Wireless Personal Communications (2020) 110:2087–2107
https://doi.org/10.1007/s11277-019-06831-y

An Improved SIP Authenticated Key Agreement Based on Dongqing et al.

Mahmood Ul Hassan[1] · **Shehzad Ashraf Chaudhry**[2] · **Azeem Irshad**[3]

Published online: 14 January 2020

Abstract
The IP multimedia subsystem represents an architectural framework to support multimedia-based services using internet protocol over wired and wireless media. These IP-based multimedia services rely on session initiation protocol (SIP) for creating, maintaining and terminating the communicative sessions, which underscores the efficiency and security of SIP protocol. Many SIP based authentication schemes have been put forward in the last decade, however with many limitations. Recently, Lu et al. and Chaudhary et al. presented SIP based authentication protocols. Then, Dongqing et al. discovered limitations in Lu et al. and Chaudhary et al. schemes, and presented an improved SIP authentication protocol. Nonetheless, we ascertain that Dongqing et al.'s protocol is prone to privileged insider attack, denial of service attack, and session specific ephemeral secret-leakage attack. Besides, this protocol assumes a strictly time synchronized system, which limits the practical effectiveness of the protocol for a real environment. We also propose an improved SIP authentication protocol that covers the limitations of Dongqing et al. protocol. Our scheme is formally proved as secure using BAN logic analysis. The performance analysis illustrates the comparison for related schemes with proposed scheme, which depicts the efficiency and robustness of the scheme over previous schemes.

Keywords Session initiation protocol · Internet multimedia subsystem · Authentication · Cryptography · Cryptanalysis · Attacks

✉ Azeem Irshad
irshadazeem2@gmail.com

Mahmood Ul Hassan
mahmood-ul-hassan@iiu.edu.pk

Shehzad Ashraf Chaudhry
ashraf.shahzad.ch@gmail.com

[1] Department of Computer Science and Software Engineering, International Islamic University, Islamabad, Pakistan

[2] Department of Computer Engineering, Istanbul Gelisim University, Istanbul, Turkey

[3] Department of Computer Science, University of Sialkot, Sialkot, Pakistan

1 Introduction

The IP multimedia subsystem provides a generic framework for voice, data and video communication services available to mobile and land users [1, 2]. The advantage of IP multimedia subsystem is to offer, by using its middleware, unique and universal mechanisms for Quality of Service standards, charging criteria, authentication and security etc. This framework is based on session initiation protocol (SIP) [3], which is a text-oriented client server protocol to manage multimedia sessions [4]. It is one of the frequently used protocols to establish online communicating sessions for multimedia services between user and server. For making use of the SIP protocol, the client needs to be authenticated from SIP server initially, which is quite significant for secure multimedia-based communicating sessions.

In the last decade, several SIP protocols could be witnessed in the academia [3, 5–8]. For this, a pioneer scheme was demonstrated by Franks et al. for HTTP [9]. Onwards, Yang et al. [10] remarked that the current SIP protocol as based on HTTP, is less secure for having vulnerability for offline password guessing threat and stolen verifier threat. Besides, the protocol was not suitable for low end power deficient devices [11, 12]. Due to the short key size of elliptic curve cryptography (ECC), it is being employed in various cryptographic protocols, including SIP protocols. Durlanik et al. [13] also presented an efficient ECC-based SIP protocol. Afterwards, Wu et al. [14] demonstrated another ECC-based SIP protocol. However, the schemes [13, 14] are found to be prone for offline password guessing and stolen verifier threat by Yoon et al. [15]. Also, Yoon et al. demonstrated another improved SIP-based authentication protocol. However, Gokhroo et al. [16] and Pu [17] indicated Yoon et al. protocol is also susceptible to guessing and replay attacks. Thereafter, Tsai [18] presented a symmetric cryptography based SIP scheme using XOR operation, but was discovered to be vulnerable to many attacks [19–22]. Yoon et al. [22] put forwarded a SIP scheme after finding attacks on Tsai [18]. Nonetheless, Xie [23] pointed out few limitations including guessing and stolen-verifier attacks in [22], and suggested an improved protocol. Then, Farash et al. [24] discovered impersonation attack and guessing attack in Xie's protocol, and presented an improved SIP protocol. Thereafter, Zhang et al. [25] designed a simple and efficient password-based SIP authentication protocol, however, Lu et al. [26] discovered that [25] is not able to resist insider attack and fails to offer mutual authentication. Lu et al. presented an improved scheme countering the limitations in [25]. Afterwards, Chaudhary et al. [27] found user and server impersonation attacks in [26]. Recently, Dongqing et al. [28] found stolen verifier attack in Lu et al. [26] and session key attack in Chaudhary et al. [27], and presented an improved scheme. We discover that Dongqing et al. [28] is again susceptible to privileged insider threat, denial of service (DoS) attack, and session specific ephemeral secret-leakage attack. Besides, the scheme bounds the system to be adhere time synchronization feature, which is a tough assumption to be implemented. Considering those limitations, we propose an efficient and secure protocol as demonstrated formally using BAN logic analysis which can be witnessed from the forthcoming sections.

2 Preliminaries

We briefly illustrate hash-based operation, Bio-hashing function and elliptic curve cryptography (ECC).

2.1 Hash Function

A symmetric key-based one sided hash digest $h : \{0,1\}^* \rightarrow Z_q^*$ encompasses the subsequent properties:

1. The hash-digest operation h generates a string of predefined size on receiving an input of random length.
2. Using the hash operation, i.e. $h(a) = b$, it is an intractable problem to compute $h^{-1}(\mathrm{b}) = \mathrm{a}$;
3. If we are given a, it is difficult in polynomial terms to calculate a', such that $a' \neq a$, but $h(a') = \mathrm{h}(a)$;
4. Additionally, it is difficult in polynomial terms to calculate the pair a, a' given that $a' \neq a$, but $h(a') = h(a)$.

2.2 Elliptic Curve Essentials

The ECC can be defined with elliptic curve E/F_q as a set of different points located in the prime field F_q, on a non-singular elliptic curve (EC) [29] as shown below:

$$\omega^2 mod\, q = (\zeta^3 + u\zeta + v)\, mod\, q \tag{1}$$

such as u, v, ζ, ω ϵ F_q and $(4u^3 + 27b^2)$ mod $q \neq 0$. We characterize an EC point as $\psi(\zeta, \omega)$ as if Eq. (1) is conformed, where the point $\eta(\zeta, -\omega)$ being negative version of ψ, also we can say $\eta = -\psi$. We take $\psi(\zeta_1, \omega_1)$ and $\eta(\zeta_2, \omega_2)$ as two separate points on the above Eq. (1), though, the line ln, as tangent of the above Eq. (1) meets ψ and η while intersecting the curve at point $-\Theta(\zeta_3, -\omega_3)$. Similarly, its reflection on x-axis is on point $\Theta(\zeta_3, \omega_3)$, i.e. $\psi + \eta = \Theta$. The range of points $\mathrm{E/F_q}$ including *point at infinity* (O) comprise an EC cyclic group, i.e. $G_q = \{(\zeta, \omega)$*:* ζ, ω ϵ F_q and (ζ, ω) ϵ $E/F_q\}$ U $\{O\}$. We can describe a scalar point multiplication operation using G_q as $\tau.\psi$ denotes the repetitive additions of ψ in itself, where ψ ϵ G_q characterize an order n, provided n being smallest positive integer, furthermore $(n.\ \psi = O)$ holds as well.

2.3 Bio-hashing

The Bio-hashing function [30] is employed to gather biometric features of a person such as finger prints so that it can be used for purpose of authenticity. In 2004, Jin et al. [31]

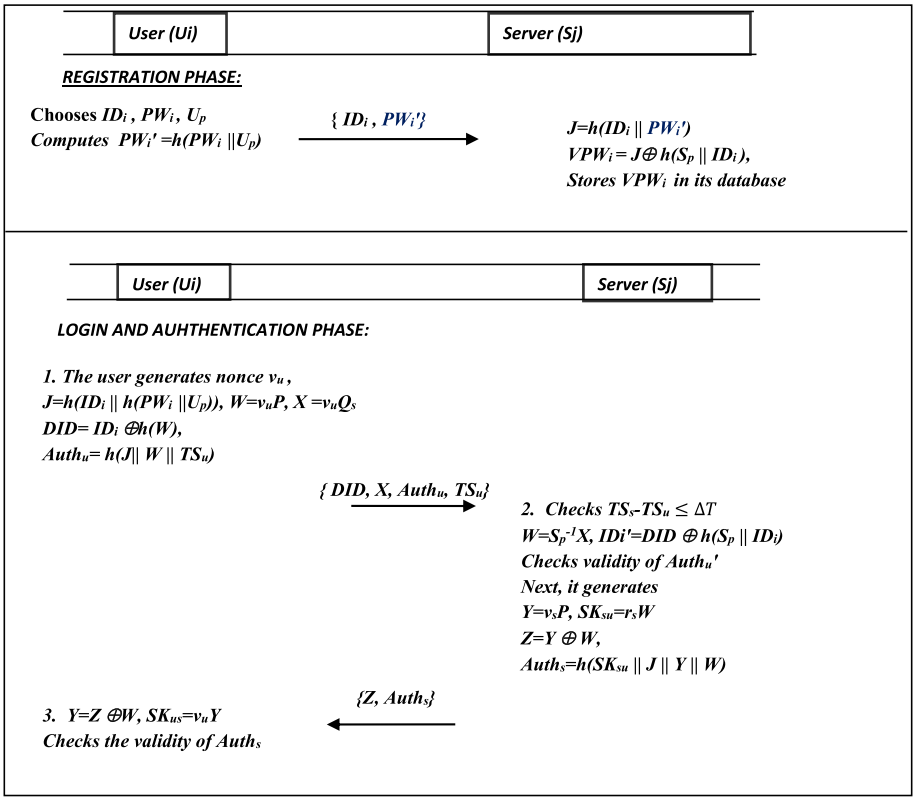

Fig. 1 Flow of registration, login and authentication procedures of Dongqing et al. model

demonstrated a two-factor authentication protocol for capturing fingerprint attributes for a particular user, and also engenders a tokenized pseudorandom number, which is then used to generate a compact code particular to some user, also known as bio-hashing. Thereafter, a more developed and worked Bio-hashing operation was demonstrated by Lumini et al. [30]. Actually, this Bio-hashing operation maps the user's oriented biometric properties on exclusive random vectors to construct a Biocode that discretizes projection coefficients, and then the resultant code could be remarked as a protected Bio-hashed password.

3 Revisiting and revewing Dongqing et al.

The design of Dongqing et al. protocol is explained in the following section.

3.1 Working of Dongqing et al. Scheme

There are three stages in the Dongqing et al.'s protocol [28] namely, registration procedure, login steps and the authentication procedure as shown in the Fig. 1. Some significant symbols employed in this protocol are mentioned in the Table 1 as given below.

Table 1 Notations

Symbols	Description
Ui, *Sj*, *RC*	ith user, jth server, registration centre
ID_i, *SIDj*	Ui's and server's identity
PWi, *BIOi*	User's password, User's biometric imprint
S_p	Sj's high entropy secret key
$Q_s = S_p P$	Sj's public key
Ts_u, Ts_s	Timestamps
$H(.)$	Bio-hashing operation
ΔT	Threshold for timestamp difference
SK_{ij}	A mutual session key constructed by Sj and Ui
$\|\|/\oplus$	concatenation and XOR functions

3.1.1 Procedure for Registration of Server

This scheme constitutes many service providers S_j, where $j = 1 \ldots \Phi$ and Φ represent the number of servers in the system. S_j generates a secret key S_p and public key $Q_s = S_p P$. The S_p is held secretly, while the public key is publicly accessible by all subscribers.

3.1.2 User Registration Phase

In user registration procedure, user is registered from Sj initially by selecting $\boldsymbol{ID_i}$, $\boldsymbol{PW_i}$ and $\boldsymbol{U_p}$. To proceed, it computes $\boldsymbol{PW_i' = h(PW_i \| U_p)}$ and submits $\{\boldsymbol{ID_i}, \boldsymbol{PW_i'}\}$ to server using secure channel. Thereafter, the server computes $\boldsymbol{J = h(ID_i \| PW_i')}$, $\boldsymbol{VPW_i = J \oplus h(S_p \| ID_i)}$ and stores $\boldsymbol{VPW_i}$ in its database to conclude the registration phase.

3.1.3 Mutual Authentication Procedure

1. In login phase, Ui generates a nonce $\boldsymbol{v_u}$ and calculates $\boldsymbol{J = h(ID_i \| h(PW_i \| U_p))}$, $\boldsymbol{W = v_u P}$, $\boldsymbol{X = v_u Q_s}$, $\boldsymbol{DID = ID_i \oplus h(W)}$ and $\boldsymbol{Auth_u = h(J \| W \| TS_u)}$ and submits the login request $\{\boldsymbol{DID}, \boldsymbol{X}, \boldsymbol{Auth_u}, \boldsymbol{TS_u}\}$ to server.
2. In authentication phase, the server computes the timestamp and compares the difference against the threshold, i.e. $\boldsymbol{TS_s - TS_u} \leq \Delta T$. If valid, then it additionally calculates $\boldsymbol{W = S_p^{-1} X}$, $\boldsymbol{IDi' = DID \oplus h(S_p \| ID_i)}$ and computes $\boldsymbol{Auth_u'}$ and verifies $\boldsymbol{Auth_u'}$. If positively verified, it checks the user's authenticity. On the other hand, it discards the message. Further, it generates $\boldsymbol{v_s}$ and computes $\boldsymbol{Y = v_s P}$, $\boldsymbol{SK_{su} = r_s W}$, $\boldsymbol{Z = Y \oplus W}$, $\boldsymbol{Auth_s = h(SK_{su} \| J \| Y \| W)}$ and sends the message $\{\boldsymbol{Z}, \boldsymbol{Auth_s}\}$ towards user.
3. The user computes $\boldsymbol{Y = Z \oplus W}$, $\boldsymbol{SK_{us} = v_u Y}$ and checks the validity of $\boldsymbol{Auth_s}$ parameter. It discards the message if the validity is not authenticated. Otherwise, validates the server and creates the session key as $\boldsymbol{SK_{us} = SK_{su}}$.

3.2 Weaknesses in Dongqing et al. Scheme

The limitations of Dongqing et al. scheme, which is found prone to privileged insider attack, denial of service threat as well as session specific temporary information

threats are described in this section. Besides, the scheme has a time synchronization problem that is difficult to implement in practical scenario. The limitations of Dongqing et al. scheme are described as below.

3.2.1 Privileged Insider Threat

In this threat, a malevolent insider in an organization may intercept the registration message contents and could manipulate it later for its malicious intentions. For instance, if the adversary (insider) gets the registration message contents, the former may initiate user impersonation attack through steps taken below:

1. Having access to ID_i and $PW_i' = h(PW_i \| U_p)$, the adversary may compute $J = h(ID_i \| h(PW_i \| U_p))$.
2. Next, it generates nonce v_a and further computes $W_a = v_a P$, $X_a = v_a Q_s$, $DID_a = ID_i \oplus h(W_a)$ and $Auth_a = h(J \| W_a \| TS_a)$ and sends the forged message $\{DID_a, X_a, Auth_a, TS_a\}$ to server.
3. On receiving the message, the server calculates the timestamp and compares the difference against the threshold, i.e. $TS_s - TS_a \leq \Delta T$. After finding it as true, the server further computes $W_a = S_p^{-1} X_a$, $ID_i = DID_a \oplus h(W_a)$, $J = VPW_i \oplus h(S_p \| ID_i)$, and ultimately $Auth_a'$ and could verify $Auth_a'$ as positive, however fake. In this manner, an insider adversary may forge server by impersonating as a user, comfortably.

3.2.2 Session Specific Ephemeral Secret-Leakage Threat

In this attack, if the temporary session parameters or variables are exposed to the attacker, the later could calculate the corresponding session key established between user and server [32]. In Dongqing et al. scheme, if the adversary is able to access the temporary session variables, the former may easily plan this attack by taking the following steps:

1. Assume, the adversary comes to know the temporary integer v_u, then it may compute $W = v_u P$ and onwards it may derive Y from Z by computing $Y = Z \oplus W$.
2. Next, the adversary may compute the shared session key SK_{us} by computing $SK_{us} = v_u Y$.

3.2.3 Denial of Service (DoS) Attack

In authentication protocols, where the user verifiers' repository is maintained on the end of server, an adversary may exploit this feature by repeatedly submitting fake requests. An attacker may replay the message $\{DID, X, Auth_u, TS_a\}$ with adding an updated timestamp TS_a, without modifying the other parameters DID, X, $Auth_u$.

Once the messages are received, the server computes the timestamp and compares the difference against the threshold, i.e. $TS_s\text{-}TS_a \leq \Delta T$. After finding it as true, it further computes $W = S_p^{-1} X$, $IDi' = DID \oplus h(S_p \| ID_i)$ and computes $Auth_u'$. Obviously, the verification of $Auth_u'$ shall fail since the timestamp is outdated in $Auth_u$. However, the adversary becomes successful in overburdening the server for computation with fake requests. Hence, the Dongqing et al. scheme is prone to Denial-of-service attack.

3.2.4 Time-Synchronization Problem

The Dongqing et al.'s protocol requires the strict clock-based time synchronization for the implementation of the protocol to avoid the replay attacks, which is however, considered as unrealistic in a practical scenario. The replay attacks could be better dealt with nonce-based methods that eliminate the stricter requirement of time synchronization.

4 Proposed Model

Our proposed protocol encompasses four stages. These stages include initialization stage, user registration, logic and authentication stage and password modification stage. These stages are illustrated as follows.

4.1 Initialization Procedure

The proposed protocol involves the participants such as user Ui and a trusted SIP server Sj. The user performs the registration process with Sj using a confidential channel. The Sj selects its master key S_p in this phase, that is used not only for registration purpose but also to verify the users in authentication phase. Next, it also constructs a public key $Q_s = S_pP$. The master key S_p is held secretly by the server, while its public key is publicly accessible by all subscribers.

4.2 Registration Phase

The registration of user with server is performed in this phase. Following steps are involved in the registration process.

1. The user selects IDi, PWi, U_p, a_1, and imprints BIOi on the biometric scanner. It calculates $PW_i' = h(PW_i \| U_p)$ and $J = h(ID_i \| PW_i') \oplus h(a_1)$. Next, it submits $\{ID_i, J\}$ to the service provider.
2. Once the server received the messages it computes $Q = J \oplus h(S_p \| ID_i)$ and store it in smart card (SC) and delivers to user by adopting a secured channel.
3. The user, then computes $R = Q \oplus h(a_1)$and replaces Q in smart card. It further calculates $R_1 = h(IDi \| PWi \| U_p)$, $R_2 = H(BIOi) \oplus U_p$ and stores R_1 and R_2 in smart card as well.

4.3 Mutual Authentication Procedure

1. To initiate the mutual authentication procedure for acquiring authenticated access to Sj's servies, Ui utilizes its SC. For this purpose, the user inputs its identity IDi, password PWi and stamps the biometric input $BIOi$ into the scanner. Then SC computes $\boldsymbol{U_p = H(BIOi) \oplus R_2}$, $\boldsymbol{R_1' = h(IDi \| PWi \| U_p)}$, and matches the equality for $\boldsymbol{R_1' = R_1}$, If true, then computes $\boldsymbol{PW_i' = h(PW_i \| U_p)}$. It, then generates random high entropy integers $\boldsymbol{v_u}$ and $\boldsymbol{n_1}$ and compute $\boldsymbol{h(S_p \| ID_i) = h(ID_i \| PW_i') \oplus R}$, $\boldsymbol{W = v_uP}$, $\boldsymbol{X = v_uQ_s}$, $\boldsymbol{DID = ID_i \oplus h(W)}$ and $\boldsymbol{Auth_u = h(h(S_p \| ID_i) \| W \| n_1)}$. In the end finally it forwards the message $\{\boldsymbol{DID}, \boldsymbol{X}, \boldsymbol{Auth_u}, \boldsymbol{n_1}\}$ to Sj for authentication.
2. Next, Sj receives parameters and computes $\boldsymbol{W = S_p^{-1}X}$, $\boldsymbol{IDi' = DID \oplus h(W)}$, $\boldsymbol{Auth_u' = h(h(S_p \| ID_i') \| W \| n_1)}$, and checks the validity of $\boldsymbol{Auth_u'}$. Next, it generates random integers $\boldsymbol{v_s}$, $\boldsymbol{n_2}$, and compute $\boldsymbol{Y = v_sP}$, $\boldsymbol{SK_{su} = h(v_sW \| h(S_p \| ID_i'))}$, $\boldsymbol{Z = Y \oplus W}$, $\boldsymbol{Auth_s = h(SK_{su} \| n_1 |}$

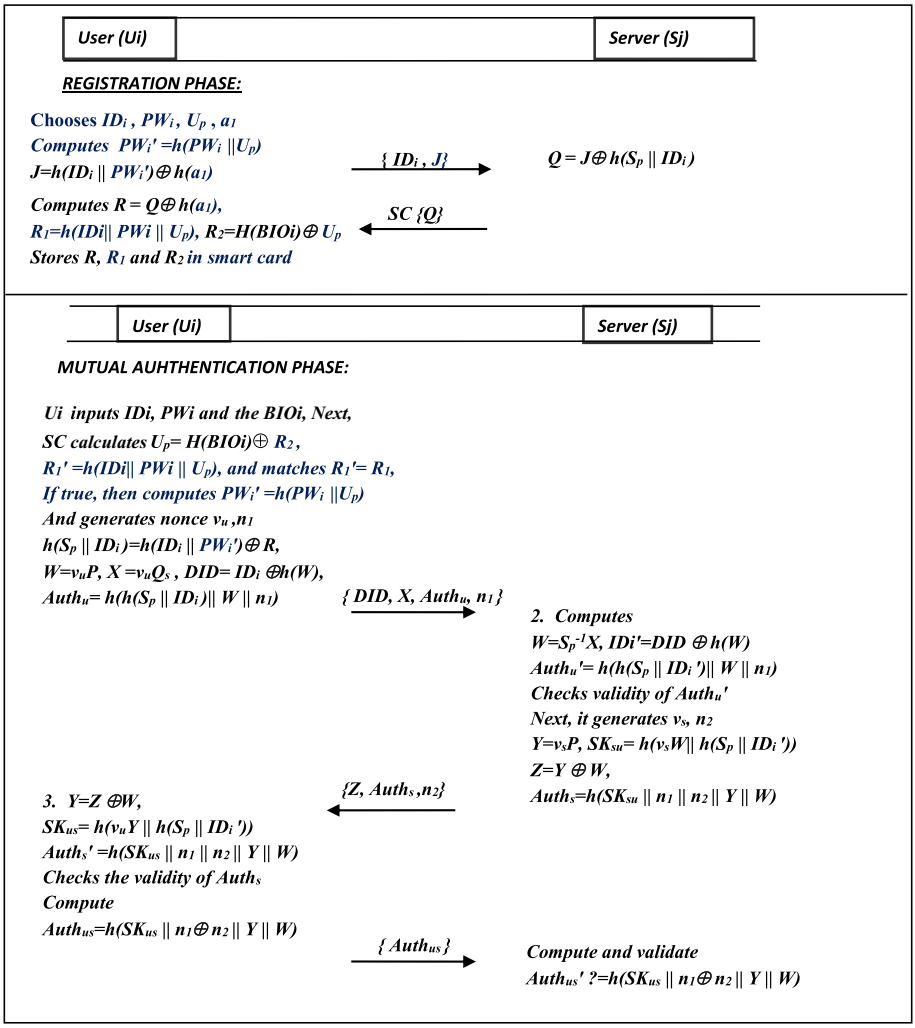

Fig. 2 Proposed authentication protocol

$\|n_2\|Y\|W$). Now it forwards the message $\{Z, Auth_s, n_2\}$ to user. The user further verify this message (Fig. 2).

3. Ui, after getting the message, computes $Y = Z \oplus W$, $SK_{us} = h(v_u Y \| h(S_p \| ID_i'))$ and $Auth_s = h(SK_{us} \| n_1 \| n_2 \| Y \| W)$. Then, it verifies the equality of $Auth_s$. If it holds true, then calculates $Auth_{us}' = h(SK_{us} \| n_1 \oplus n_2 \| Y \| W)$ and forwards $Auth_{us}$ to server for further procedures.
4. *Sj*, upon getting the message, computes $Auth_{us}' = h(SK_{us} \| n_1 \oplus n_2 \| Y \| W)$. Next, it monitors the equality match for $Auth_{us}' ? = Auth_{us}$. In case, the equality is proved to be true, it marks the user as valid for constructing the session key, or else, it aborts the session.

4.4 Password Modification Phase

Ui may update its password with a novel password i.e. PWi^{new} upon calling the specified procedure. Initially, the Ui inserts its smart card into the reader and captures by inputting the corresponding identity, password besides imprinting the biometric factor ($BIOi^*$) in biometric reader device. Next, the smart card constructs $\boldsymbol{U_p = H(BIOi) \oplus R_2}$, $\boldsymbol{R_1' = h(IDi \| PWi \| U_p)}$, and matches the equation for $\boldsymbol{R_1' = R_1}$, If true, then it allows the user to change the password by following the steps as given below:

1. The Ui computes $\boldsymbol{R_1^{new} = h(IDi \| PWi^{new} \| U_p)}$ by employing the new password $\boldsymbol{PWi^{new}}$.
2. Next, it calculates $\boldsymbol{h(S_p \| ID_i) = h(IDi \| h(PWi \| U_p)) \oplus R.}$
3. After deriving $h(S_p \| ID_i)$ it further calculates $\boldsymbol{R^{new} = h(IDi \| h(PWi^{new} \| U_p)) \oplus} h(S_p \| ID_i)$.
4. Next, it updates the parameters R_1 and R in smart card with R_1^{new} and R^{new}.

5 Security Analysis

This section discusses the security on informal terms, validates the security features on the basis of automated security tool, and verifies the properties of the contributed protocol using formal security analysis under Burrows-Abadi-Needham logic (BAN) as given under.

5.1 Informal Discussion on Protocol's Security

This sub-section presents the informal discussion on the security of the proposed protocol.

5.1.1 Replay Threats

Such threats can be launched if the attacker replays any eavesdropped or intercepted message to forge or misrepresent any legitimate participant. An adversary, upon intercepting the public messages $\{\boldsymbol{DID}, \boldsymbol{X}, \boldsymbol{Auth_u}, \boldsymbol{n_1}, \boldsymbol{Z}, \boldsymbol{Auth_s}, \boldsymbol{n_2}, \boldsymbol{Auth_{us}}\}$ could try to replay these messages on either of the sides to misrepresent the legal members in the protocol. However, Ui verifies the authenticity of Sj and dispels the probability of replay attack through calculating $Auths'$ and checking the equality for $\boldsymbol{Auth_s' ? = Auth_s'}$. The computation of $\boldsymbol{Auth_s'}$ needs a factor n_1, that is concatenated with other factors to evade this attack. Similarly, Sj could prevent this attack after calculating and checking the equality for $\boldsymbol{Auth_{us}' ? = Auth_{us}}$ in the third step of authentication protocol. The occurence of n_2 parameter in the calculation of $\boldsymbol{Auth_{us}}$. Hence the proposed scheme can prevent a reply attack.

5.1.2 Offline-Password Guessing Threat

This threat will be posed to the system if an attacker attempts to recover the user's password either by intercepting the content $\{\boldsymbol{DID}, \boldsymbol{X}, \boldsymbol{Auth_u}, \boldsymbol{n_1}, \boldsymbol{Z}, \boldsymbol{Auth_s}, \boldsymbol{n_2}, \boldsymbol{Auth_{us}}\}$ being

transmitted, or embezzle with the smart card factors $\{\boldsymbol{R}, \boldsymbol{R_1}, \boldsymbol{R_2}\}$. In these factors, $\boldsymbol{R_1}$ is generated by using a constituent $\boldsymbol{PWi}$, i.e. $\boldsymbol{R_1} = \boldsymbol{h}(\boldsymbol{IDi} \| \boldsymbol{PWi} \| \boldsymbol{U_p})$. An attacker cannot guess the password out of $\boldsymbol{R_1}$ unless it recovers the U_p factor from $\boldsymbol{R_2}$, which again depends upon the access of *BIOi* parameter. Therefore, the proposed protocol is resistant to offline password guessing threat.

5.1.3 Stolen Verifier Attacks

The information being stored on server's end could be exposed and the attacker can steal valuable information. If the server has repository of user-specific verifiers e.g. password or any other shared secret. The adversary may use it to masquerade as legitimate user it is called stolen verifier attack.

The proposed protocol unlike Dongqing et al. protocol does not maintain any sort of repository of verifiers on the side of Sj server that helps to rule out the possibility of this attack.

5.1.4 Stolen Smart Card Threat

An attacker may get the smart card contents and attempt to misuse those contents for launching any guessing attack.

After stealing the smart card, the adversary might attempt to embezzle with the recovered data. Nevertheless, as proved in Sect. 5.1.2, the attacker may not guess PWi from SC factors $\{\boldsymbol{R}, \boldsymbol{R_1}, \boldsymbol{R_2}\}$. Therefore, notwithstanding with the stolen SC parameters, that adversary cannot launch any forgery attack for not having biometric parameter $\boldsymbol{BIOi}$.

5.1.5 Session Key Confidentiality

This security characteristic advocates that the established session key (*SK*) must be held with the legal session members, i.e. Ui or Sj, and not others.

In the proposed model, the *SK* is produced by calculating $\boldsymbol{SK_{su}} = \boldsymbol{SK_{us}} = \boldsymbol{h}(\boldsymbol{v_u Y} \| \boldsymbol{h}(\boldsymbol{S_p} \| \boldsymbol{ID_i'}))$. For generating a legitimate *SK* the adversary needs $\boldsymbol{v_u}$ and *BIOi* parameters for accessing $\boldsymbol{h}(\boldsymbol{S_p} \| \boldsymbol{ID_i'})$, besides getting the smart card contents. The $\boldsymbol{v_u}$ is a high entropy integer, and cannot be guessed in polynomial amount of time, and the construction of $\boldsymbol{v_u Y}$ is bounded by ECDLP. Similarly, the unavailability of *BIOi* parameter to the adversary leads to the protection of session key *SK*, and cannot be computed until the above parameters are accessed.

5.1.6 Known-Key Security

The compliance to this feature of security entails the protection of private secret keys of concerned session participants, in case, the current session key *SK* is compromised.

In proposed protocol, if the session key $\boldsymbol{SK_{su}} = \boldsymbol{SK_{us}} = \boldsymbol{h}(\boldsymbol{v_u Y} \| \boldsymbol{h}(\boldsymbol{S_p} \| \boldsymbol{ID_i'})$ is revealed by mistake, the adversary might not guess user's password $\boldsymbol{PWi}$ or the master secret key $\boldsymbol{S_p}$ of server. Therefore the proposed scheme is well secured for the known key security.

5.1.7 Perfect Forward Secrecy

This attribute of security ensures the confidentiality regarding session keys, assuming the high entropy private key of either user or server is exposed to the adversary.

Our protocol complies with perfect forward secrecy, notwithstanding the fact, that the long-term and high entropy secrets of participating members are exposed. That is, if the server's master key S_p is revealed, an attacker might not calculate previous session keys due to short of knowledge regarding $v_u Y$ in a session key $SK_{us} = h(v_u Y \| h(S_p \| ID_i'))$.

5.1.8 Mutual Authentication

This property makes certain that the interacting members must authenticate mutually one another in the same authentication scheme.

The contributed protocol complies with this property for both members. An atttacker after intercepting the open content of the communication messages $\{DID, X, Auth_u, n_1, Z, Auth_s, n_2, Auth_{us}\}$ may attempt to change or replay the content towards both ends for deceiving the legal members. Nonetheless, the concerned participants verify the authenticity of each other, and annul the chances for possible modification or replaying the content after calculating and checking the equations $Auth_{us}'? = h(SK_{us} \| n_1 \oplus n_2 \| Y \| W)$ and $Auth_s'? = h(SK_{us} \| n_1 \| n_2 \| Y \| W)$. Hence, in our scheme both of the members can mutually authenticate one another.

5.1.9 Anonymous Authentication

This security feature warrants privacy or anonymity to the user during its interaction with the server in login and authenticated key agreement phase. An adversary may not produce the original identities of the communicants on employing the eavesdropped messages.

In contributed scheme, Ui submits its dynamic identity DID in the form of $DID = ID_i \oplus h(W)$ after having computed the factor W. The adversary might not get the Ui's identity IDi from DID, until it gets access to server's master key S_p and compute W from X. Therefore, this scheme provides sufficient anonymity to the user.

5.1.10 Privileged Insider Threat

A malevolent insider might intercept the contents of registration query as submitted by the user during registration phase. In contributed protocol, we employed a random number a_1 to encrypt $h(ID_i \| PW_i')$ parameter. As a result, the malicious insider, after encryption is unable to derive $h(ID_i \| PW_i')$ from J due to that encryption. The server again encrypts the same with $h(S_p \| ID_i)$ and submits the smart to user after storing the result in it. The user finally decrypts the same using a_1 and recovers the result. In this manner, a malevolent insider might not be able to recover any secret parameter from the registration request and hence, the contributed scheme is resistant to malicious insider threat.

5.1.11 Session-Specific Ephemeral Secret-Leakage Attack

If session-specific ephemeral integers are exposed, an attacker could attempt to compute session keys. Nonetheless, contrary to Dongqing et al., the proposed protocol is resistant to

```
(**************** Channels ****************)
free Sec_Ch:channel [private].   (*Secure Channel*)
free Pub_Ch: channel.      (*Public Channel*)
(********** Constants & Variables **********)
const P:bitstring.
free IDi:bitstring.
free PWi:bitstring [private].
free a1:bitstring [private].
free Up:bitstring [private].
free Sp:bitstring [private].
free BIOi:bitstring [private].
(********** Constructor **********)
fun h(bitstring):bitstring.
fun XOR(bitstring,bitstring):bitstring.
fun CONCAT(bitstring,bitstring):bitstring.
fun ECPM(bitstring,bitstring):bitstring.
fun INVERSE(bitstring):bitstring.
(********** Destructors & Equations **********)
equation forall a:bitstring,b:bitstring; XOR(XOR(a,b),b)=a.
equation forall c:bitstring; INVERSE(INVERSE(c))=c.
(**************** Events ****************)
event begin_User_U(bitstring).
event end_User_U(bitstring).
event begin_Server_S(bitstring).
event end_Server_S(bitstring).
(******************** Queries ********************)
free SK:bitstring [private].
query attacker(SK).
query id:bitstring; inj-event(End_User_U(id)) ==> inj-event(Begin_User_U(id)) .
query id:bitstring; inj-event(End_Server_S(id)) ==> inj-event(Begin_Server_S(id)) .
```

Fig. 3 Channels, constructor, destructor, events and equations

this threat. This is due to the fact that session key $SK_{su} = SK_{us} = h(v_u Y \| h(S_p \| ID_i'))$ could be calculated, if the attacker might approach both $v_u Y$ and $h(S_p \| ID_i')$ parameters. Even, if the v_u parameter is leaked to the adversary, it may compute $v_u P$, however, it may not access the other parameter, which can only be computed using ***BIOi*** biometric value. Therefore, the presented scheme is protected from temporary information threat.

5.2 Automated Security Verification

ProVerif [33, 34] is one of the widely recognized automated protocol-verifier as adopted by most researchers in the current protocols. Proverif is employing applied π calculus rules in order to verify the protocols implementing encryption, hash, and Diffie–Helman operations etc. We also used this tool for testing the security strength of our contributed scheme.

We begin with the verification and testing procedure through identifying two communication channels, i.e., a private channel Sec_Ch and a public channel Pub_Ch between participants. The channels, constants and variables, constructor & de-constructor, equations, events and queries as used in the Proverif simulation of proposed model, is shown in Fig. 3.

The two events have been modeled between user and server. The initiating and ending event for the user are begin_User_U(bitstring) and End_User_U(bitstring). Similarly, these events for the server are Begin_Server_S(bitstring) and End_Server_S(bitstring). We have

Fig. 4 UserUi process

```
(****************** User (Ui) ******************)
let UserUi=
(*****Registration *****)
new a1:bitstring;
new Up:bitstring;
let PWi'=h(CONCAT(PWi, Up)) in
let J=XOR(h(CONCAT(IDi, PWi')), h(a1)) in
out (Sec_Ch,(IDi, J));
in(Sec_Ch,(xQ:bitstring));
let R = XOR(Q, h(a1)) in
let R1=h(CONCAT(IDi, PWi, Up)) in
let R2=XOR(H(BIOi), Up) in
 (***** Login and Authentication *****)
event begin_User_U(IDi);
let Up = XOR(H(BIOi), R2) in
let R1'=h(CONCAT(IDi, PWi, Up)) in
if (R1'= Ri) then
let PWi'=h(CONCAT(PWi, Up)) in
new vu: bitstring;
new n1: bitstring;
let h(CONCAT(Sp, IDi))=XOR(h(CONCAT(IDi, PWi')), R) in
let W=ECPM(vu, P) in
let X=ECPM (vu, Qs) in
let DID=XOR(IDi, h(W)) in
let Authu=h(CONCAT(h(Sp, IDi), W, n1)) in
out (Pub_Ch,(DID, X, Authu, n1));
in(Pub_Ch,(xZ:bitstring, xAuths:bitstring, xn2:bitstring));
let Y=XOR(xZ, W) in
let SKus=h(CONCAT(ECPM(vu, Y), h(CONCAT(Sp, IDi')) in
let Auths'= h(CONCAT(SKus, n1, n2, Y, W)) in
if Auths'=Auths then
let Authus=h(CONCAT(SKus, XOR(n1, n2), Y, W)) in
out (Pub_Ch(Authus));
event End_User_U(IDi)
else
0.
```

described two separate procedures, i.e., User_U and Server_S for modelling user and server processes, respectively. The process User_U submits the calculated parameters IDi, PWi′, Up, a1 using secure channel Sec_Ch towards Server_S. Then, after receiving the registration request, the User_U process further computes Q and forwards to user. The user calculates R1, R2 and stores in smart card. In mutual authentication procedure, the User_U process compares Ri and Ri′ after computing Ri′. It further calculates PWi′, W, X, DID and Authu. Then, it submits {DID, X, Authu, n1} towards Server_S using Pub_Ch. Next, it receives {xZ, xAuths, xn2} from Server_S. It calculates Y, SKus, Auths' and compares Auths and Auths'. Finally, it submits Authus towards Server_S for verification, and proceeds for calculating the session key SK as shown in Fig. 4. Likewise, the Server_S process receives xIDi, xJ from User_U process as registration request. Next, it computes Q and submits to User_U utilizing secure channel Sec_Ch. In mutual authentication phase, the Server_S process receives {xDID, xX, xauthu, xn1} and computes W, IDi', Authu' and compares Authu' with Authu. If positive, then computes Y, SKsu, Z and Auths and submits {Z, Auths, n2} to User_U using Pub_Ch. Further, it receives xAuthus from the same process, and computes Authus'. Next, it validates the user

```
(******************** Server (Sj) ********************)
 let Server_S=

(****** Registration ******)
in(Sec_Ch,(xIDi: bitstring, xJ:bitstring));
let Q=XOR(J, h(CONCAT(Sp, IDi))) in
out(Sec_Ch,(Q));
(****** Login and Authentication ******)
event Begin_Server_S(Qs);
in(Pub_Ch,(xDID:bitstring, xX:bitstring, xAuthu:bitstring, xn1:bitstring));
let W=MULT(INVERSE(Sp), xX) in
let IDi'=XOR(xDID, h(W)) in
let Authu'=h(CONCAT(h(CONCAT(Sp, IDi')), W, xn1)) in
if (Authu' = xAuthu) then
new vs:bitstring;
new n2:bitstring;
let Y = ECMP(vs, P) in
let SKsu=h(CONCAT(ECMP(vs, W), h(CONCAT(Sp, IDi')))) in
let Z=XOR(Y, W) in
let Auths=h(CONCAT(SKsu, xn1, n2, Y, W)) in
out(Pub_Ch,(Z,Auths,n2));
in(Pub_Ch,(xAuthus:bitstring);
let Authus'=h(CONCAT(SKus, XOR(xn1, n2), Y, W)) in
if Authus' =xAuthus then
event End_Server_S(Qs)
else
0.
```

Fig. 5 ServerSj process

on matching the two parameters Authus' and xAuthus. Otherwise, it aborts the protocol, as shown in Fig. 5.

The two participants may interact by establishing many sessions, so these two processes are deemed to be in replication as illustrated below.

process

((!User_U) | (!Server_S))

We get to the understated results after applying queries for this simulation.

$$\text{RESULT inj} - \text{event(End\_Server\_S(id))} = => \text{inj} - \text{event(Begin\_Server\_S(id))} \text{ is true.} \tag{2}$$

$$\text{RESULT inj} - \text{event(End\_User\_U(id\_1683))} = => \text{inj} - \text{event(begin\_User\_U(id\_1683))} \text{ is true.} \tag{3}$$

$$\text{RESULT not attacker(SK[])} \text{ is true.} \tag{4}$$

The results from Eqs. (2) and (3) depict that both processes initiated and terminated successfully, while the result in Eq. (4) suggests that the attacker query could not expose the session key as constructed between the processes in mutual authentication procedure.

5.3 Formal Security Analysis (BAN Logic)

This formal analysis section presents the analysis on security employing Burrows-Abadi-Needham logic (BAN) logic [35] and random oracle model (ROM). The BAN logic analyzes the security aspects with a focus on mutual authentication and the robustness of computed session key between the communicants. We define the following terms to promote the understanding of readers before describing BAN logic.

Principals acts as participating agents in this model.

Keys are meant for symmetric-encryption.

Nonces be the non-repeatable parts of the forwarded content.

Some further notations that are employed in the BAN logic analysis are stated below:

$\phi|\equiv \Upsilon$:ϕ believes Υ.
$\phi \lhd \Upsilon$:ϕ sees Υ.
$\phi|\sim \Upsilon$:ϕ once said Υ.
$\phi \Rightarrow \Upsilon$:$\phi$ has got jurisdiction over Υ;
$\sharp(\Upsilon)$: The message Υ is fresh.
$(\Upsilon)_{Z}$: The formulae Υ is used in arrangement with formulae Υ.
(Υ, Z): Υ or Z represent a component of the message (Υ, Z).
$(\Upsilon, Z)_K$: Υ or Z is encrypted using key K.
$\phi \overset{K}{\longleftrightarrow} \phi'$:$\phi$ andϕ' may secretly contact through shared key K.
$\langle \Upsilon, Z \rangle_K$: Υ or Z is hashed with key K.

Some of the logical rules are employed in this proof as listed below:

$R1$: Message meaning rule: $\dfrac{\phi|\equiv \phi \overset{K}{\leftrightarrow} \phi',\ \phi \lhd \langle \Upsilon \rangle_Z}{\phi|\equiv \phi'|\sim \Upsilon}$

$R2$: Nonce verification rule: $\dfrac{\phi|\equiv \sharp(\Upsilon),\ \phi|\equiv \phi'|\sim\ \Upsilon}{\phi|\equiv \phi'|\equiv\ \Upsilon}$

$R3$: Jurisdiction rule: $\dfrac{\phi|\equiv \phi' \Rightarrow \Upsilon,\ \phi|\equiv \phi'|\equiv\ \Upsilon}{\phi|\equiv \Upsilon}$

$R4$: Freshness conjuncatenation rule: $\dfrac{\phi|\equiv \sharp(\Upsilon)}{\phi|\equiv \sharp(\Upsilon,\ Z)}$

$R5$: Belief rule: $\dfrac{\phi|\equiv(\Upsilon),\ \phi|\equiv(Z)}{\phi|\equiv(\Upsilon,\ Z)}$

$R6$: Session keys rule: $\dfrac{\phi|\equiv \sharp(\Upsilon),\ \phi|\equiv \phi'|\equiv\ \Upsilon}{\phi|\equiv \phi \overset{K}{\leftrightarrow} \phi'}$

Our contributed protocol must achieve the understated objectives or goals to support the security attributes by employing BAN logic.

$$\mathbf{G1}\text{: Sj}| \equiv \text{Ui} \overset{\boldsymbol{SKsu}}{\longleftrightarrow} \text{Sj}$$

$$\mathbf{G2}\text{: Sj}| \equiv \text{Ui}| \equiv \text{Ui} \overset{\boldsymbol{SKsu}}{\longleftrightarrow} \text{Sj}$$

$$\mathbf{G3}\text{: Ui}| \equiv \text{Ui} \overset{\boldsymbol{SKsu}}{\longleftrightarrow} \text{Sj}$$

$$\mathbf{G4}\text{: Ui}| \equiv \text{Sj}| \equiv \text{Ui} \overset{\boldsymbol{SKsu}}{\longleftrightarrow} \text{Sj}$$

First, we convert the communicated message contents into idealized form as shown underneath:

M_1: Ui $\rightarrow$ Sj: $\boldsymbol{DID}, \boldsymbol{X}, \boldsymbol{Auth_u}, \boldsymbol{n_1}$: $\{\langle IDi\rangle_{h(W)}, \boldsymbol{v_uQ_s}, \langle \boldsymbol{h(S_p \| ID_i)}, \boldsymbol{n_1}\rangle_W, \boldsymbol{n_1}\}$
M_2: Sj $\rightarrow$ Ui: $\boldsymbol{Z}, \boldsymbol{Auth_s}, \boldsymbol{n_2}$: $\{\boldsymbol{Z}, \langle \boldsymbol{h(v_s\, v_uP \| h(S_p \| ID_i'))}, \boldsymbol{n_1}, \boldsymbol{n_2}\rangle_{Y,W}, \boldsymbol{n_2}\}$
M_3: Ui $\rightarrow$ Sj: $\boldsymbol{Auth_{us}}$: $\langle \boldsymbol{h(v_u\, v_sP \| h(S_p \| ID_i'))}, \boldsymbol{n_1 \oplus n_2}\rangle_{Y,W}\}$

The subsequent assumptions are built to verify the security features of our protocol.

§1: Ui $\models \sharp n_1$
§2: Sj $\models \sharp n_2$
§3: Ui $\models$ Sj $\overset{(SK_{us},W,Y)}{\longleftrightarrow}$ Ui
§4: Sj $\models$ Sj $\overset{(SK_{us},W,Y)}{\longleftrightarrow}$ Ui
§5: Ui $\models$ Sj $\models$ Ui $\overset{(SK_{us},W,Y)}{\longleftrightarrow}$ Sj
§6: Sj $\models$ Ui $\models$ Ui $\overset{(SK_{us},W,Y)}{\longleftrightarrow}$ Sj
§7: Ui $\models$ Sj $\Rightarrow \boldsymbol{v_sP}$
§8: Sj $\models$ Ui $\Rightarrow \boldsymbol{v_uP}$

Thirdly, the developed idealized forms such as M_1, M_2 and M_3 of this model may be further utilized by employing the above postulates.

Following derivations are obtained by the above notations, idealization and the premises.

Considering the idealized forms, i.e. M1 and M3:

$\mathbf{M_1}$: Ui $\rightarrow$ Sj: $\boldsymbol{DID}, \boldsymbol{X}, \boldsymbol{Auth_u}, \boldsymbol{n_1}$: $\{\langle IDi\rangle_{h(W)}, \boldsymbol{v_uQ_s}, \langle \boldsymbol{h(S_p \| ID_i)}, \boldsymbol{n_1}\rangle_W, \boldsymbol{n_1}\}$
$\mathbf{M_3}$: Ui $\rightarrow$ Sj: $\boldsymbol{Auth_{us}}$: $\langle \boldsymbol{h(v_u\, v_sP \| h(S_p \| ID_i'))}, \boldsymbol{n_1 \oplus n_2}\rangle_{Y,W}\}$

By applying the seeing rule, we have

Q1: Sj $\lhd \boldsymbol{DID}, \boldsymbol{X}, \boldsymbol{Auth_u}, \boldsymbol{n_1}$: $\{\langle IDi\rangle_{h(W)}, \boldsymbol{v_uQ_s}, \langle \boldsymbol{h(S_p \| ID_i)}, \boldsymbol{n_1}\rangle_W, \boldsymbol{n_1}\}$
Q2: Sj $\lhd \boldsymbol{Auth_{us}}$: $\langle \boldsymbol{h(v_u\, v_sP \| h(S_p \| ID_i'))}, \boldsymbol{n_1 \oplus n_2}\rangle_{Y,W}\}$

Now using Q1, Q2, §3 and $R1$, we have

Q3: Sj $\models$ Ui $\sim \{\langle IDi\rangle_{h(W)}, \boldsymbol{v_uQ_s}, \langle \boldsymbol{h(S_p \| ID_i)}, \boldsymbol{n_1}\rangle_W, \boldsymbol{n_1}\}$
Q4: Sj $\models$ Ui $\sim \langle \boldsymbol{h(v_u\, v_sP \| h(S_p \| ID_i'))}, \boldsymbol{n_1 \oplus n_2}\rangle_{Y,W}\}$

Referring Q3, Q4, §1, $R4$ and $R2$, we have

Q5: Sj $\models$ Ui $\models \{\langle IDi\rangle_{h(W)}, \boldsymbol{v_uQ_s}, \langle \boldsymbol{h(S_p \| ID_i)}, \boldsymbol{n_1}\rangle_W, \boldsymbol{n_1}\}$
Q6: S $\models$ Ui $\models \langle \boldsymbol{h(v_u\, v_sP \| h(S_p \| ID_i'))}, \boldsymbol{n_1 \oplus n_2}\rangle_{Y,W}\}$

Referring Q5, Q6, §4, §8 and $R3$, we get

Q7: Sj $\models \{\langle IDi\rangle_{h(W)}, \boldsymbol{v_uQ_s}, \langle \boldsymbol{h(S_p \| ID_i)}, \boldsymbol{n_1}\rangle_W, \boldsymbol{n_1}\}$
Q8: Sj $\models \langle \boldsymbol{h(v_u\, v_sP \| h(S_p \| ID_i'))}, \boldsymbol{n_1 \oplus n_2}\rangle_{Y,W}\}$

Using Q7, Q8, §4, $(\boldsymbol{SK_{su}} = \boldsymbol{SK_{us}} = \boldsymbol{h(v_uY \| h(S_p \| ID_i'))}$ and $R6$, we get

Q9: Sj $\models$ Ui $\overset{\boldsymbol{SKsu}}{\longleftrightarrow}$ Sj ($\mathbf{G1}$)

According to Q9, §6 we apply *R*6 as

Q10: Sj $|\equiv$ Ui $|\equiv$ Ui $\overset{SKsu}{\longleftrightarrow}$ Sj (**G2**)

Next, again visualizing the idealized form M2:

$\mathbf{M_2}$: Sj $\rightarrow$ Ui: $\boldsymbol{Z}$, $\boldsymbol{Auth_s}$, $\boldsymbol{n_2}$: $\{\boldsymbol{Z}, \langle \boldsymbol{h}(\boldsymbol{v_s}\ \boldsymbol{v_u P} \| \boldsymbol{h}(\boldsymbol{S_p} \| \boldsymbol{ID_i}')), \boldsymbol{n_1}, \boldsymbol{n_2}\rangle_{Y,W}, \boldsymbol{n_2}\}$

By applying again the seeing rule, we have

Q11: Ui $\triangleleft \boldsymbol{Z}$, $\boldsymbol{Auth_s}$, $\boldsymbol{n_2}$: $\{\boldsymbol{Z}, \langle \boldsymbol{h}(\boldsymbol{v_s}\ \boldsymbol{v_u P} \| \boldsymbol{h}(\boldsymbol{S_p} \| \boldsymbol{ID_i}')), \boldsymbol{n_1}, \boldsymbol{n_2}\rangle_{Y,W}, \boldsymbol{n_2}\}$

According to Q11, §4 and *R*1, we have

Q12: Ui $|\equiv$ Sj $\sim \{\boldsymbol{Z}, \langle \boldsymbol{h}(\boldsymbol{v_s}\ \boldsymbol{v_u P} \| \boldsymbol{h}(\boldsymbol{S_p} \| \boldsymbol{ID_i}')), \boldsymbol{n_1}, \boldsymbol{n_2}\rangle_{Y,W}, \boldsymbol{n_2}\}$

Using Q12, §2, *R*4 and *R*2, we have

Q13: Ui $|\equiv$ Sj $|\equiv \{\boldsymbol{Z}, \langle \boldsymbol{h}(\boldsymbol{v_s}\ \boldsymbol{v_u P} \| \boldsymbol{h}(\boldsymbol{S_p} \| \boldsymbol{ID_i}')), \boldsymbol{n_1}, \boldsymbol{n_2}\rangle_{Y,W}, \boldsymbol{n_2}\}$

Referring Q13, §3, §7 and *R*3, we get

Q14: Ui$\equiv \{\boldsymbol{Z}, \langle \boldsymbol{h}(\boldsymbol{v_s}\ \boldsymbol{v_u P} \| \boldsymbol{h}(\boldsymbol{S_p} \| \boldsymbol{ID_i}')), \boldsymbol{n_1}, \boldsymbol{n_2}\rangle_{Y,W}, \boldsymbol{n_2}\}$

From Q14, §3, ($\boldsymbol{SK_{su}} = \boldsymbol{SK_{us}} = \boldsymbol{h}(\boldsymbol{v_u Y} \| \boldsymbol{h}(\boldsymbol{S_p} \| \boldsymbol{ID_i}'))$), and *R*6, we get

Q15: Ui $|\equiv$ Ui $\overset{SKus}{\longleftrightarrow}$ Sj (**G3**)

According to Q15, §5, we apply *R*6 as

Table 2 Functionality comparison of multi-server schemes

	Zhang et al. [17]	Chaudhary et al. [5]	Dongqing et al. [28]	Ours
Anonymity	×	√	√	√
Resists privileged insider threat	√	√	×	√
Mutual authentication	√	√	√	√
Resists stolen smart card threat	√	√	√	√
Resists replay attack	√	√	√	√
Resists offline password guessing threat	√	√	√	√
Resists session specific temporary information threat	√	√	×	√
Resists user impersonation threat	×	√	√	√
Resistant to session key threat	×	×	√	√
Resists denial-of-service threat	√	√	×	√
No strict time synchronization required	√	√	×	√
Perfect forward secrecy	√	√	√	√

Table 3 Computational comparison

	Zhang et al. [17]	Chaudhary et al. [5]	Dongqing et al. [28]	Ours
Authentication messages	$10T_H+6T_{ESM}$	$7T_H+6T_{ESM}$	$9T_H+6T_{ESM}$	$13T_H+6T_{ESM}$
Delay (ms)	13.379	13.372	13.376	13.3859

Q16: $\text{Ui} \mid\equiv \text{Sj} \mid\equiv \text{Ui} \overset{SKus}{\longleftrightarrow} \text{Sj}$ (**G4**)

The demonstrated analysis of the BAN logic firmly proves that our proposed protocol follows mutual authentication while the constructed session key is agreed and shared mutually between participants (Sj and Ui).

6 Performance Evaluation Analysis

In this section, we evaluate and compare the security of the contributed protocol with Dongqing et al.'s SIP authentication scheme and other existing protocols. The Table 2 depicts the comparison of various protocols regarding immunity of threats, which specifies that the proposed scheme as a robust authentication scheme against Dongqing et al. The comparison as depicted in Table 2 comprises Dongqing et al. [28], Chaudhary et al. [5], Zhang et al. [25], and proposed scheme, which portrays that our protocol is immune to attacks than its contemporary schemes as indicated. Although the protocol bears a little extra cost in comparison with [5, 25, 28] schemes, however it is secure against many threats notably replay attack, offline-password guessing attack, privileged insider attack, denial of service attack, session specific ephemeral secret-leakage attack, and session key attack. The extra cost of proposed scheme is in terms of few more hash operations, that does not adds much to the cost, however the proposed scheme becomes resilient to attacks as posed to earlier schemes.

To compare the computational overhead in Table 3, we indicate one-way hash function with T_H and elliptic scalar point multiplication T_{ESM}, and ignoring the lightweight XOR operation due to negligible overhead. The computational cost of Zhang et al., Chaudhary et al., Dongqing et al.' scheme and proposed scheme amounts to $10T_H+6T_{ESM}$, $7T_H+6T_{ESM}$, $9T_H+6T_{ESM}$ and $13T_H+6T_{ESM}$ with computational delays amounting to 13.379 ms, 13.372 ms, 13.376 ms, 13.859 ms, respectively. Most of these protocols utilize 6 scalar point multiplications, but the number of hash operations varies. Although, there is little difference in computational cost of these protocols, however the resistance to attacks varies with each protocol. For instance, the proposed scheme is resistant to all attacks, while Dongqing et al.'s scheme is prone to Privileged insider attack and session-specific ephemeral secret-leakage threat. The Chaudhary et al.'s protocol is found to be vulnerable for session key attack, and Zhang et al. does not offer anonymity feature to user, and is also prone to impersonation attack.

The scalar point operation could be the decisive factor for measuring the efficiency of a protocol. The Lin et al. takes 4 T_{ESM} operations, while the proposed scheme takes 6 T_{ESM} operations. Although, Lin et al. takes two less point multiplication operations as compared to proposed protocol, however, the later is resistant to many attacks that Lin et al. scheme doesn't. Thus, in the light of above performance evaluation analysis shown in Tables 2 and

3, we can say the proposed protocol is a more secure multi-server authentication protocol than Lin et al., with a bit added computational cost than its counterpart.

7 Conclusion

The SIP protocol provides IMS structural framework the basis for the maintenance of voice and multimedia based sessions. Recently, Dongqing et al. discovered limitations in Lu et al. and Chaudhary et al.'s SIP authentication protocols, and demonstrated an improved SIP authentication protocol. In this work, we elaborated that Dongqing et al.'s scheme is still prone to privileged insider attack, denial of service (DoS) attack, and session specific ephemeral secret-leakage attacks, other than a limitation of time synchronization. Thus, to counter the limitations in Dongqing et al., we propose an improved SIP authentication protocol which is formally proved as secure using BAN logic analysis in the preceding sections. The comparative analysis of proposed and contemporary schemes depicts the supremacy of contributed protocol in terms of security and efficiency.

References

1. 3rd Generation Partnership Project: Technical Specification Group Services and System Aspects; IP multimedia subsystem (IMS). 3GPP TS 23.228 V11.4.0 (2012).
2. Poikselkä, M., Niemi, A., Khartabil, H., & Mayer, G. (2007). *The IMS: IP multimedia concepts and services* (2nd Edn.). ISBN: 978-0-470-03183-4.
3. Arkko, J., Torvinen, V., Camarillo, G., Niemi, A., & Haukka, T. (2003). Security mechanism agreement for the session initiation protocol (sip). *Cognitiva, 12*(1), 37–61.
4. Salsano, S., Veltri, L., & Papalilo, D. (2002). *SIP security issues: The SIP authentication procedure and its processing load*. Piscataway: IEEE Press.
5. Chaudhry, S. A., Naqvi, H., Sher, M., Farash, M. S., & Hassan, M. U. (2015). An improved and provably secure privacy preserving authentication protocol for sip. *Peer-to-Peer Networking and Applications, 10,* 1–15.
6. Yi, P. L., & Wang, S. S. (2010). A new secure password authenticated key agreement scheme for sip using self-certified public keys on elliptic curves. *Computer Communications, 33*(3), 372–380.
7. Thomas, M. (2001). *SIP security requirements*. IETF Intemet dren (draftthomas-sip-sec-reg'OO. txt).
8. Yoon, E. J., Shin, Y. N., Il, S. J., & Yoo, K. Y. (2010). Robust mutual authentication with a key agreement scheme for the session initiation protocol. *IETE Technical Review, 27*(3), 203–213.
9. Leach, P. J., Franks, J., Luotonen, A., Hallam-Baker, P. M., Lawrence, S. D., Hostetler, J. L., & Stewart, L. C. (1999). HTTP authentication: Basic and digest access authentication.
10. Yang, C. C., Wang, R. C., & Liu, W. T. (2005). Secure authentication scheme for session initiation protocol. *Computers & Security, 24*(5), 381–386.
11. Denning, D. E., & Sacco, G. M. (1981). Timestamps in key distribution systems. *Communications of the ACM, 24*(8), 533–536.
12. He, D., Chen, J., & Chen, Y. (2012). A secure mutual authentication scheme for session initiation protocol using elliptic curve cryptography. *Security and Communication Networks, 5*(12), 1423–1429.
13. Durlanik, A., & Sogukpinar, I. (2005). Sip authentication scheme using ecdh. *Screen, 137,* 3367.
14. Liufei, W., Zhang, Y., & Wang, F. (2009). A new provably secure authentication and key agreement protocol for sip using ecc. *Computer Standards & Interfaces, 31*(2), 286–291.
15. 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. *Computer Communications, 33*(14), 1674–1681.
16. Gokhroo, M. K., Jaidhar, C. D., & Tomar, A. S. (2011). Cryptanalysis of sip secure and efficient authentication scheme. In: *IEEE international conference on communication software and networks*, pp. 308–310.
17. Pu, Q. (2010). Weaknesses of SIP authentication scheme for converged VoIP networks. *IACR Cryptol* ePrint Arch, 464.

18. Jia, L. T. (2009). Efficient nonce-based authentication scheme for session initiation protocol. *International Journal of Network Security, 8*(1), 12–16.
19. Arshad, R., & Ikram, N. (2013). Elliptic curve cryptography based mutual authentication scheme for session initiation protocol. *Multimedia Tools and Applications, 66*(2), 165–178.
20. Chen, T., Yeh, H., Liu, P., Hsiang, H., & Shih, W. (2010). A secured authentication protocol for sip using elliptic curves cryptography. *Communications in Computer and Information Science, 119,* 46–55.
21. Lin, C. L., & Hwang, T. (2003). A password authentication scheme with secure password updating. *Computers & Security, 22*(1), 68–72.
22. Yoon, E. J., & Yoo, K. Y. (2009). Cryptanalysis of ds-sip authentication scheme using ecdh. In: *International conference on new trends in information and service science*, pp. 642–647.
23. Xie, Q. (2012). A new authenticated key agreement for session initiation protocol. *International Journal of Communication Systems, 25*(1), 47–54.
24. Farash, M. S., & Attari, M. A. (2013). An enhanced authenticated key agreement for session initiation protocol. *Information Technology And Control, 42*(4), 333–342.
25. Zhang, Z., Qi, Q., Kumar, N., Chilamkurti, N., & Jeong, H. Y. (2015). A secure authentication scheme with anonymity for session initiation protocol using elliptic curve cryptography. *Multimedia Tools and Applications, 74*(10), 3477–3488.
26. Yanrong, L., Li, L., Peng, H., & Yang, Y. (2016). A secure and efficient mutual authentication scheme for session initiation protocol. *Peer-to-Peer Networking and Applications, 9*(2), 1–11.
27. Chaudhry, S. A., Khan, I., Irshad, A., Ashraf, M. U., Khan, M. K., & Ahmad, H. F. (2016). A provably secure anonymous authentication scheme for session initiation protocol. *Security and Communication Networks, 9,* 5016–5027.
28. Xu, D., Zhang, S., Chen, J., & Ma, M. (2017). A provably secure anonymous mutual authentication scheme with key agreement for SIP using ECC. *Peer-to-Peer Networking and Applications, 11,* 837–847.
29. Vanstone, A. (1997). Elliptic curve cryptosystem—The answer to strong, fast public-key cryptography for securing constrained environments. *Information Security Technical Report, 2*(2), 78–87.
30. Lumini, A., & Loris, N. (2007). An improved Bio-hashing for human authentication. *Pattern Recognition, 40*(3), 1057–1065.
31. Jin, A. T. B., Ling, D. N. C., & Goh, A. (2004). Bio-hashing: Two factor authentication featuring fingerprint data and tokenised random number. *Pattern Recognition, 37*(11), 2245–2255.
32. Odelu, V., Das, A. K., & Goswami, A. (2014). A secure effective key management scheme for dynamic access control in a large leaf class hierarchy. *Information Sciences, 269*(4), 270–285.
33. Mansoor, K., Ghani, A., Chaudhry, S. A., Shamshirband, S., & Ghayyur, S. A. K. (2019). Securing IoT based RFID systems: A robust authentication protocol using symmetric cryptography. *Sensors*, *19*(21), 4752. https://doi.org/10.3390/s19214752.
34. Ghani, A., Mansoor, K., Mehmood, S., Chaudhry, S. A., & Rahman, A. U. (2019). M Najmus Saqib, Security and key management in IoT based wireless sensor networks: An authentication protocol using symmetric key. *International Journal of Communication Systems*, *32*(16), e4139. https://doi.org/10.1002/dac.4139.
35. Burrows, M., Abadi, M., & Needham, R. M. (1871). A logic of authentication. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences, 1989*(426), 233–271.

Mahmood Ul Hassan is a Ph.D. scholar at International Islamic University Islamabad Pakistan. His research interests include SIP security, multi-server key agreement and authentication, elliptic curve cryptography, security, MANETs, LTE security, smart grid security, IoT security.

Shehzad Ashraf Chaudhry received the master's and Ph.D. degrees (with Distinction) from International Islamic University Islamabad, Pakistan, in 2009 and 2016, respectively. He is currently working as an Associate Professor with the Department of Computer Engineering, Faculty of Engineering and Architecture, Istanbul Gelisim University, Istanbul, Turkey. He has authored over 75 scientific publications appeared in different international journals and proceedings, including 60 in SCI/E journals. With an H-index of 20 and an I-10 index 37, his work has been cited over 1375 times. He has also supervised over 35 graduate students in their research. His current research interests include lightweight cryptography, elliptic/hyper elliptic curve cryptography, multimedia security, E-payment systems, MANETs, SIP authentication, smart grid security, IP multimedia subsystem, and next generation networks. He occasionally writes on issues of higher education in Pakistan. Dr. Chaudhry was a recipient of the Gold Medal for achieving 4.0/4.0 CGPA in his Masters. Considering his research, Pakistan Council for Science and Technology granted him the Prestigious Research Productivity Award, while affirming him among Top Productive Computer Scientist in Pakistan. He has served as a TPC member of various international conferences and is an Active Reviewer of many ISI indexed journals.

Azeem Irshad received master's degree from Arid Agriculture University, Rawalpindi, Pakistan. Then he completed his PhD from International Islamic University, Islamabad, Pakistan. He has authored more than 60 international journal and conference publications, including 30 SCI-E journal publications. His research work has been cited over 575 times with 12h-index and 13 i-10-index. He received Top Peer-Reviewer Award from Publons in 2018 with 104 verified reviews. He has served as a reviewer for more than 38 reputed journals including IEEE Systems Journal, IEEE Communications Magazine, IEEE Transactions on Industrial Informatics, IEEE Consumer Electronics Magazine, Computer Networks, Information Sciences, CAEE, Cluster Computing, AIHC, Journal of Supercomputing and Wireless Personal Communications, notably. His research interests include strengthening of authenticated key agreements in SIP multimedia, Cloud-IoT, WBAN, TMIS, WSN, Ad hoc Networks, e-health clouds and multi-server architectures.