability of $\mathcal{A}$ made off-line password guessing attack is $\frac{q_s^2}{D}$. Combining the Eqs. 6, 7, 8, 9, 10 and 11 one gets the announced result as:

$$Adv_{\Pi,D}(\mathcal{A}) = \frac{2q_h^2}{p} + \frac{2q_s}{p} + \frac{(q_s + q_e)^2}{p} + 2q_h Adv_G^{cdh}(\mathcal{A}) + \frac{2q_h}{p} + \frac{2q_s^2}{D}$$

□

Algorithm 1 Simulation of send query

```
On the query $Send(\Pi_U^i, start)$, assume that $U_i$ is in correct state, then we proceed as follows:
Choose a number $u \in_R Z_p^*$, compute $D_i = uP_{pub}$, $DID_i = ID_i \oplus D_i^y$, $C_i = uP$, $V_i = h(ID_i||T_i||h(ID_i||b||PW_i)||D_i^x)$. This query returns $\langle DID_i, C_i, V_i, T_i \rangle$ as answer.
On a query $Send(S, < DID_i, C_i, V_i, T_i >)$, assume that $S$ is in correct state, we continue as follows.
if $T_i' - T_i > \Delta T$ then
    Reject the message.
else Compute $D_s = mC_i$, $ID_i = DID_i \oplus D_s^y$, $RPW_i = Y_i \oplus h(ID_i||mk)$ and $V_i^* = h(ID_i||T_i||RPW_i||D_s^x)$
    if $V_i^* \neq V_i$ then
        Reject the request.
    else Compute $C_s = sP$, $Z_s = sC_i$, $sk_s = h(ID_i||T_i||RPW_i||T_s||D_s^x||Z_s^x)$ and $V_s = h(ID_i||T_i||sk_s||T_s||Z_s^y)$. The query returns $< C_s, V_s, T_s >$ as answer.
    end if
end if
On a $Send < C_s, V_s, T_s >$, assume that $U_i$ is in correct state, we continue as follows:
if then $T_s' - T_s > \Delta T$
    Reject the message $< C_s, V_s, T_s >$.
else Compute $Z_i = uC_s$, $sk_s = h(ID_i||T_i||h(ID_i||b||PW_i)||T_s||D_s^x||Z_i^x)$ and $V_s^* = h(ID_i||T_i||sk_i||T_s||Z_i^y)$
    if $V_s^* \neq V_s$ then
        Reject the response.
    else The user instance accepts.
    end if
end if
```

Algorithm 2 Simulation of Execute query

```
On a query $Reveal(\Pi_U^i)$, we proceed as follows:
if The instance $\Pi_U^i$ is accepted then
    This query answered the session key.
end if
```

5.2 Further security discussion of the proposed scheme

In this section, we discuss that the proposed scheme have all the security feature of authentication and key agreement protocols including user's anonymity.

Proposition 1 *The proposed scheme could provide user's anonymity with unlinkability.*

Proof The login message $\{DID_i, C_i, V_i, T_i\}$ includes DID_i instead of ID_i. To retrieve ID_i from DID_i is equivalent to compute $umkP$ using uP and mkP as $DID_i = ID_i \oplus umkP^y$. Computation of $umkP$ using uP and mkP is equivalent to elliptic curve computational DiffieHellman (EC-CDH) problem. As EC-CDH is considered to be a computationally hard problem (defined in Definition 2), $\mathcal{A}$ cannot retrieve ID_i from DID_i. Moreover, user randomly chooses value u for each session, which makes $umkP$ different for each session so as DID_i. Additionally, no information is repeated in consecutive communications. This shows that our scheme also achieve unlinkability property. □

Proposition 2 *The proposed scheme could withstand privileged-insider attack.*

Proof During the registration phase, a legal user U_i submits masked password RPW_i to the server S instead of original password PW_i, where $RPW_i = h(ID_i||b||PW_i)$ for randomly selected value b. Thus, an insider cannot achieve the password PW_i due to the non-retrieval property of the one-way hash function $h(\cdot)$. Moreover, the insider cannot guess the password as user does not submit b to the server. This shows that the proposed scheme resists insider attack. □

Proposition 3 *The proposed scheme could resist stolen verifier attack.*

Proof In proposed scheme, the server stored Y_i, where $Y_i = h(ID_i||mk) \oplus RPW_i$. Thus, to retrieve, RPW_i from Y_i, the adversary needs user's identity ID_i and server's secret key mk. It is noted that neither the smart card nor the transmitted messages includes user's identity ID_i in the proposed scheme. Additionally, the server key is consider secret. As the server secret key is only known to the server, the adversary cannot achieve RPW_i. This shows that our proposed scheme withstands the stolen verifier attack. □

Proposition 4 *The proposed scheme could resist off-line password guessing attack.*

Proof In this attack, an adversary may try to guess a legal user U_i's password PW_i using the transmitted messages. In proposed scheme, the adversary may try to verify the password using the condition $V_i = h(ID_i||T_i||h(ID_i||b||PW_i)||umkP^x)$ or $V_s = h(ID_i||T_i||sk_s||T_s||usP^x)$, where $sk_s = h(ID_i||T_i||h(ID_i||b||PW_i)||T_s||D_s^x||Z_s^x)$. However, this attempt cannot succeed in the proposed scheme which is justified below:

– To verify the guessed password PW_i^* using $V_i = h(ID_i||T_i||h(ID_i||b||PW_i)||umkP^x)$, $\mathcal{A}$ has to compute uP_{pub}. To compute uP_{pub} using uP and P_{pub}, is equivalent to solve EC-CDH problem.
– To verify the guessed password PW_i^* using $V_s = h(ID_i||T_i||sk_s||T_s||usP^x)$, $\mathcal{A}$ has to compute suP. The computation of suP using uP and sP, is equivalent to solve EC-CDH problem.

It is clear from the above discussion that guessing password in the proposed scheme is equivalent to solve EC-CDH problem, which is hard. □

Proposition 5 *The proposed scheme could withstand replay and man-in-the-middle attacks.*

Proof The login and verification messages include the timestamp. The maximum transmission delay ΔT is in communication, does not allow to repeat the old transmitted message. To update the timestamp of message $\{DID_i, C_i, V_i, T_i\}$ with $\{DID_i, C_{\mathcal{A}}, V_{\mathcal{A}}, T_{\mathcal{A}}\}$ for current timestamp $T_{\mathcal{A}}$, $\mathcal{A}$ has to compute $V_{\mathcal{A}}$ which requires the user U_i's password PW_i and identity ID_i as $V_{\mathcal{A}} = h(ID_i||T_{\mathcal{A}}||h(ID_i||b||PW_i)||aP_{pub}^x)$ for timestamp $T_{\mathcal{A}}$

and random value a. Since the user's password PW_i is secret, $\mathcal{A}$ cannot achieve it. Moreover, to replace $\{sP, V_s, T_s\}$ with $\{aP, V_s^*, T_E\}$, $\mathcal{A}$ has to compute $v = h(ID_i||T_i||sk_s||T_s||auP^y)$, which also requires RPW_i and ID_i. As only valid user know ID_i and RPW_i, our proposed scheme resists the replay and man-in-the-middle attacks. □

Proposition 6 *The proposed scheme could resist user impersonation attack.*

Proof In such an attack, an adversary may try to masquerade as a legitimate user U_i to successfully login to the server S. However, our proposed scheme resists this attack.

- $\mathcal{A}$ may try to login to the server S using the replay attack. However, the proposed scheme resists the replay attack.
- $\mathcal{A}$ may try to generate a valid login message $\{DID_{\mathcal{A}}, aP, V_{\mathcal{A}}, T_{\mathcal{A}}\}$ for a random value a and current timestamp $T_{\mathcal{A}}$, where $V_{\mathcal{A}} = h(ID_i||T_{\mathcal{A}}||h(ID_i||b||PW_i)||aP_{pub}^x)$. However, the adversary cannot compute $V_{\mathcal{A}}$ as computation of $V_{\mathcal{A}}$ requires PW_i and ID_i.

It is clear that the adversary cannot generate valid login message as PW_i and ID_i are only known to user. This shows that the proposed scheme resists user impersonation attack. □

Proposition 7 *The proposed scheme could withstand server impersonation attack.*

Proof In this attack, an adversary can masquerade as the server S and try to respond with a valid message to the user U_i. When a user U_i sends a login message $\{DID_i, u'P, V_i', T_i'\}$ to the server S, the adversary intercepts this message and try to respond with a valid message, where $V_i' = h(ID_i||T_i'||h(ID_i||b||PW_i)||u'mkP^x)$. However, the proposed scheme resist this attack as follows:

- $\mathcal{A}$ may try to respond using the old transmitted message $< C_s, V_s, T_s >$ of S. This attempt cannot succeed as the login and response message includes timestamp and proposed scheme resists replay attack.
- $\mathcal{A}$ may try to generate a valid response message $< aP, V_s', T_{\mathcal{A}} >$ for current timestamp $T_{\mathcal{A}}$, where computation of $V_s' = h(ID_i||T_i'||sk_s'||T_E||au'P^x)$ and $sk_s = h(ID_i||T_i'||RPW_i||T_{\mathcal{A}}||u'mkP^x||au'P^x)$ require RPW_i and ID_i.

This shows that our proposed scheme has the ability to resist the server impersonation attack. □

Proposition 8 *The proposed scheme could support mutual authentication.*

Proof In our scheme, the server S verifies the authenticity of user U_i's request by checking the condition $V_i = h(ID_i||T_i||h(ID_i||b||PW_i)||D_i^x)$ during the authentication phase. To compute V_i, U_i's ID_i and PW_i are needed. Therefore, $\mathcal{A}$ cannot forge. Additionally, V_i includes timestamp, the adversary cannot replay the old message. This shows that the server S can correctly verify the message source. U_i also verifies the authenticity of the server S with the condition $V_s = h(ID_i||T_i||sk_s||T_s||Z_s^y)$, which also requires ID_i and PW_i as $sk_s = h(ID_i||T_i||RPW_i||T_s||D_s^x||Z_s^x)$ and $RPW_i = h(ID_i||b||PW_i)$. This shows that the user U_i can also correctly verify the server S challenge. Hence, mutual authentication between U_i and S can successfully achieved in our scheme. □

Proposition 9 *The proposed scheme could have Key freshness property.*

Proof Note that in our scheme, each established session key $h(ID_i||T_i||RPW_i||T_s||umkP^x||suP^x)$ includes timestamp T_i and T_s and random values u and s. The timestamp are used to achieve the freshness for each session. Uniqueness property of timestamp for each session, guaranties the unique key for each session. The unique key construction for each session shows that the proposed scheme supports the key freshness property. □

Proposition 10 *The proposed scheme could have known key secrecy property.*

Proof In our scheme, if a previously established session key $h(ID_i||T_i||RPW_i||T_s||umkP^x||suP^x)$ is compromised, the compromised session key reveals no information about other session keys due to following reasons:

- Each session key is hashed with one-way hash function. Therefore, no information can be retrieved from the session key.
- Each session key includes timestamp, which ensures different key for each session.

Since no information about other established session keys from the compromised session key is extracted, our proposed scheme achieves the known key secrecy property. □

Proposition 11 *The proposed scheme could have forward secrecy and perfect forward secrecy.*

Proof Forward secrecy states that compromise of user's secret value does not lead to compromise of the established session keys. The perfect forward secrecy states that using compromised secret key of server, an adversary cannot compute established session keys. In the proposed scheme, using user's secret value PW_i and server's secret key mk, an adversary $\mathcal{A}$ cannot compute the session key due to the following fact:

- To compute the session key sk, user identity ID_i, usP and $umkP$ are needed along with PW_i as session key $sk = h(ID_i||T_i||h(ID_i||b||PW_i)||T_s||D_s^x||Z_s^x)$, where $Z_s = mkuP = (D_s^x, D_s^y)$ and $D_s = mkuP = (D_s^x, D_s^y)$,.
- Neither the smart card nor transmitted messages includes ID_i. The transmitted message include DID_i. Using PW_i, $\mathcal{A}$ cannot derive ID_i, but using mk, $\mathcal{A}$ can get ID_i.
- The computation of $umkP$ using uP and mkP is equivalent to solve EC-CDH problem. But, using mk, $\mathcal{A}$ can compute $umkP$.
- To compute usP using uP and sP is also equivalent to solve EC-CDH problem. Moreover, $\mathcal{A}$ cannot achieve u from uP and s from sP using mk and PW_i.

As computation of usP is also equivalent to solve EC-CDH problem using uP and sP. This shows that our scheme preserves forward secrecy and perfect forward secrecy. □

6 Discussion

In this section we compare the performance of the proposed scheme with some recently proposed authentication schemes for SIP using ECC, namely, Zhang et al.'s protocol [43],

Table 3 Computational overhead comparison of the proposed scheme with related schemes

Protocols/Overhead	User side	Server side	Total computation
Zhang et al.'s protocol [43]	$6T_h + 4T_{ecm} + 1T_{eca}$	$5T_h + 4T_{ecm} + 1T_{eca} + 1T_{inv}$	$11T_h + 8T_{ecm} + 2T_{eca} + 1T_{inv}$
Tu et al.'s protocol [33]	$5T_h + 4T_{ecm} + 1T_{eca}$	$5T_h + 3T_{ecm}$	$10T_h + 7T_{ecm} + 1T_{eca}$
Farash's protocol [10]	$5T_h + 4T_{ecm} + 1T_{eca}$	$5T_h + 3T_{ecm}$	$10T_h + 7T_{ecm} + 1T_{eca}$
Yeh et al.'s protocol [39]	$8T_h + 4T_{ecm} + 2T_{eca}$	$5T_h + 4T_{ecm} + 2T_{eca}$	$13T_h + 8T_{ecm} + 4T_{eca}$
Jiang et al.'s protocol [20]	$5T_h + 4T_{ecm} + 1T_{eca}$	$4T_h + 4T_{ecm} + 1T_{eca}$	$9T_h + 8T_{ecm} + 2T_{eca}$
Zhang et al.'s protocol [43]	$4T_h + 3T_{ecm}$	$5T_h + 3T_{ecm}$	$9T_h + 6T_{ecm}$
Proposed scheme	$4T_h + 3T_{ecm}$	$4T_h + 3T_{ecm}$	$8T_h + 6T_{ecm}$

Tu et al.' protocol [33], Farash's protocol [10], Zhang et al.'s protocol [44], Jiang et al. [20] and Yeh et al.'s protocol [39].

In Table 3, we compare the computational overhead of our scheme with Zhang et al.'s protocol [43], Tu et al.' protocol [33], Farash's protocol [10], Zhang et al.'s protocol [44], Jiang et al. [20] and Yeh et al.'s protocol [39]. For calculate the computational overhead, we use the following notations: T_{ecm}: time complexity of executing an elliptic curve point multiplication operation; T_{eca}: time complexity of executing an elliptic curve point addition; T_{inv}: time complexity of executing modular inversion; T_h: time complexity of executing a one-way hash function. From Fig. 2, we see that our scheme requires computation cost from user's side and server's side are $4T_h + 3T_{ecm}$ and $4T_h + 3T_{ecm}$, respectively. It is clear from Table 3 that our proposed scheme is more efficient as compared to Zhang et al.'s protocol [43], Tu et al.' protocol [33], Farash's protocol [10], Zhang et al.'s protocol [44], Jiang et al. [20] and Yeh et al.'s protocol [39].

In Table 4, we have compared the communication overhead of the proposed scheme with other schemes, namely, Zhang et al.'s protocol [43], Tu et al.' protocol [33], Farash's protocol [10], Zhang et al.'s protocol [44], Jiang et al. [20] and Yeh et al.'s protocol [39]. For the login phase and authentication phase. We assume that the hash digest (output) is 160 bits, if we use SHA-1 hash function [29], timestamp is 32 bits, $username$ is 160 bits, realm identity $realm$ is 32 bits and random nonce/number is 160 bits. We take 160-bit ECC cryptosystem, because its security is same as 1024-bit RSA cryptosystem. Thus, for an elliptic curve $E_p(a, b)$, each parameter p, a and b requires 160 bits. Then, a point $P = (x_P, y_P) \in E_p(a, b)$ requires $(160 + 160) = 320$ bits. In our scheme, the REQUEST message $\{DID_U, C_i, V_i, T_i\}$ requires $(160+160+320+32) = 672$ bits, the CHALLENGE message $\{C_s, V_s, T_s\}$ requires $(320 + 160 + 32) = 512$ bits. As a result, our scheme needs $(672 + 512) = 1184$ bits for the communication overhead of two transmitted messages. From Table 4, it is clear that our scheme requires less communication overhead from Zhang et al.'s protocol [43], Tu et al.' protocol [33], Farash's protocol [10], Jiang et al. [20] and Yeh et al.'s protocol [39] and it is equally efficient to Zhang et al.'s protocol [44].

Finally, in Table 5, we have summarized the comparison of security features provided by the proposed scheme and other schemes, where symbol 'Yes' used if the protocol support the attribute, otherwise, 'No' is used. From this table, it is clear that the proposed scheme provides better security features. The proposed scheme has the ability to supports other good features such as user's anonymity and formal verification. The scheme is superior in terms of features as compared to relevant SIP authentication schemes: Zhang et al.'s protocol [43], Tu et al.' protocol [33], Farash's protocol [10], Zhang et al.'s protocol [44], Jiang et al. [20] and Yeh et al.'s protocol [39].

Table 4 Communication overhead comparison between our scheme and recently proposed SIP schemes

Scheme	Communication overhead
Zhang et al.'s protocol [43]	3 messages (1664 bits)
Tu et al.'s protocol [33]	3 messages (1664 bits)
Farash's protocol [10]	3 messages (1504 bits)
Yeh et al.'s protocol [39]	3 messages (1888 bits)
Jiang et al. protocol [20]	3 messages (1664 bits)
Zhang et al.'s protocol [43]	3 messages (1184 bits)
Proposed Scheme	2 messages (1184 bits)

Table 5 Features comparison between our scheme and other schemes

Security features	[43]	[33]	[10]	[20]	[39]	[44]	Proposed scheme
User anonymity	No	No	No	No	No	Yes	Yes
Insider attack	Yes	Yes	Yes	Yes	Yes	No	Yes
Off-line password guessing attack	Yes	Yes	Yes	Yes	No	Yes	Yes
Stolen smart card attack	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Denial-of-service attack	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Known session keys attack	Yes	Yes	Yes	Yes	Yes	Yes	Yes
User impersonation attack	No	No	Yes	Yes	Yes	Yes	Yes
Server impersonation attack	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Man-in-the middle attack	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Replay attack	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Mutual authentication	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Session key agreement	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Forward secrecy	Yes	Yes	Yes	Yes	No	Yes	Yes

7 Conclusion

We have discussed the merits and demerits of the existing authentication schemes for SIP in the literature. The analysis indicates that the existing schemes are failing to resist various attacks or does not protect user anonymity. In this work, we have presented a password based authenticated key agreement scheme for SIP without using smart card. The proposed scheme achieves user anonymity without imposing extra computation overhead. The proposed scheme also requires less computation and computation overhead compare to other related authentication scheme for SIP using elliptic curve cryptography. It supports mutual authentication and session key agreement where the user and server can correctly identify the legitimacy of each other and can draw a common key. Our scheme satisfies all desirable security attributes which are demonstrated in the security analysis through both informal and formal security analysis. Considering the security and efficiency, the proposed scheme provides strong authentication with anonymity for SIP.

Conflict of interests The author declares that he has no conflict of interest.

References

1. Abdalla M, Pointcheval D (2005) Interactive diffie-hellman assumptions with applications to password-based authentication. In: Financial Cryptography and Data Security. Springer, pp 341–356
2. Arkko J, Torvinen V, Camarillo G, Niemi A, Haukka T (2003) Security mechanism agreement for sip sessions, draft-ietfsip-sec-agree-04. txt
3. Arshad R, Ikram N (2013) Elliptic curve cryptography based mutual authentication scheme for session initiation protocol. Multimed Tools Appl 66(2):165–178
4. Arshad H, Nikooghadam M (2014) An efficient and secure authentication and key agreement scheme for session initiation protocol using ecc, Multimedia Tools and Applications, pp 1–17. doi:10.1007/s11042-014-2282-x
5. Bellare M, Canetti R, Krawczyk H (1996) Keying hash functions for message authentication. In: Advances in Cryptology (CRYPTO'96). Springer, pp 1–15

6. Boyd C, Mao W (1994) On a limitation of ban logic. In: Advances in CryptologyEUROCRYPT93. Springer, pp 240–247
7. Burrows M, Abadi M, Needham RM (1989) A logic of authentication, Proceedings of the Royal Society of London. A Math Phys Sci 426(1871):233–271
8. Dolev D, Yao AC (1983) On the security of public key protocols. IEEE Trans Inf Theory 29(2):198–208
9. Durlanik A, Sogukpinar I (2005) Sip authentication scheme using ecdh. World Enformatika Socity Transations on Engineering Computing and Technology 8:350–353
10. Farash M (2014) Security analysis and enhancements of an improved authentication for session initiation protocol with provable security, Peer-to-Peer Networking and Applications, pp 1–10. doi:10.1007/s12083-014-0315-x
11. Farash M, Attari M (2014) A provably secure and efficient authentication scheme for access control in mobile pay-tv systems. Multimed Tools Appl:1–20. doi:10.1007/s11042-014-2296-4
12. Farash MS, Attari MA (2013) An enhanced authenticated key agreement for session initiation protocol. Inf Technol Control 42(4):333–342
13. Gokhroo M, Jaidhar C, Tomar A (2011) Cryptanalysis of sip secure and efficient authentication scheme. In: IEEE 3rd International Conference on Communication Software and Networks (ICCSN-2011). IEEE, pp 308–310
14. He D, Chen J, Hu J (2012) An ID-based client authentication with key agreement protocol for mobile client–server environment on ECC with provable security. Inf Fusion 13(3):223–230
15. He D, Chen J, Chen Y (2012) A secure mutual authentication scheme for session initiation protocol using elliptic curve cryptography. Secur Commun Netw 5(12):1423–1429
16. He D, Kumar N, Chen J, Lee Cc, Ilamkurti NC, Yeo SS (2013) Robust anonymous authentication protocol for health-care applications using wireless. Med Sensor Netw 21(1):49–60
17. Huang H-F, Wei W-C (2006) A new efficient authentication scheme for session initiation protocol. Computing 1(2):1–3
18. Irshad A, Sher M, Rehman E, Ch S, Hassan M, Ghani A (2013) A single round-trip sip authentication scheme for voice over internet protocol using smart card, Multimedia Tools and Applications, pp 1–18. doi:10.1007/s11042-013-1807-z
19. Islam SH (2014) Provably secure dynamic identity-based three-factor password authentication scheme using extended chaotic maps. Nonlinear Dyn 78(3):2261–2276
20. Jiang Q, Ma J, Tian Y (2014) Cryptanalysis of smart-card-based password authenticated key agreement protocol for session initiation protocol of zhang et al., International Journal of Communication Systems. doi:10.1002/dac.2767
21. Jo H, Lee Y, Kim M, Kim S, Won D (2009) Off-line password-guessing attack to Yang's and Huang's authentication schemes for session initiation protocol. In: Fifth International Joint Conference on INC, IMS and IDC (NCM '09), pp 618–621. doi:10.1109/NCM.2009.251
22. Koblitz N (1987) Elliptic curve cryptosystems. Math Comput 48(177):203–209
23. Miller VS (1986) Use of elliptic curves in cryptography. In: Advances in Cryptology (CRYPTO'85). Springer, pp 417–426
24. Mishra D, Mukhopadhyay S (2013) Cryptanalysis of Pairing-Free Identity-Based Authenticated Key Agreement Protocols. In: Inf Syst Secur. LNCS, pp 247–254
25. Pu Q (2010) Weaknesses of sip authentication scheme for converged voip networks. IACR Cryptol ePrint Arch 2010:464
26. Riaz S, Lee S-W (2014) A robust multimedia authentication and restoration scheme in digital photography. Multimed Tools Appl 73(3):1291–1321. doi:10.1007/s11042-013-1592-8
27. Rosenberg J, Schulzrinne H, Camarillo G, Johnston A, Peterson J, Sparks R, Handley M, Schooler E et al. (2002) Sip: session initiation protocol, Technical Report, RFC 3261, Internet Engineering Task Force
28. Salsano S, Veltri L, Papalilo D (2002) Sip security issues: the sip authentication procedure and its processing load. IEEE Netw 16(6):38–44
29. Secure Hash Standard (1995) FIPS PUB 180-1, National Institute of Standards and Technology (NIST), U.S. Department of Commerce
30. Syverson P, Cervesato I (2001) The logic of authentication protocols. In: Foundations of Security Analysis and Design. Springer, pp 63–137
31. Thomas M et al (2001) IETF Intemet dren (draftthomas-sip-sec-reg'OO. txt, Sip security requirements
32. Tsai JL (2009) Efficient nonce-based authentication scheme for session initiation protocol. IJ Netw Secur 9(1):12–16
33. Tu H, Kumar N, Chilamkurti N, Rho S (2014) An improved authentication protocol for session initiation protocol using smart card, Peer-to-Peer Networking and Applications, pp 1936–6442. doi:10.1007/s12083-014-0248-4

34. Wu L, Zhang Y, Wang F (2009) A new provably secure authentication and key agreement protocol for sip using ecc. Comput Stand Interf 31(2):286–291
35. Wu S, Pu Q, Kang F (2013) Practical authentication scheme for sip. Peer-to-Peer Netw Appl 6(1):61–74
36. Xie Q (2012) A new authenticated key agreement for session initiation protocol. Int J Commun Syst 25(1):47–54
37. Xu J, Zhu W-T, Feng D-G (2009) An improved smart card based password authentication scheme with provable security. Comput Stand Interfaces 31(4):723–728
38. Yang C-C, Wang R-C, Liu W-T (2005) Secure authentication scheme for session initiation protocol. Comput Secur 24(5):381–386
39. Yeh H-L, Chen T-H, Shih W-K (2014) Robust smart card secured authentication scheme on sip using elliptic curve cryptography. Comput Stand Interf 36(2):397–402
40. Yi X, Zheng G, Li M, Ma H, Zheng C (2014) Efficient authentication of scalable media streams over wireless networks. Multimed Tools Appl 71(3):1913–1935. doi:10.1007/s11042-012-1324-5
41. Yoon E-J, Yoo K-Y, Kim C, Hong Y-S, Jo M, Chen H-H (2010) A secure and efficient sip authentication scheme for converged voip networks. Comput Commun 33(14):1674–1681
42. Yoon E-J, Shin Y-N, Jeon I-S, Yoo K-Y (2010) Robust mutual authentication with a key agreement scheme for the session initiation protocol. IETE Tech Rev 27(3):203–213
43. 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, International Journal of Communication Systems. doi:10.1002/dac.2499
44. Zhang Z, Qi Q, Kumar N, Chilamkurti N, Jeong H-Y (2014) A secure authentication scheme with anonymity for session initiation protocol using elliptic curve cryptography, Multimedia Tools and Applications, pp 1–12. doi:10.1007/s11042-014-1885-6

Dheerendra Mishra completed his Bachelor of Science and Master of Science degrees from Jiwaji University, India in 2003 and 2005, respectively. He received Ph.D. from the Indian Institute of Technology, Kharagpur, India, in 2014. Currently, he is working as assistant professor, Department of Mathematics, The LNM Institute of Information Technology, Jaipur-302031, India. His research interests include digital rights management system, access control in cloud, cryptographic protocols.