Peer-to-Peer Netw. Appl. (2018) 11:900–916
DOI 10.1007/s12083-017-0595-z

A robust mutual authentication scheme for session initiation protocol with key establishment

Venkatasamy Sureshkumar[1] · **Ruhul Amin**[2] · **R. Anitha**[1]

Received: 21 December 2016 / Accepted: 19 July 2017 / Published online: 29 July 2017

Abstract The Session Initiation Protocol (SIP) is a communication protocol that controls multimedia communication sessions. As the Internet users widely use SIP services, mutual authentication between the user and SIP server becomes an important issue. Several authentication protocols for SIP have been proposed for enhancing security and better complexities. Very recently, Lu et al. proposes an authenticated key agreement protocol for SIP and claims that it withstands various attacks and efficient. This paper points out that their protocol does not provide one of the most important features user anonymity. In addition, the same protocol is not able to resist user impersonation attack, server impersonation attack and fails to provide mutual authentication. The paper also presents an improved mutual authentication and key establishment protocol that conquers the security weaknesses in Lu et al.'s protocol. Informal security analysis is also carried out for several security properties. The formal proof for the correctness of mutual authentication and session key agreement is provided using BAN logic. It is shown that the proposed protocol is provably secure against identity and password guessing attacks in the random oracle model. The performance of the proposed scheme is compared with that of the existing related Elliptic Curve Cryptography (ECC) based schemes for SIP and shown that our scheme outperforms the others.

Keywords Authentication · Key agreement · Elliptic curve cryptography · Session initiation protocol · BAN logic

1 Introduction

Nowadays, the Internet telephony for video and voice calls are widely popular in the online multimedia services. There are many applications of the Internet that require the creation and management of a session. A session can be a collaborative multi-media conference or a simple two-way telephone call or a simple call to exchange data between two endpoints [22, 33, 45]. An endpoint can be a laptop, a smartphone or any device which can send and receive multimedia data through the Internet. This makes possible to implement services like web page click-to-dial, voice-enriched e-commerce or instant messaging with peer lists in an IP based system. The implementation of these applications is complicated as users are addressable by multiple names, move between endpoints and communicate via different media [21]. Various protocols have been proposed which carry several forms of real-time multimedia session data. SIP is a signaling protocol in an IP based network used to control multimedia communication sessions by creating, modifying, and terminating a session with several users over the Internet.

✉ Venkatasamy Sureshkumar
sand@amc.psgtech.ac.in; sanand1980serte@gmail.com

Ruhul Amin
amin_ruhul@live.com

R. Anitha
anitha_nadarajan@mail.psgtech.ac.in

[1] Department of Applied Mathematics and Computational Sciences, PSG College of Technology, Coimbatore 641004, India

[2] Department of Computer Science and Engineering, Thapar University, Patiala 147004, India

1.1 Background of SIP

SIP is used in Voice over Internet Protocol (VoIP) technology and is a choice for services related to VoIP [10, 27]. SIP is an application layer protocol, which works in association with a VoIP application to control multimedia communication sessions over the Internet. SIP applications include instant messaging, file transferring, online games, video conferencing and streaming multimedia distribution. SIP protocol is used for the Internet telephony and multimedia distribution between two or more endpoints [13, 31]. For instance, a user initiates a telephone call to another user using SIP, or someone creates a conference call with many participants. Two SIP endpoints can communicate without any intermediary SIP infrastructure. The SIP protocol is designed to be free from the underlying transport protocol. As it has only a limited set of commands, it is very simple. Also, anyone can read a SIP message passed between the endpoints in a SIP session, as it is text-based. But it should be noted that SIP is limited to only the setup and control of sessions [43]. The details of the data exchanged within a session related to an audio/video media is not controlled by SIP and is taken care of by other protocols. SIP is an agile, general-purpose tool, and is still growing and being modified to take into account all relevant features as the technology expands and evolves [44].

SIP defines user agents as well as several types of server network elements. SIP is executed between two user agents. Each user agent has two components client and server. Client at initiator user agent side sends a request to the server at responder user agent side, server sends a challenge to the client and the client sends the response to the challenge and hence complete an authentication process with establishment of a session key. After successful completion of the authentication process with key establishment, the client and server establish a channel between the two user agents using the session key. Using this established channel, the two user agents can start to exchange their multimedia data securely by executing SIP.

An Internet user can access the remote server to avail various multimedia services using SIP. In this scenario, the user authenticates to the server and vice-versa. This requires a vigorous mutual authentication scheme for SIP with key agreement to provide a strong security.

1.2 Related work

The SIP authentication scheme originates from the basic and digest access authentication protocol for hyper text transport [15]. On following that, several authentication schemes for SIP have been initiated in the literature [12, 28, 29, 31]. Later, Yang et al. [37] pointed out that these schemes are insecure and presented an improved authentication scheme for SIP based on the Diffie-Hellman key agreement technique. Huang & Wei [20] showed that Yang et al.'s scheme computation cost is high and it is not applicable for the system that has limited computational power. Moreover, He et al. [18] showed that Yang et al.'s scheme fails to resist the Denning-Sacco, off-line password guessing and stolen-verifier attacks.

Tsai [32] designed a robust authentication scheme for SIP that uses only XOR operations and one-way hash functions. However, Yoon et al. [40] pointed out that the scheme by Tsai [32] is vulnerable to the Denning-Sacco, stolen-verifier and off-line password guessing attacks. They presented an enhanced protocol to overcome the drawbacks in Tsai's scheme. However, Xie [36] has pointed out that Yoon et al. [40] protocol fails to resist off-line password guessing and stolen-verifier attacks. They also presented an improved scheme that rectifies the stipulated shortcomings. Unfortunately, Farash et al. [14] showed that Xie's protocol cannot withstand off-line password guessing and the impersonation attacks. They also proposed an enhanced protocol that overcomes the weaknesses in Xie's scheme.

Since ECC uses limited key size and provides the same level of security in comparison with RSA cryptosystem, it is better to use ECC to design SIP protocol. Wu et al. [35] designed a provably secure ECC based authenticated key establishment protocol. However, Yoon et al. [41] showed that Wu et al.'s scheme fails to withstand the Denning-Sacco, off-line password guessing and stolen verifier attacks. In order to overcome the stipulated issues, Yoon et al. presented an enhanced authentication protocol for SIP using ECC. However, Gokhroo et al. [17] pointed out that Yoon et al.'s scheme is vulnerable to the replay attack and off-line password guessing attack.

Anonymity also becomes a significant issue in SIP since it could protect user's privacy. However, Zhang et al. [42] pointed out that these authentication schemes are not providing user anonymity. They also presented an efficient and secure password based authenticated key agreement scheme for SIP. Recently Lu et al. [23] showed that Zhang et al.'s scheme is vulnerable to insider attack and it does not provide proper mutual authentication. In order to overcome the drawbacks in Zhang et al. scheme, Lu et al. [23] proposed a secure and efficient mutual authentication scheme for SIP. However, in this paper, it is shown that Lu et al. [23] protocol does not preserve user anonymity and it is completely insecure against impersonation attack.

It should be noted that most of the existing authentication protocols are not preserves user anonymity and not secured against off-line password guessing attack. Therefore, there is a need to develop a robust user anonymous authentication protocol for SIP that will resist off-line password guessing attack and many other known security attacks.

In this paper, an anonymity preserving mutual authentication protocol for SIP environment overcoming the security issues of Lu et al. [23] protocol is designed. Mutual authentication property of the proposed protocol is proved using BAN logic and the informal security analysis ensures resilience of off-line password guessing attack, impersonation attack, man-in-the-middle attack etc. The protocol is shown to be provably secure in random oracle model. In addition, our protocol offers password change and password recover facility to the registered users.

1.3 Adversary model

The adversary model in this section describes some valid assumptions about the authentication protocol, these assumptions are widely accepted [3, 4, 25, 30].

- An adversary can eavesdrop all the messages communicated between the protocol entities over the public channel, whereas the adversary has no control over the messages communicated over the private channel.
- The adversary can guess low-entropy identity and password of the user without difficulty, but guessing more than one secret parameter at a time is not feasible in polynomial time.
- The adversary can delete, modify, reroute and resend the eavesdropped messages.
- A legitimate user can be an adversary or vice versa.
- The probability of success for guessing the user's identity or password composed of n characters is approximately $\frac{1}{2^{6n}}$ as mentioned in [1, 6]

The paper is organised as follows. Section 2 describes Lu et al.'s scheme in brief and the security pitfalls of the same scheme are detailed in Section 3. The proposed protocol is described in Section 4. The BAN logic analysis and further security discussion of the proposed protocol against several security attacks are provided in Section 5. Section 6 details the comparative analysis of the proposed protocol with the existing related research. Section 7 concludes the paper.

2 Overview of Lu et al. scheme

Lu et al.'s scheme [23] consists of three phases mainly (1) registration, (2) authentication and (3) password change phase. The description of the three phases are detailed as below:

2.1 Registration phase

Step R1: User U selects his/her own identity ID, password PW and a secret key p_u. After that, U computes $PWD = h(PW||p_u)$ and sends $m_1 = \langle ID, PWD \rangle$ to the server S through a private channel.

Step R2: Upon the receipt of the registration message from the user, server S computes $VPW = h(ID||PWD) \oplus h(p_s)$ and backlog VPW in its database, where p_s is the secret key of the server.

2.2 Login and authentication phase

Step L1: User selects a random number r_u and computes

$$\begin{aligned} T &= h(ID||h(PW||p_u)) \\ A &= r_u \cdot P \\ B &= T \oplus A \\ HID &= ID \oplus T \\ C &= h(ID||A) \end{aligned}$$

where P is a point generator in the cyclic group of ECC cryptosystem. After that U sends the message $m_2 = \langle B, HID, C \rangle$ to S.

Step L2: On receiving this message, S extracts T' from VPW as $T' = VPW \oplus h(p_s)$ and computes

$$\begin{aligned} A' &= B \oplus T' \\ ID' &= HID \oplus T' \\ C' &= h(ID'||A') \end{aligned}$$

The server also verifies the correctness of the equation $C' \stackrel{?}{=} C$. If it is correct, then the server selects a random number r_s and computes

$$\begin{aligned} D &= r_s \cdot P \\ SK_s &= r_s \cdot A \\ Auth_s &= h(SK_s||T||A) \end{aligned}$$

At last, S transmits the challenge message $m_3 = \langle D, Auth_s \rangle$ to U.

Step L3: After receiving this message from the server, U computes $SK_u = r_u \cdot D$, $Auth'_s = h(SK_u||T||A)$ and verifies whether $Auth'_s \stackrel{?}{=} Auth_s$ holds. If it does not hold, the user terminates the session. Otherwise, U computes $Auth_u = h(SK_u||T||D)$ and sends the response message $m_4 = \langle Auth_u \rangle$ to the server.

Step L4: In receipt of this message, the server computes $Auth'_u = h(SK_s||T||D)$ and checks for the correctness of $Auth'_u \stackrel{?}{=} Auth_u$. If it is correct, then the server and user share the symmetric key $SK = SK_u = SK_s$ for the future session.

2.3 Password change phase

The user selects his new password PW^{new} and the new secret key p^{new}. Then the following steps are executed between the user and server:

Step PC1: User computes $V = h(SK||h(ID||h(PWD||p_u)))$, $M = h(ID||SK) \oplus h(ID||h(PWD^{new}||p_u^{new}))$ and sends the message $m_5 = \langle ID, V, M\rangle$ to the server.

Step PC2: After receiving password change request, server computes $V' = h(SK||VPW \oplus h(p_s))$ and checks the correctness of $V' \stackrel{?}{=} V$. If it is correct, the server computes $VPW^{new} = h(p_s) \oplus h(ID||SK) \oplus M$ and then puts VPW^{new} in the place of VPW in its database.

3 Security pitfalls in Lu et al. scheme

This section briefly describes several security pitfalls of the scheme proposed by Lu et al. such as off-line identity guessing attack, user impersonation attack, server impersonation attack and inefficient registration phase. The details of all the security weaknesses of Lu et al. scheme are shown below:

Off-line identity guessing attack In online business, it is commonly assumed that the internet user perpetually chooses easy to recall the identity for his/her benefits and as mentioned in [2], an adversary can guess it due to preserving low-entropy property. The authors in protocol [23] claimed that their protocol is user-anonymous. Hence, a malicious user is not able to find the identity of a legal user. However, the scheme in [23] is not user-anonymous due to revealing of the identity of the user by the adversary as below.

Algorithm 1

1: Input: $\langle B, HID, C\rangle$
2: Output: correct ID.
3: Adversary chooses a identity ID.
4: Adversary computes $C' = h(ID||(B \oplus (ID \oplus HID)))$
5: **if** $(C == C')$ **then**
6: Returns(ID); Succeeded in guessing user's identity.
7: **else**
8: Go to step *3*
9: **end if**

An adversary captures the message $m_1 = \langle B, HID, C\rangle$ and extracts all the components B, HID, and C from m_1. After that, the adversary guesses the user identity as ID and computes $C' = h(ID||(B \oplus (ID \oplus HID)))$. The adversary checks whether his guess is correct or not by verifying $C' \stackrel{?}{=} C$. If this condition is satisfied, the guessed identity is correct; else he guesses another identity and follows the same method.

$$\begin{aligned} C &= h(ID||A) \\ &= h(ID||(B \oplus T)) \\ &= h(ID||(B \oplus (ID \oplus HID))) \\ &= C'. \end{aligned}$$

Further a procedure to reveal the identity is provided in Algorithm 1.

Also during the execution of password change phase, ID can be retrieved easily by the adversary, as it is transmitted in plain-text form.

User impersonation attack After succeeded in ID guessing, an attacker can impersonate as a legal user. The attacker intercepts the login message $m_2 = \{B, HID, C\}$ of the legal user and extracts all the components. The attacker chooses a random number $r_u^* \in Z_q^*$ and computes $A^* = r_u^* \cdot P$ since P is a public parameter. The attacker also computes $T = HID \oplus ID$, $B^* = T \oplus A^*$ and $C^* = h(ID||A^*)$. Finally, the attacker sends the login message $m_2^* = \{B^*, HID, C^*\}$ to the server. This message will be authenticated in the server side as below:

Corresponding to HID, the server extracts VPW from the database and computes $T' = VPW \oplus h(p_s)$, $T' = T$ because HID is of the legal user, computes $A' = B^* \oplus T'$, $A' = A^*$ because $T' = T$ and computes $ID' = HID \oplus T'$, $ID' = ID$ because $T' = T$ and computes $C' = h(ID'||A') = C^*$ because $ID' = ID$ and $A' = A^*$. Thus, Lu et al. scheme is vulnerable to user impersonation attack.

Server impersonation attack As mentioned above, any attacker can extract T from the login message m_2 and he can impersonate as a legal server. The attacker chooses a random number r_s^*, computes $D^* = r_s^* \cdot P$, $A = B \oplus T$, $SK_s^* = r_s^* \cdot A$ and $Auth_s^* = h(SK_s^*||T||A)$. Finally, the attacker sends the challenge message $m_3^* = \{D^*, Auth_s^*\}$ to the user as a legal server. The legal user U computes $SK_u^* = r_u \cdot D^* = r_u r_s \cdot P = SK_s^*$ and $Auth_s' = h(SK_u^*||T||A)$ which is equal to $Auth_s^*$ because $SK_u^* = SK_s^*$. Hence, the message m_3 is authenticated in the user side. Thus, Lu et al.'s scheme is vulnerable to server impersonation attack.

Fails to preserve server secrecy The attacker somehow hacks the stored information from the server side and tries to break the security system. Here, we assume that information VPW is compromised somehow, and it is known to an attacker. If the attacker succeeds in guessing user identity ID, he can compute $T = HID \oplus ID$. Then, the attacker can compute the secret information $h(p_s)$ of the server as $VPW \oplus T$. It is noticed that long term secret information of the server should not be disclosed under any circumstances. Thus, this protocol fails to preserve server secrecy.

Inefficient registration phase During registration, the user chooses secret key p_u and uses it during login phase. Hence, the user should keep in mind three information ID, PW and p_u. As p_u is high entropy information, it is very hard to remember. Hence, the protocol is not user-friendly and realistic.

Inefficient authentication in password change phase Efficient password change phase requires user's login information for user authentication. The server needs to check the user's information for which the user should have sent the login information before password change. In Lu et al.'s scheme user authentication in password change phase is done without getting any login information, which is not efficient. Moreover, user authentication and password change process are carried out simultaneously. Hence, the password change phase is inefficient.

Replay attack In this instance, the adversary masquerades as a legal user by reusing the data received from the previously executed protocol. However, Lu et al.'s protocol cannot withstand the replay attack as below:

- Suppose an adversary traps the previous login message $m_2 = \langle B, HID, C\rangle$, where $B = T \oplus A$, $T = h(ID||h(PW||p_u))$, $A = r_u \cdot P$, $HID = ID \oplus T$ and $C = h(ID||A)$. Now, the adversary sends the message $m_2 = \langle B, HID, C\rangle$ to the server without altering it.
- After receiving the login message, the server extracts T' from VPW as $T' = VPW \oplus h(p_s)$ and computes $A' = B \oplus T'$, $ID' = HID \oplus T'$, $C' = h(ID'||A')$. The server also verifies the correctness of the equation $C' \stackrel{?}{=} C$. Obviously, this will be satisfied at server side.
- The server selects a random number r_s, computes $D = r_s \cdot P$, $SK_s = r_s \cdot A$, $Auth_s = h(SK_s||T||A)$ and sends $m_3 = \langle D, Auth_s\rangle$ to U.
- Finally, the adversary tries to compute the session SK_u or SK_s, which relies on the unknown secrets r_u and r_s and it is confirmed that the adversary cannot compute the session key of the scheme.

Even though, the adversary cannot compute the session key, the adversary still makes server's communication channel busy. However, this attack leads to a denial-of-service attack, if the adversary repeats many times previously captured login messages. This justification demonstrates that the Lu et al.'s scheme is vulnerable to replay attack.

4 Proposed protocol

In this section, we design a mutual authentication protocol for SIP with a key agreement scheme that overcomes the stipulated weaknesses found in Lu et al.'s scheme and provides strong security.

4.1 System setup phase

Server selects an additive cyclic group G of elliptic curve $E(F(q))$ defined over a finite field $F(q)$ of prime order q, a secret key x_s in Z_q^* and point generator P of G as mentioned in [5]. The server computes $P_s = x_s \cdot P$ and announces the public parameters $\langle G, q, P, P_s, E, h(\cdot)\rangle$, where $h : \{0,1\}^* \rightarrow \{0,1\}^q$ a collision resistant one-way hash function and keeps the private key x_s safely.

4.2 Registration phase

In this phase, user U registers with the server S. Since the registration phase is executed only once for a user, it can be done in a private secure channel.

Step PR1: User chooses his/her own identity ID, password PW and computes $HIP = h(ID||PW)$. The user also computes $HID = h(ID)$, $RP = ID \oplus PW$ and then sends the registration message $M_1 = \langle HID, HIP, RP\rangle$ to the server via secure channel. Although, we have used ID in the construction of all the three parameters HID, HIP and RP, the adversary has no access to the user ID. According to the assumption in Section 1.3, the adversary has no control over the messages communicated via the private channel. However in Lu et al. protocol, the ID is sent as a plain message in the registration phase.

Step PR2: On the receipt of the registration message from the user, server computes $UPW = h(HID||x_s) \oplus HIP$, where $x_s \in Z_q^*$ is the private key of the server. Finally, the server stores the pair $\langle UPW, RP\rangle$ corresponding to HID into his database. As these stored parameters contain three secret values namely ID, PW and x_s, it is not possible to predict. Thus, the ID is masked and stored in the system. Moreover, RP is computed only for the purpose of password recovery phase and it will not be used anywhere else in the protocol. This registration process is also depicted in Fig. 1.

4.3 Authentication and key generation phase

If the user U wants to login, the following steps should be performed.

Step PL1: U chooses a random number $r_u \in Z_q^*$, computes $R_u = r_u \cdot P$, $TR_u = r_u \cdot P_s$, where $P_s = x_s \cdot P$ is the public key of the server. The user U also computes $HID = h(ID)$, $HIP = h(ID||PW)$, $D = HID \oplus h(TR_u)$ and $A_u = h(HIP||TR_u||TS_u)$, then U sends the login message $M_2 = \langle R_u, D, A_u, TS_u\rangle$ to the server, where TS_u is the current timestamp chosen by the user. The parameters in M_2 do not contain ID, PW, HID or HIP in the explicit form. However, the parameters $D = h(ID) \oplus h(TR_u)$, $A_u = h(h(ID||PW)||TR_u||TS_u)$ contains ID and PW implicitly, it is computationally infeasible to extract them. More particularly, the parameters D and A_u contain the secret value TR_u.

Step PL2: After receiving the login message from the user, server checks $TS_s - TS_u < \triangle T$, where TS_s is the current timestamp chosen by the server and $\triangle T$ is an acceptable time delay. If the condition does not hold, the server aborts the connection, otherwise S computes $TR_u^* = x_s \cdot R_u$ and $HID = D \oplus h(TR_u^*)$. The server extracts HIP from the stored UPW corresponding to HID in the database. Now, the server computes $A_u^* = h(HIP||TR_u^*||TS_u)$ and checks for the correctness of $A_u^* \stackrel{?}{=} A_u$. If it is incorrect, then the server terminates the session. Otherwise, the server S confirms that U is a legitimate user, generates a random number $r_s \in Z_q^*$ and computes $R_s = r_s \cdot P$, $DK = r_s \cdot R_u$, $A_s = h(HIP||R_s||DK||TS_s)$ and $SK = h(TR_u^*||DK||HIP^*)$. Then the server sends the response message $M_3 = \langle R_s, A_s, TS_s\rangle$ to the user U.

Step PL3: Upon receiving the response message from the server, user checks $TS_{u1} - TS_s < \triangle T$, where TS_{u_1} is the current timestamp chosen by the user. If it does not hold, the user aborts the connection otherwise, computes $DK^* = r_u \cdot R_s$, $A_s^* = h(HIP||R_s||DK^*||TS_s)$ and checks whether $A_s^* \stackrel{?}{=} A_s$. If it is satisfied, then U confirms that S is a legitimate server, computes $SK = h(TR_u||DK^*||HIP)$, $T = h(A_u||A_s||SK)$ and sends $M_4 = \langle T\rangle$.

Step PL4: The message M_4 is a confirmation message to the server that confirms that the user has created the session key SK by computing $T^* = h(A_u||A_s||SK)$ and by checking $T^* \stackrel{?}{=} T$.

This login, authentication and key establishment phase is also depicted in Fig. 2.

4.4 Password change phase

If the user's password is leaked by any means, then there is a need for the user to change the password. The password based authentication protocol is robust, if it has an efficient password change phase. Hence, the password change provision is included in our protocol. It is assumed that the user has successfully completed the login authentication phase and established a new session key SK with old PW before

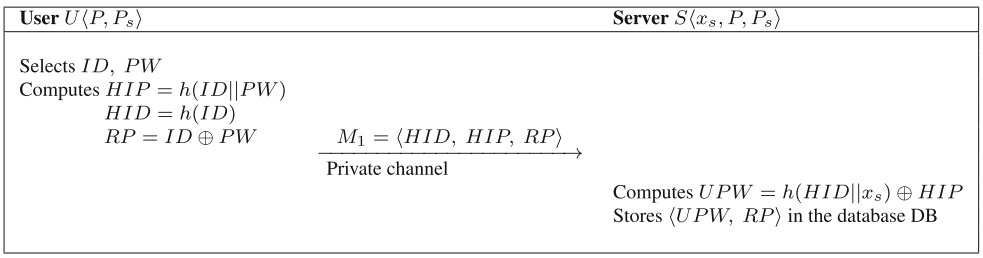

Fig. 1 User registration phase

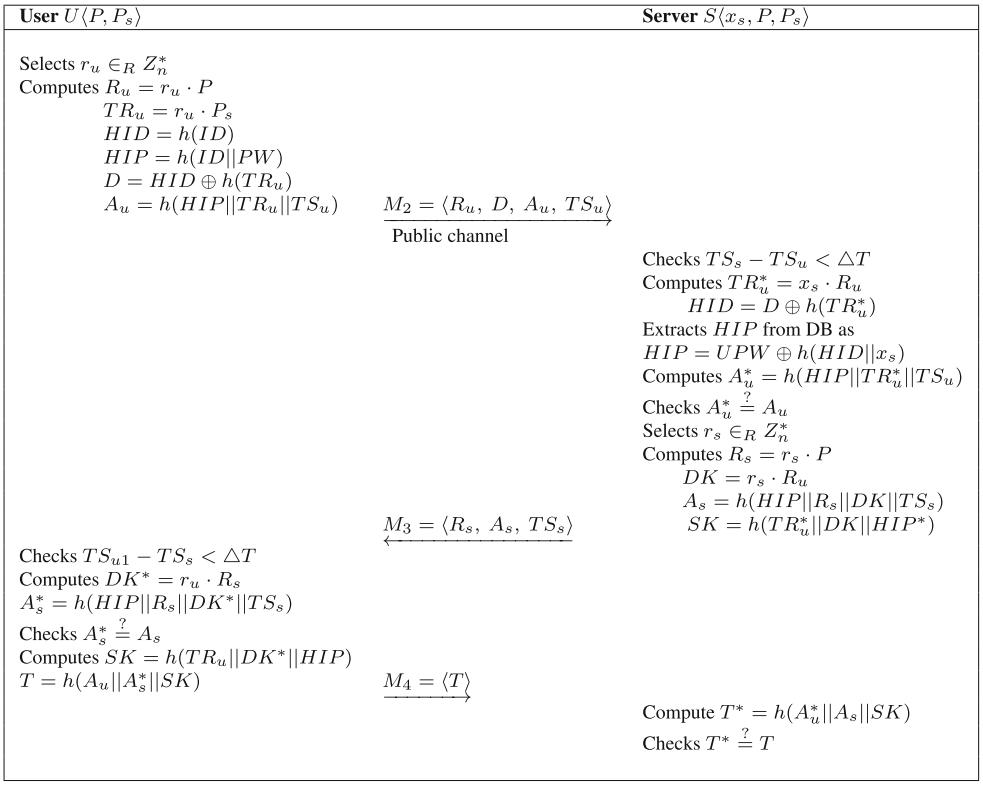

Fig. 2 Login and key establishment phase

starting of the password change phase. The description of this phase is as follows:

Step PPC1: The user selects a random number $r_1 \in Z_q$ and computes the following

$$\begin{aligned} R_1 &= r_1 \cdot P \\ DR_1 &= r_1 \cdot P_s \\ HID &= h(ID) \\ PK_1 &= HID \oplus h(DR_1) \\ HIP &= h(ID||PW) \\ CF &= h(HID||HIP) \end{aligned}$$

then sends the message $M_5 = \langle R_1, PK_1, CF\rangle$ to the server. On receipt of this message, the server computes $DR_1^* = x_s \cdot R_1$, $HID^* = PK_1 \oplus h(DR_1^*)$, retrieves UPW from the database corresponding to the received HID^* and computes $TID^* = h(HID^*||x_s)$, $HIP^* = UPW \oplus TID^*$ and $CF^* = h(HID^*||HIP^*)$. The server checks whether the received CF matches with the computed CF^*. If it does not hold, server rejects the session otherwise server selects a random number $r_2 \in Z_q$, computes $R_2 = r_2 \cdot P$, $DR_2 = r_2 \cdot R_1$ and sends the message $M_6 = \langle R_2, \text{“}Enter\ new\ password''\rangle$.

Step PPC2: The user selects his/her own new password PW^{new} and computes $DR_2^* = r_1 \cdot R_2$, $PK_2 = ID \oplus PW^{new} \oplus h(DR_2^*)$, $HIP^{new} = h(ID||PW^{new})$, the commitment $\phi = h(HID||DR_2^*||HIP||HIP^{new})$ and $DPW = h(DR_2^*||HIP) \oplus HIP^{new}$. Then, the user sends the message $M_7 = \langle R_1, PK_2, \phi, DPW\rangle$ to the server.

Step PPC3: After receiving the password change request, server computes $HIP^{new} = DPW \oplus h(DR_2||HIP)$, $\phi^* = h(HID||SK||HIP||HIP^{new})$ and checks whether $\phi^* \stackrel{?}{=} \phi$ holds or not. If it is satisfied, then the server computes $UPW^{new} = TID^* \oplus HIP^{new}$, $RP^{new} = PK_2 \oplus h(DR_2)$ and replaces $\langle HID, UPW, RP\rangle$ with $\langle HID, UPW^{new}, RP^{new}\rangle$ in its database, otherwise terminates the session. In this phase unlike Lu et al. scheme, checking the condition $\phi^* \stackrel{?}{=} \phi$ prevent any other user to impersonate U between login authentication

phase and password change phase. This password change phase is also depicted in Fig. 3.

4.5 Password recovery phase

Suppose a user wants to login into the server after long-time and forgets his/her password. The re-registration process consumes time and therefore, it is essential to recover the password for the user. The registered password can be recovered in our protocol by executing the following steps.

Step PPR1: The user U chooses a random number $r \in Z_q^*$ and computes the following

$$\begin{aligned} K_1 &= r \cdot P \\ K_2 &= r \cdot P_s \\ HID &= h(ID) \\ PWR &= HID \oplus h(K_2) \\ Auth &= h(P||K_2) \end{aligned}$$

then U sends the message $M_8 = \langle K_1, PWR, Auth \rangle$ to the server.

Step PPR2: After receiving M_8, the server computes $K_2^* = x_s \cdot K_1$, $Auth^* = h(P||K_2^*)$ and checks $Auth^* \stackrel{?}{=} Auth$. If it is not satisfied, the server terminates the session, otherwise the server computes $HID^* = PWR \oplus h(K_2^*)$ and extracts RP, UPW corresponds to HID^* from the database. The server computes $TID = h(HID||x_s)$, $HIP = UPW \oplus TID$, $AS = h(RP||HIP)$ and sends the message $M_9 = \langle RP, AS \rangle$ to the user. Since AS contain HIP which is composed of two secrets ID and PW inside hash. It is computationally infeasible to predict two unknown parameters simultaneously, which is addressed in the security analysis section.

Step PPR3: Upon receiving the message M_8, user recovers the password by computing $PW = RP \oplus ID$, also computes $AS^* = h(RP||h(ID||PW))$ and ensures it by checking $AS \stackrel{?}{=} AS^*$.

Lu et al. scheme does not contain this phase and this phase is also essential. The password recovery phase is also depicted in Fig. 4.

5 Analysis of the proposed protocol

In this section, we analyze our protocol informally to prove that it overcomes the security pitfalls in Lu et al. scheme.

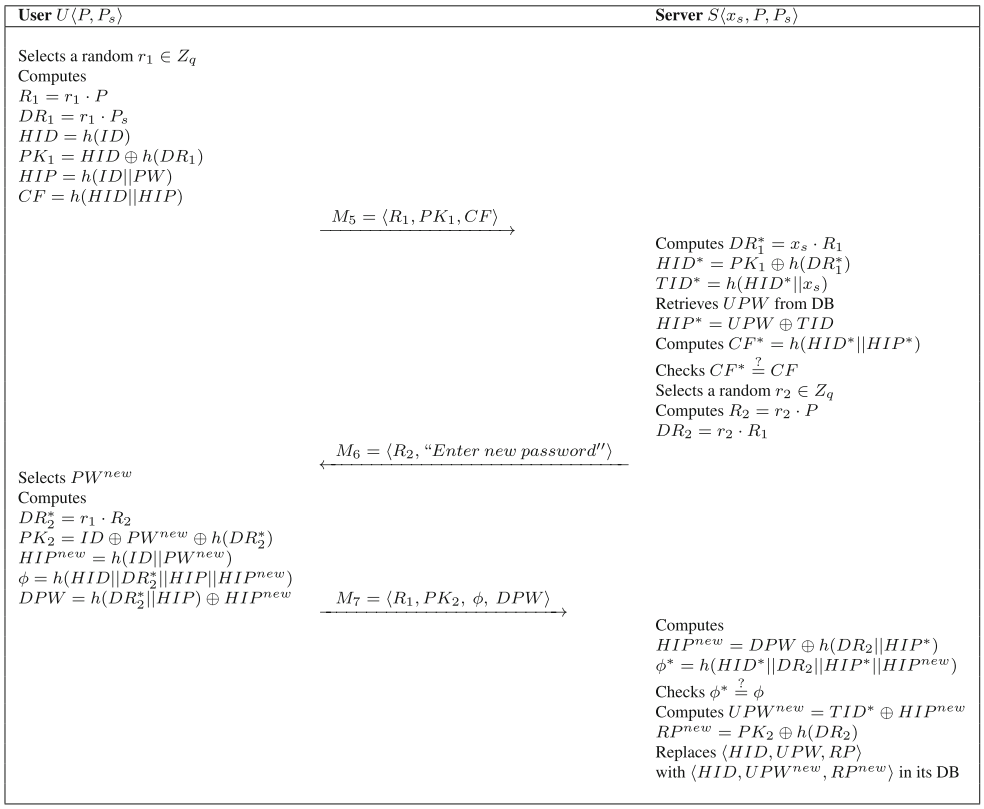

Fig. 3 Password change phase

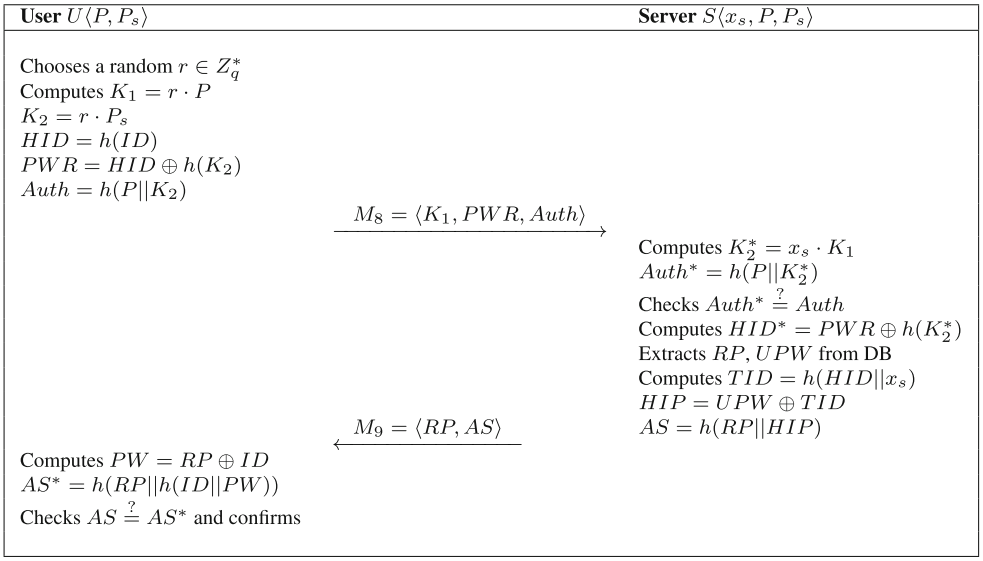

Fig. 4 Password recovery phase

Further, the analysis using BAN logic formally proves that our scheme achieves mutual authentication between user and server and the analysis in random oracle model shows that the scheme is provably secure against identity and password guessing attacks.

5.1 Informal security analysis

With respect to the adversary model described in Section 1.3, the protocol is analyzed against relevant security attacks.

Proposition 1 *The proposed protocol can withstand the off-line identity guessing attack.*

Proof Theidentity of the user is masked using one-way hash function, hence the attacker cannot extract it while executing the proposed protocol. Suppose a user selects a low entropy identity for his/her easy remembrance. In this situation, an attacker may guess the user's identity and try to verify his/her guess. The attacker captures the login message $m_2 = \{R_u, D, A_u\}$ of the protocol, and the attacker extracts the parameters R_u, D, and A_u. The parameter D is constructed as $D = HID \oplus h(TR_u)$, where $HID = h(ID)$ and $TR_u^* = x_s \cdot R_u$. If the attacker attempts to check the correctness of his/her guessed user identity, he/she needs to guess two factors ID and x_s simultaneously which is computationally infeasible. If each of the user identity and server secret x_s are of length n, then the probability of success for the correct guess is $\frac{1}{2^{12n}}$. The parameter A_u is constructed as $A_u = h(HIP||TR_u||R_u)$,where $HIP = H(ID||PW)$. The attacker cannot extract ID from this, however the attacker can guess ID^g and try to check his/her guess. The parameter A_u contains three unknown factors ID, PW and TR_u. To check the correctness, the attacker needs to guess three factors simultaneously and the probability for success is $\frac{1}{2^{18n}}$ which is significantly small. Hence, the proposed protocol can withstand the off-line identity guessing attack. □

Proposition 2 *The proposed protocol can withstand the off-line password guessing attack.*

Proof Suppose that an attacker guess the user's password and try to verify it. The attacker captures the login-response messages $M_2 = \{R_u, D, A_u\}$, $m_3 = \{R_s, A_s\}$, and extracts the parameters R_u, D, R_s, A_u and A_s from these messages. The parameter A_u is constructed as $A_u = h(HIP||TR_u||R_u)$, where $HIP = h(HID||PW)$. If the attacker attempts to check the correctness of his/her guessed user identity, he/she needs to guess three factors ID, PW and TR_u simultaneously. The probability of success for the correct guess is $\frac{1}{2^{12n+m}}$ which is very small. The parameter A_s is constructed as $A_s = h(HIP||R_s||DK)$, where $DK = r_s \cdot R_u$ and r_s is a secret random value known only to the server. If the attacker attempts to check the correctness of his/her guessed user identity, he/she needs to guess three factors ID, PW and DK simultaneously. Then the probability of success for the correct guess is $\frac{1}{2^{18n}}$ which is a negligible one. □

Proposition 3 *The proposed protocol preserves user anonymity property.*

Proof Suppose an attacker tries to identify a particular user and captures the login message of the proposed protocol. The login message is $m_2 = \{R_u, D, A_u\}$, where $R_u = r_u \cdot P$,

$D = HID \oplus h(TR_u)$, $A_u = h(HIP||TR_u||TS_u)$, $R_s = r_s \cdot P$ and $A_s = h(HIP||R_s||DK)$. As the parameter R_u contains the random value r_u, message m_2 is dissimilar in all the communications and hence the attacker cannot realize a particular message for a specific user. Also the user identity ID is masked using the hash function, the attacker cannot extract the ID from the transmitted messages. Thus, the proposed protocol provides user anonymity property. □

Proposition 4 *The proposed protocol provides untraceability.*

Proof If an attacker wants to trace the legal user then he/she captures all the login-response messages transmitted in the protocol. The login and response messages are $m_2 = \{R_u, D, A_u\}$ and $m_3 = \{R_s, A_s\}$ respectively, where $R_u = r_u \cdot P$, $D = HID \oplus h(TR_u)$, $A_u = h(HIP||TR_u||TS_u)$, $R_s = r_s \cdot P$, $A_s = h(HIP||R_s||DK)$, $R_s = r_s \cdot P$ and $A_s = h(HIP||R_s||DK)$. Each time the login message m_2 is different since it contains D, A_u and $R_u = r_u \cdot P$, where r_u is a randomly selected value. Moreover, the response message contains parameter R_s which is constructed using the random value r_s. This makes the attacker suspicious to link the login and response messages with the concerned user. Hence, attacker fails to break untraceability property. □

Proposition 5 *The proposed protocol is secure against insider attack.*

Proof In the proposed protocol, the user sends masked password for registration to the server. Therefore, an insider of the system can not extract the user ID and password due to the collision resistant and non-invertible property of the cryptographic one-way hash function. Therefore, the proposed scheme can withstand privileged insider attack. □

Proposition 6 *The proposed protocol can withstand user impersonation attack.*

Proof In this attack, an attacker captures the legal user's login message and creates forged login message. We claim that, the attacker cannot succeed in his/her attempt. The attacker can choose r_u^a in random and compute R_u^a, TR_u^a and $h(TR_u^a)$ but, computation of correct HID is infeasible as it contains the secret parameter ID. Similarly, the attacker cannot compute correct A_u as it involves the component HIP that contains two unknown parameters ID, PW of the legal user. The probability of success for the correct guess is $\frac{1}{2^{12n}}$ which is a negligible one. Thus, the attacker cannot compute valid D and A_u. Hence proposed protocol withstands user impersonation attack. □

Proposition 7 *The proposed protocol can withstand server impersonation attack.*

Proof In this attack, an attacker entraps the response message of a legal server and tries to create a forged response message. We claim that the attacker cannot succeed in his/her attempt. The attacker can select r_s^a at random and compute R_s^a but, cannot create TR_s and HIP as they contain the unknown parameters x_s and ID, PW respectively. □

Proposition 8 *The proposed protocol can withstand the session key computation attack.*

Proof After performing successful mutual authentication, the server and client attempts to establish a fresh session key which they can use for their secure communication. The session key SK is the hash value of R_u, R_s, DK^* and HIP. Even though R_u, R_s are known parameters, DK^* and HIP are unknown parameters. Hence the computation of session key is infeasible. □

Proposition 9 *The proposed protocol is secure against replay attack.*

Proof In this attack, the attacker captures all the messages transmitted in the proposed protocol and tries to impersonate as a legal user/server using the captured messages. But, R_u is generated freshly in each session, $DK = r_s \cdot R_u$ is computed in server side at each session using R_u. If the message m_2 is a replay message then m_3 will not be authenticated in the user side, as user computes $DK^* = r_u \cdot R_s$ and checks $A_s^* \stackrel{?}{=} A_s$. In addition, both the messages m_2 and m_3 include timestamp, which also thwart replay attack. Hence, the proposed protocol is secure against replay attack. □

Proposition 10 *The proposed protocol provides perfect forward secrecy.*

Proof If an adversary compromise the long-term secret values such as user identity, user password and server's secret key, then adversary cannot compute the session key $SK = h(TR_u||DK^*||HIP)$ as it contains the component $DK = r_u r_s \cdot P$. In which the random values r_u, r_s are freshly generated, they cannot be predicted and it is an Elliptic Curve Discrete Logarithm Problem(ECDLP) [5]. Hence, the proposed protocol preserves perfect forward secrecy. □

Proposition 11 *The proposed protocol is secure against stolen-verifier attack.*

Proof Suppose an adversary steals the pair $\langle UPW, RP\rangle$ from the server's database and tries to extract user's HID and HIP to impersonate the server. This attack is impossible as $UPW = h(HID||x_s) \oplus HIP$ contains three unknown factors HID, x_s and HIP. Also, the guess of any parameter has the success probability $\frac{1}{2^{|2n}}$, which is highly negligible. □

Proposition 12 *The proposed protocol is secure against man-in-the-middle attack.*

Proof Suppose an adversary wants to know the session key by performing a man-in-the-middle attack. The adversary can choose r_u^a in random, as computation of HID requires the knowledge of the secret parameter ID, the adversary cannot compute HID. To construct M_2, the two parameters D and A_u are essential, which requires the knowledge of HID and HIP. Thus, the creation M_2 for the adversary becomes a tedious process. Also, the adversary can attempt to create forged response message after capture the message from the legal server. The adversary cannot succeed in the attempt. The adversary can select r_s^a at random, but cannot create a valid A_s as its construction requires the knowledge of the two unknown parameters HIP and DK. Hence, the adversary cannot compute the session key by performing a man-in-the-middle attack. □

5.2 Authentication proof using BAN logic

Burrows, Abadi and Needham (BAN) coined a logic for proving the correctness of authentication and key establishment protocols formally [9]. The BAN logic is one of the formal methods which is used for the analysis of security protocols. The concept of a 'fresh message' and all public and shared key primitives are modelled using the logic. Using this it is possible to idealize a challenge-response protocol. The belief of an entity in the truth of a statement is the basis for the logic. A statement needs not be true in the accepted sense of truth. A validation with BAN logic doesn't indicate that there are no attacks on the protocol always. A proof with the BAN logic is a valid proof of correctness, with respect to the assumptions given in [19]. However, the interpretation of the logic and the logic does rule out possible attacks are questionable.

5.2.1 Notations

This section details with some of the syntax of the BAN logic. Other syntactical rules are found in the article of Burrows, Abadi and Needham [9].

- P **believes** that X holds: $P \mid\equiv X$. It means that P believes that in the current run of the protocol that the formula X is true. it just shows what P believes X.
- P **sees** the formula X: $P \triangleleft X$. It can be said as: P holds X.
- P has **jurisdiction** over X: $P \mid\Rightarrow X$. The entity P has complete control over the formula X. This can be used when reasoning over Certificate Authorities.
- P has **once said** the formula X: $P \mid\sim X$. The past holds all earlier runs of the protocol and earlier messages of the current run of the protocol.
- X is **fresh**: $\sharp(X)$. The formula X is recent. The formula has not been used before, X is a nonce.
- X is **combined** with Y: $\langle X\rangle_Y$. The formula X is combined with the formula Y.
- X is **hashed** with Y: $(X)_Y$. The formula X is hashed with the formula Y.
- P and Q **share** a secret formula : $P \overset{X}{\rightleftharpoons} Q$. The formula X is a secret known only to P and Q.

5.2.2 Rules of inference

A short overview of the introduction, usage and elimination rules are given. The overview is not complete, but is sufficient for the analysis in this section. The rules are also the most used rules.

Message-meaning rule: If P believes that the secret Y is shared with Q and sees $\langle X\rangle_Y$, then P believes that Q once said X.

$$\frac{P \mid\equiv Q \overset{Y}{\rightleftharpoons} P,\ P \triangleleft \langle X\rangle_Y}{P \mid\equiv Q \mid\sim X}$$

For random values

$$\frac{P\ Chooses\ random\ X}{P \mid\equiv \sharp X}$$

Nonce-verification rule: If P believes that X could have been uttered only recently (in the present) and that Q once said X (either in the past or in the present), then P believes that Q believes X.

$$\frac{P \mid\equiv \sharp X,\ P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\equiv X}$$

Jurisdiction rule: If P believes that Q has jurisdiction over X then P trusts Q on the truth of X

$$\frac{P \mid\equiv Q \mid\Rightarrow X,\ P \mid\equiv Q \mid\equiv X}{P \mid\equiv X}$$

Freshness rule: If one part of a formula is fresh, then the entire formula must also be fresh:

$$\frac{P \mid\equiv \sharp X}{P \mid\equiv \sharp (X,\ Y)}$$

Session key rule: If the principal P believes that the parameter X is fresh and the principal P and Q believe X, which are the necessary parameters of the session key, then principal P believes that s/he shares the session key K with Q.

$$\frac{P \mid\equiv \sharp X, P \mid\equiv Q \mid\equiv X}{P \mid\equiv P \overset{K}{\rightleftharpoons} Q}$$

A composite message can be made when a principal believes in both parts. This can be generalised to more than two parts

$$\frac{P \mid\equiv X,\ \ P \mid\equiv Y}{P \mid\equiv (X,\ Y)}$$

Additional rules on multipart messages.

$$\frac{P \mid\equiv Q \mid\sim (X,\ Y)}{P \mid\equiv Q \mid\sim X}$$

$$\frac{P \mid\equiv Q \mid\equiv (X,\ Y)}{P \mid\equiv Q \mid\equiv X}$$

$$\frac{P \mid\equiv (X,\ Y)}{P \mid\equiv X}$$

$$\frac{P \lhd (X,\ Y)}{P \lhd X}$$

5.2.3 Protocol idealization

The concrete protocol needs to be idealized in the BAN logic syntax for the proper analysis.

$$m_2 = U \rightarrow S\ :\ R_u, D : \langle HID \rangle_{h(TR_u)}, A_u : (TS_u, TR_u)_{U \overset{HIP}{\rightleftharpoons} S}$$
$$m_3 = S \rightarrow U\ :\ R_s, A_s : (R_s, DK, TS_s)_{U \overset{HIP}{\rightleftharpoons} S}$$
$$m_4 = U \rightarrow S\ :\ T : (A_u, A_s)_{U \overset{SK}{\rightleftharpoons} S}$$

The primary goals are to confirm that the user believes, the server belief that the user and server shares the same session key and vice versa. To achieve these goals, we need two more security goals; user and server believes that they have shared the same session key. As the session key contains two secret parameters DK and TR_u, server should believe TR_u and user should believe DK. Now we list the following goals based on the BAN logic, which needs to be achieved to ensure the formal verification of the mutual authentication.

5.2.4 Security goals

The list of security goals need to be achieved are listed below.

$$G_1\ :\ S \mid\equiv TR_u$$
$$G_2\ :\ U \mid\equiv DK$$
$$G_3\ :\ U \mid\equiv U \overset{SK}{\rightleftharpoons} S$$
$$G_4\ :\ S \mid\equiv U \overset{SK}{\rightleftharpoons} S$$
$$G_5\ :\ U \mid\equiv S \mid\equiv U \overset{SK}{\rightleftharpoons} S$$
$$G_6\ :\ S \mid\equiv U \mid\equiv U \overset{SK}{\rightleftharpoons} S$$

5.2.5 Initial assumptions

The principals U and S believe that the random numbers generated in the protocol are fresh. In addition, the server and user should have the control on the secret parameters TR_u and DK respectively, which they have created. Based on the set of required security goals,the following assumptions about the initial state of the protocol are made to analyze the proposed protocol.

$$A_1\ :\ U \mid\equiv \sharp R_u$$
$$A_2\ :\ S \mid\equiv \sharp R_s$$
$$A_3\ :\ U \mid\equiv \sharp R_s$$
$$A_4\ :\ S \mid\equiv \sharp R_u$$
$$B_1\ :\ U \mid\equiv U \overset{TR_u}{\rightleftharpoons} S$$
$$B_2\ :\ U \mid\equiv U \overset{HIP}{\rightleftharpoons} S$$
$$B_3\ :\ S \mid\equiv U \overset{HIP}{\rightleftharpoons} S$$
$$C_1\ :\ S \mid\equiv U \mid\Rightarrow TR_u$$
$$C_2\ :\ U \mid\equiv S \mid\Rightarrow DK$$

5.2.6 Scheme analysis

To achieve the required security goals, sequence of rules are imposed on the idealized protocol along with the initial assumptions. Applying seeing rule on m_2, we get $S \lhd \langle R_u,\ TR_u \rangle_{HIP}$ and by assumption B_3, $S \mid\equiv U \overset{HIP}{\rightleftharpoons} S$. Now applying message meaning rule,

$$\frac{S \mid\equiv U \overset{HIP}{\rightleftharpoons} S,\ S \lhd \langle R_u,\ TR_u \rangle_{HIP}}{S \mid\equiv U \mid\sim \langle R_u,\ TR_u \rangle}$$

Thus, we have $S1\ :\ S \mid\equiv U \mid\sim \langle R_u,\ TR_u \rangle$. By assumption A_4, $S \mid\equiv \sharp R_u$ implies that $S \mid\equiv \sharp (R_u, TR_u)$. Using nonce verification rule with $S1$, we get $S \mid\equiv U \mid\equiv (R_u, TR_u)$. Applying conjunctions rule, $S2\ :\ S \mid\equiv U \mid\equiv (TR_u)$.

Applying Jurisdiction rule on C_1 and $S2$, we get $G_1 : S \mid\equiv TR_u$ which is nothing but **goal** G_1.

Seeing rule on m_3 implies that $U \triangleleft (R_s, DK)_{U \overset{HIP}{\rightleftharpoons} S}$, applying message meaning rule with the assumption B_3, we get $S3 : U \mid\equiv S \mid\sim \langle R_s, DK\rangle$. By assumption $A_3 : U \mid\equiv \sharp R_s$ and applying conjunctions rule, we get $S4 : U \mid\equiv \sharp(R_s, DK)$. Using nonce verification rule on $S4$ and $S3$, we get $S5 : U \mid\equiv S \mid\equiv \langle R_s, DK\rangle$. Applying conjunction rule, $S5 : U \mid\equiv S \mid\equiv DK$. Applying Jurisdiction rule on C_2 and $S5$, we get $G_2 : U \mid\equiv DK$ which is our **goal** G_2.

As TR_u is the necessary parameter for the construction of SK in the server side, by freshness rule, $\sharp(r_u)$ implies that $S \mid\equiv \sharp(TR_u)$. Applying the session key rule on $S2$, we get $G_3 : S \mid\equiv U \overset{SK}{\rightleftharpoons} S$ which is our **goal** G_3.

Similarly, DK is the necessary parameter for the construction of SK in the user side, by freshness rule, $\sharp(r_s)$ implies that $U \mid\equiv \sharp(DK)$. Applying the session key rule on $S5$, we get $G_4 : U \mid\equiv U \overset{SK}{\rightleftharpoons} S$. Hence, **goal** G_4 is achieved.

Using A_1 and nonce verification rule with G_3, we get **goal** G_5 which is $G_5 : U \mid\equiv S \mid\equiv U \overset{SK}{\rightleftharpoons} S$. Using C_2 and nonce verification rule with G_4, we get $G_6 : S \mid\equiv U \mid\equiv U \overset{SK}{\rightleftharpoons} S$ which is our **goal** G_6.

Thus, BAN logic is used to analyze the security of the proposed protocol and the results show that the protocol achieves mutual authentication between the user and the server.

5.3 Formal security proofs in random oracle model

The formal security analysis of the proposed protocol using random oracle model [11, 16, 24, 26] is presented in this section. The advantage and reveal oracle for a hash function of an adversary are defined as follows:

Definition 1 The advantage of an adversary $\mathcal{A}$ for finding collision in the cryptographic hash function h is $Adv^h_{\mathcal{A}}(t) = Prob([m_1, m_2] \leftarrow_R \mathcal{A}\ such\ that\ h(m_1) = h(m_2))$, where $Adv^h_{\mathcal{A}}(t)$ denotes the advantages of the probability over the random selection by $\mathcal{A}$ in the time duration t, $[m_1, m_2] \leftarrow_R \mathcal{A}$ denotes the messages $[m_1, m_2]$ selected by $\mathcal{A}$ are random and $Prob(E)$ denotes the probability of an event E. The hash function $h(\cdot)$ is said to be collision resistant if for any small value ϵ, $Adv^h_{\mathcal{A}}(t) \leq \epsilon$.

Now we define the reveal oracle as follows:

Definition 2 Reveal oracle denoted by $RORACLE(\cdot)$ is defined as an oracle that will unconditionally output the string m for a given hash value $h(m)$.

Definition 3 Extract oracle denoted by $EORACLE(\cdot)$ is defined as an oracle that will unconditionally output the string x for a given two ECC points $x \cdot P$ and P.

Algorithm 2 $ALGO1^h_{\Phi}$

Input: $\langle P,\ P_s,\ R_u,\ D,\ A_u,\ TS_u, R_s,\ A_s,\ TS_s, T\rangle$
Output: 0 or 1.
Capture the login message $\langle R_u,\ D,\ A_u,\ TS_u\rangle$ and response message $\langle R_s,\ A_s,\ TS_s\rangle$ in the execution of the protocol, where $R_u = r_u \cdot P$, $D = HID \oplus h(TR_u)$, $HID = h(ID)$, $TR_u = r_u \cdot P_s$, $A_u = h(HIP||TR_u||TS_u)$, $HIP = h(ID||PW)$, $R_s = r_s \cdot P$, $A_s = h(HIP||R_s||DK||TS_s)$, $DK = r_s \cdot R_u$, $T = h(A_u||A_s||SK)$, $SK = h(TR_u||DK||HIP)$ and TS_u, TS_s are timestamps.
Call $RORACLE()$ on input A_u for retrieving HIP and TR_u as HIP^* and TR^*_u, $\leftarrow RORACLE(A_u||TS_u)$
Call $RORACLE()$ on input HIP^* for retrieving ID and PW as ID' and PW', $\leftarrow RORACLE(HIP^*)$
Compute $HIP' = h(ID'||PW')$
if $HIP^* == HIP'$ **then**
    Compute $A^*_u = h(HIP^*||TR^*_u||TS_u)$
    if $A^*_u == A_u$ **then**
        Accept ID' and PW' as the correct identity and password of the valid user respectively.
        Return (1) Success
    else
        Return (0) Failure
    end if
else
    Return (0) Failure
end if

Theorem 1 *Suppose that the cryptographic hash function closely behaves like an oracle, the proposed protocol is provably secure against an adversary for obtaining the secret parameters $\langle ID, PW\rangle$ of a legal user though the adversary knows all the messages transmitted in the public channel.*

Proof Consider an adversary $\mathcal{A}$ who has the capacity to derive the identity and password of a legal user from the proposed protocol Φ. As mentioned in Section 1.3, $\mathcal{A}$ can capture the login message $\langle R_u,\ D,\ A_u,\ TS_u\rangle$ and response message $\langle R_s,\ A_s,\ TS_s\rangle$ in the execution of the protocol Φ. Then, $\mathcal{A}$ uses reveal oracle to execute the algorithm $ALGO^h_{\Phi}$ for deriving ID and PW of a legal user as depicted in the Algorithm 2.

Now we define the probability of success for the algorithm $ALGO^h_{\Phi}$ as $SUC1^h_{\Phi} = |Prob(ALGO^h_{\Phi} = 1) - 1|$. According to Definition 1, the advantage of $ALGO^h_{\Phi}$ is given by $Adv^h_{\Phi}(t_1, qr_1) = Max_{\mathcal{A}}(SUC1^h_{\Phi})$ where

maximum is taken over all the adversary $\mathcal{A}$ with the number of queries qr_1 made to $RORACLE()$ in the execution time t_1. The proposed protocol is provably secure against the adversary $\mathcal{A}$ for obtaining the secret parameters $\langle ID, PW\rangle$ if for any small value ϵ, $Adv^h_{\mathcal{A}}(t_1, qr_1) \leq \epsilon$. According to the algorithm $ALGO^h_{\Phi}$, the adversary $\mathcal{A}$ can obtain the secret parameters ID , PW and succeeded provided he/she has the ability to invert the hash function $h(\cdot)$. But, it is computationally infeasible to invert a hash function. Thus, $SUC1^h_{\Phi} \leq \epsilon$ for all the attackers. Since Adv^h_{Φ} depends on $SUC1^h_{\Phi}$ and maximum is taken over all the adversary $\mathcal{A}$, we have $Adv^h_{\mathcal{A}}(t_1, qr_1) \leq \epsilon$ for any small value ϵ. Thus, the proposed protocol is provably secure against the adversary $\mathcal{A}$ for obtaining the secret parameters $\langle ID, PW\rangle$. □

Algorithm 3 $ALGO2^h_{\Phi}$

Input: $\langle P,\ P_s,\ R_u,\ D,\ A_u,\ TS_u, R_s,\ A_s,\ TS_s, T\rangle$
Output: 0 or 1.
Capture the login message $\langle R_u,\ D,\ A_u,\ TS_u\rangle$ and response message $\langle R_s,\ A_s,\ TS_s\rangle$ in the execution of the protocol, where $R_u = r_u \cdot P$, $D = HID \oplus h(TR_u)$, $HID = h(ID)$, $TR_u = r_u \cdot P_s$, $A_u = h(HIP||TR_u||TS_u)$, $HIP = h(ID||PW)$, $R_s = r_s \cdot P$, $A_s = h(HIP||R_s||DK||TS_s)$, $DK = r_s \cdot R_u$, $T = h(A_u||A_s||SK)$, $SK = h(TR_u||DK||HIP)$ and TS_u, TS_s are timestamps.
Call $RORACLE()$ on input A_u for retrieving HIP and TR_u as HIP^* and TR^*_u, $\leftarrow RORACLE(A_u||TS_u)$
Call $RORACLE()$ on input HIP^* for retrieving ID and PW as ID' and PW', $\leftarrow RORACLE(HIP^*)$
Call $RORACLE()$ on input R_u for retrieving TR_u and x_s as TR'_u and x'_s, $\leftarrow RORACLE(R'_u)$
Compute $HID' = D \oplus h(TR'_u)$
if $HID^* == HID'$ **then**
    Compute $A^*_u = h(HIP^*||TR^*_u||TS_u)$
    if $A^*_u == A_u$ **then**
        Call $EORACLE()$ on input R_u and R_s for retrieving r_u, r_s as r'_u, r'_s, $\leftarrow EORACLE(R'_u, R_s)$
        Compute $R'_u = r'_u \cdot P$ and $R'_s = r_s \cdot P$
        if $R'_u == R_u \&\& R'_s == R_s$ **then**
            Accept r'_u, r'_s as the correct random numbers of the valid user and server respectively.
            Return (1) Success
        else
            Return (0) Failure
        end if
    else
        Return (0) Failure
    end if
else
    Return (0) Failure
end if

Theorem 2 *Suppose that the cryptographic hash function closely behaves like an oracle, the proposed protocol is provably secure against an adversary for obtaining the secret parameters $\langle r_u, r_s\rangle$ and the session key SK between the server and the legal user though the adversary knows all the messages transmitted in the public channel.*

The proof of this theorem is similar to that of the previous theorem using the Algorithm 3.

6 Performance comparison

In this part, we compare the performance of the designed protocol with that of the existing related protocols in two aspects; computational complexity and the advanced security functionalities.

6.1 Security features comparison

Table 1, shows the comparison of the proposed protocol with the related existing protocols in the aspect of several security functionalities. It is noted that, our scheme withstands relevant security threats and achieves required security attributes than other protocols. More preciously, Table 1 confirms that our scheme withstands all the security attacks mentioned in Lu et al. scheme.

6.2 Computational cost comparison

We have provided computational cost comparisons of our protocol with several related protocols in Table 2. The time complexity that measures the computation cost associated with hash and point multiplication operations can be expressed as $T_{pm} >> T_{pa} >> T_{inv} >> T_h$. It is most important for the cryptographic protocol that it must be free from security attacks. Though our scheme has more time complexity than [7, 8, 14, 18, 23], our scheme withstands the security attacks over those schemes and in these schemes user needs to maintain an additional random number as secret which makes protocol is not user-friendly and realistic. In addition to that, our scheme has the password recovery phase, which is not included in any of the existing SIP authentication protocols.

7 Conclusion

In this paper, we have showed that the recently proposed Lu et al.'s SIP authentication protocol has several security vulnerabilities including impersonation attack. In addition to that a robust authentication scheme for SIP with key establishment technique is proposed . Our scheme additionally

Table 1 Functionality and security comparison

Schemes ⇓	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}
Yoon et al. [39]	×	✓	✓	✓	×	✓	×	×	✓	✓
Xie [36]	×	✓	✓	✓	×	✓	×	✓	✓	×
He et al. [18]	×	✓	✓	×	×	✓	✓	✓	✓	✓
Farash-Attari [14]	×	✓	✓	✓	×	✓	✓	✓	✓	✓
Arshad-Ikram [8]	×	✓	✓	✓	×	×	✓	×	✓	✓
Zhang et al. [42]	✓	✓	✓	✓	×	×	✓	✓	✓	✓
Tu et al. [34]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Yeh et al. [38]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Zhang et al. [44]	✓	×	✓	✓	×	✓	✓	✓	✓	✓
Arshad-Nikooghadam [7]	×	×	✓	×	✓	✓	✓	✓	✓	✓
Lu et al. [23]	×	×	×	✓	✓	×	✓	✓	×	✓
Proposed Protocol	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

C_1: Achieves user anonymity, C_2: Achieves mutual authentication, C_3: Achieves perfect forward secrecy, C_4: Provides known session key security, C_5: Withstand insider attack, C_6: Withstand user masquerade attack, C_7: Withstand off-line password guessing attack, C_8: Withstand stolen-verifier attack, C_9: Withstand replay attack, C_{10}: Withstand man-in-the-middle attack ✓: Withstands the attack or satisfy that property; ×: Do not withstand the attack or not satisfy that property

contains password recovery phase, which is important for real-time application and this feature does not exist in the existing schemes. Informal security analysis against security attacks are described. The formal proof of correctness for mutual authentication using BAN logic is provided. In addition, the proposed protocol is shown to be provably secure against identity and password guessing attacks in random oracle model. Finally, we compared our scheme with the existing related schemes and shown that our scheme has better security with less complexity than the others. As a future work, the proposed protocol can still be enhanced such that it reduce the number of cryptographic operations used and achieves more security properties. Also, the automatic verification of the proposed protocol needs to be done using the popular tool Automated Validation of Internet Security Protocols and Applications (AVISPA).

Table 2 Computational cost comparison

Protocols ⇓	Computational cost
Yoon et al. [39]	$4T_{pm} + 6T_h + 3T_{pa}$
Xie [36]	$6T_{pm} + 6T_h + 1T_{pa}$
He et al. [18]	$6T_{pm} + 6T_h$
Farash-Attari [14]	$6T_{pm} + 8T_h$
Arshad-Ikram [8]	$5T_{pm} + 8T_h$
Zhang et al. [42]	$8T_{pm} + 9T_h + 2T_{pa}$
Tu et al. [34]	$6T_{pm} + 8T_h + 1T_{pa}$
Yeh et al. [38]	$12T_{pm} + 13T_h$
Zhang et al. [44]	$6T_{pm} + 8T_h$
Arshad-Nikooghadam [7]	$4T_{pm} + 9T_h + T_{inv}$
Lu et al. [23]	$4T_{pm} + 9T_h$
Proposed Protocol	$6T_{pm} + 12T_h$

T_h : Complexity of executing a hash function, T_{pm} : Complexity of executing a scalar point multiplication algorithm, T_{inv} : Complexity of executing a scalar point inverse, T_{pa} : Complexity of executing a ECC point addition algorithm

References

1. Amin R, Biswas G (2015) Cryptanalysis and design of a three-party authenticated key exchange protocol using smart card. Arab J Sci Eng 40(11):3135–3149
2. Amin R, Biswas G (2015) A novel user authentication and key agreement protocol for accessing multi-medical server usable in tmis. J Med Syst 39(3):1–17
3. Amin R, Biswas G (2015) A secure three-factor user authentication and key agreement protocol for tmis with user anonymity. J Med Syst 39(8):1–19
4. Amin R, Biswas G (2016) A secure light weight scheme for user authentication and key agreement in multi-gateway based wireless sensor networks. Ad Hoc Netw 36:58–80
5. Amin R, Islam SH, Biswas G, Khan MK, Kumar N (2015) An efficient and practical smart card based anonymity preserving user authentication scheme for tmis using elliptic curve cryptography. J Med Syst 39(11):1–18
6. Amin R, Islam SH, Biswas G, 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):1–20
7. 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
8. Arshad R, Ikram N (2013) Elliptic curve cryptography based mutual authentication scheme for session initiation protocol. Multimed Tools Appl 66(2):165–178
9. Burrows M, Abadi M, Needham RM (1989) A logic of authentication. In: Proceedings of the royal society of london a: Mathematical, physical and engineering sciences, vol 426. The Royal Society, pp 233–271

10. Chaudhry SA, Naqvi H, Sher M, Farash MS, Hassan MU (2015) An improved and provably secure privacy preserving authentication protocol for sip. Peer-to-Peer Network Appl 1–15
11. Das AK, Paul NR, Tripathy L (2012) Cryptanalysis and improvement of an access control in user hierarchy based on elliptic curve cryptosystem. Inform Sci 209:80–92
12. Duanfeng S, Qin L, Xinhui H, Wei Z (2004) Security mechanisms for sip-based multimedia communication infrastructure. In: International conference on communications, circuits and systems, ICCCAS 2004., vol 1. IEEE, pp 575–578
13. Farash MS (2016) Security analysis and enhancements of an improved authentication for session initiation protocol with provable security. Peer-to-Peer Network Appl 9(1):82–91
14. Farash MS, Attari MA (2013) An enhanced authenticated key agreement for session initiation protocol. Inform Technol Control 42(4):333–342
15. Franks J, Hallam-Baker P, Hostetler J, Lawrence S, Leach P, Luotonen A, Stewart L (1999) Http authentication: Basic and digest access authentication
16. Giri D, Sherratt RS, Maitra T, Amin R (2015) Efficient biometric and password based mutual authentication for consumer usb mass storage devices. IEEE Trans Consum Electron 61(4):491–499
17. Gokhroo MK, Jaidhar C, Tomar AS (2011) Cryptanalysis of sip secure and efficient authentication scheme. In: 2011 IEEE 3rd International conference on communication software and networks (ICCSN). IEEE, pp 308–310
18. 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
19. He D, Wang D (2015) Robust biometrics-based authentication scheme for multiserver environment. IEEE Syst J 9(3):816–823
20. Huang HF, Wei WC (2006) A new efficient authentication scheme for session initiation protocol. Computing 1(2):1–3
21. 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
22. Kumari S, Chaudhry SA, Wu F, Li X, Farash MS, Khan MK (2015) An improved smart card based authentication scheme for session initiation protocol. Peer-to-Peer Netw Appl. 1–14
23. Lu Y, Li L, Peng H, Yang Y (2016) A secure and efficient mutual authentication scheme for session initiation protocol. Peer-to-Peer Network Appl 9(2):449–459
24. Maitra T, Giri D (2014) An efficient biometric and password-based remote user authentication using smart card for telecare medical information systems in multi-server environment. J Med Syst 38(12):142
25. Messerges TS, Dabbish EA, Sloan RH (2002) Examining smart-card security under the threat of power analysis attacks. IEEE Trans Comput 51(5):541–552
26. Mishra D, Das AK, Mukhopadhyay S (2014) A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards. Expert Syst Appl 41(18):8129–8143
27. 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 Network Appl 9(1):171–192
28. Rosenberg J, Schulzrinne H, Camarillo G, Johnston A, Peterson J, Sparks R, Handley M, Schooler E et al (2002) Sip: Session initiation protocol. Tech. rep., RFC 3261 Internet Engineering Task Force
29. Salsano S, Veltri L, Papalilo D (2002) Sip security issues: The sip authentication procedure and its processing load. Netw IEEE 16(6):38–44
30. Sureshkumar V, Amin R, Anitha R (2017) An enhanced bilinear pairing based authenticated key agreement protocol for multi-server environment. Int J Commun Syst. doi:10.1002/dac.3358
31. Tam K, Goh H (2002) Session initiation protocol. In: 2002 IEEE International conference on industrial technology, 2002. IEEE ICIT'02., vol 2. IEEE, pp 1310–1314
32. Tsai JL (2009) Efficient nonce-based authentication scheme for session initiation protocol. Int J Netw Secur 9(1):12–16
33. Tu H, Kumar N, Chilamkurti N, Rho S (2015) An improved authentication protocol for session initiation protocol using smart card. Peer-to-Peer Network Appl 8(5):903–910
34. Tu H, Kumar N, Chilamkurti N, Rho S (2015) An improved authentication protocol for session initiation protocol using smart card. Peer-to-Peer Network Appl 8(5):903–910
35. Wu L, Zhang Y, Wang F (2009) A new provably secure authentication and key agreement protocol for sip using ecc. Comput Standards Interf 31(2):286–291
36. Xie Q (2012) A new authenticated key agreement for session initiation protocol. Int J Commun Syst 25(1):47–54
37. Yang CC, Wang RC, Liu WT (2005) Secure authentication scheme for session initiation protocol. Comput Secur 24(5):381–386
38. Yeh HL, Chen TH, Shih WK (2014) Robust smart card secured authentication scheme on sip using elliptic curve cryptography. Comput Standards Interf 36(2):397–402
39. Yoon EJ, Shin YN, Jeon IS, Yoo KY (2010) Robust mutual authentication with a key agreement scheme for the session initiation protocol. IETE Tech Rev 27(3):203–213
40. Yoon EJ, Yoo KY (2009) Cryptanalysis of ds-sip authentication scheme using ecdh. In: International conference on new trends in information and service science, 2009. NISS'09. IEEE, pp 642–647
41. 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
42. 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 27(11):2691–2702
43. Zhang L, Tang S, Zhu S (2016) A lightweight privacy preserving authenticated key agreement protocol for sip-based voip. Peer-to-Peer Netw Appl 9(1):108–126
44. 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
45. Zheng X, Oleshchuk V (2010) A survey on peer-to-peer sip based communication systems. Peer-to-peer Network Appl 3(4):257–264

Venkatasamy Sureshkumar received his M.Sc and M.Phil degree in Mathematics from Bharathiyar University in 2004 and Alagappa University in 2005 respectively. Currently, he is working as an Assistant Professor in the Department of Applied Mathematics and Computational Sciences, PSG College of Technology, Coimbatore, Tamilnadu, India. His research interests include Security protocols and Formal methods.

Ruhul Amin received his B.Tech. and M.Tech. degree from West Bengal University of Technology in Computer Science and Engineering Department in 2009 and 2013 respectively. He is currently working as a lecturer in the Department of Computer Science Engineering, Thapar University, Patiala, India. His current research interests include Cryptographic authentication protocol and security in wireless sensor network.

R. Anitha is an Associate Professor in the Department of Applied Mathematics and Computational Sciences, PSG College of Technology, Coimbatore, India. She received her PhD degree from Bharathiyar University, Coimbatore in 1997. She is Life member of CRSI, ISTE and Member of ACM. Her research interests include Cryptography, Security protocols, Information security and system security.