Cryptanalysis and Improvement of a Multi-Server Authenticated Key Agreement by Chen and Lee’s Scheme

Azeem Irshad, Husnain Naqvi, Shehzad Ashraf Chaudhry
Department of Computer Science & Software Engineering, International Islamic University, Islamabad
e-mail: irshadazeem2@gmail.com, husnain.naqvi@iiu.edu.pk, shahzad@iiu.edu.pk

Muhammad Usman
Department of Computer Science, Faculty of Natural Science, Quaid-I-Azam University, Islamabad, Pakistan
e-mail: musman@qau.edu.pk

Muhammad Shafiq
Department of Information Technology, University of Gujrat, Gujrat, Pakistan, e-mail: shafiq.pu@gmail.com

Omid Mir
The Institute of Networks and Security, Johannes Kepler University Linz, Austria, e-mail: mir@ins.jku.at

Ambrina Kanwal
Department of Computer Science, Bahria University, Islamabad, Pakistan, e-mail: ambrina_ikanwal@yahoo.com

Corresponding author: shahzad@iiu.edu.pk

Multi-server authentication makes convenient to benefit from services of various service providers on the basis of one-time registration through a trusted third party. Since, the users are reluctant to register themselves separately from all servers due to the hassle of remembering many passwords and other cost constraints. The multi-server authentication enables the immediate provision of services by the real-time verification of users on an insecure channel. The literature for multi-server oriented authenticated key agreement could be traced...
back to Li et al. and Lee et al., in 2000. Since then, numerous multi-server authentication techniques have been put forth. Nonetheless, the research academia looks for more secure and efficient authentication protocols. Recently, Chen and Lee’s scheme presented a two-factor multi-server key agreement protocol, which is found to be prone to impersonation, stolen smart card, key-compromise impersonation attack, and trace attacks. Besides, the scheme is also found to have the inefficient password modification procedure. We propose an improved protocol that counters the above limitations in almost an equivalent computation cost. Moreover, our protocol is supplemented with formal security analysis using BAN logic along with performance analysis and evaluation.

**KEYWORDS:** Multi-server authentication, cryptanalysis, biometrics, remote authentication, attack.

### 1. Introduction

In peer-to-peer environment, Multi-Server Authentication (MSA) permits the quick accessibility of numerous online multimedia-based services to users on the basis of a single registration. The architecture of MSA [39, 29] is favorable for both sides, i.e. service providers as well as users. This is because the users need not remember more than one password due to single registration of a trusted third party. Similarly, the MSA architecture eases the service providers of the maintenance of verifier database for each registered user, and the trouble of individualized registrations. The users count on a single registration of a trusted third party to benefit from services of different service providers. The MSA setting would comprise numerous users (Ui), servers (Sj), and a registration centre (RC). The trust flows from RC towards users and servers, as RC registers these entities in the initialization setup on confidential channel. Then onwards, the users could benefit themselves of the services offered by service providers.

Previously, numerous MSA-based schemes gearing towards augmentation in security and efficiency have been presented. Yet, it is believed on account of frequent threats and weaknesses that more resilient MSA protocols need to be demonstrated. Earlier in 2000, Lee and Chang [34] presented a key agreement protocol for MSA framework. The scheme was found vulnerable to masquerading attack and compromised anonymity [51]. Thereafter, Tsaur [47] presented a remote subscriber-based MSA scheme employing RSA crypto-primitives as well as Lagrange interpolating polynomials. The protocol was exposed to password-guessing threat [47]. Then, Li et al. [35] presented a password-based MSA protocol in an artificial neural network system, which requires high training time and a bit higher cost. Thereafter, Lin et al. [39] presented an ElGamal digital signature related MSA protocol. However, the scheme was found too costly for the memory requirements to be applied in smart card based applications. After that, Juang [29] presented a symmetric cryptosystem-based MSA protocol, however having scalability problems due to maintaining users’ verifier-repository at server’s end for all users. Next, Chang and Lee [8] also demonstrated a MSA protocol, which was discovered to be susceptible for privileged-insider and server masquerading threats [38, 18]. Liao and Wang [38], subsequently introduced another dynamic ID-based remote user authenticated key agreement for MSA architecture. Hsiang and Shih [18] discovered the protocol [38] as vulnerable to privileged insider and spoofing threats, and also put forward an improved protocol. Lee et al. [33] remarked that the scheme [18] is unable to accomplish mutual authentication agreement, and onwards demonstrated an enhanced protocol. Nonetheless, Chen and Lee [11] analyzed that the protocol [7] is incapable of providing the security feature of smart card security to comply with two-factor authentication. Besides, [7] also could not resist impersonation attack and bears an inefficient password modification steps due to RC’s involvement. After a deep analysis of Chen and Lee’s scheme [11], we came to know that this protocol sustains stolen smart card attack that leads to the disclosure of session key and password. This protocol is prone to spoofing and trace attack as well. In addition, the scheme bears a defective password-alteration phase. The current study work ascertains few weaknesses in Chen and Lee’s scheme [11] and presents an enhanced protocol ensuring security with efficiency as supported with formal analysis. Moreover, the strength of session key establishment in our protocol is ratified under BAN-logic and random oracle-based formal analysis.
Section 2 relates to a review of Chen and Lee’s protocol. Section 3 studies the cryptanalysis of Chen and Lee’s protocol. Section 4 illustrates our proposed work. Section 5 would demonstrate the security and performance evaluation analysis. The last section wraps up the presented work.

2. Preliminaries

This section briefly illustrates some salient features of hash function, and bio-hashing process.

2.1. Hash Function

A hash function [9, 14, 26], defined as \( h: \{0,1\}^* \rightarrow \{0,1\}^* \) where \( \tau \) denotes a safe length, generating \( \gamma \) string of pre-determined length as output from inputting a random string \( \omega \) of any length, i.e., \( \gamma = h(\omega) \), maintains the following characteristics:

- To define the first characteristic, a one-way hash function serves as a hard problem to alter the string \( \omega \) without updating the hash digest \( h(\omega) \).
- For the second characteristic, it is hard to form a string \( \omega \) generating \( h(\omega) \) as preimage resistance.
- For the third characteristic, it is hard to produce \( \omega \) and \( \omega' \) provided \( \omega \neq \omega' \) where as \( h(\omega) = h(\omega') \) holds as well.

The advantage of attacker may be shown by the following formalization:

\[
Adv^\text{HASH}_A(t_e) = \text{Pro}[ (\omega, \omega') \leftarrow A: \omega \neq \omega' \text{ and } h(\omega) = h(\omega') ]
\] (1)

where \( \text{Pro}[E_o] \) stands for the event \( E_o \)‘s probability of conducting the random experiment, and \( (\omega, \omega') \leftarrow B \) depicts the selected random pair \( (\omega, \omega') \) by the adversary. In this set-up, \( A \) is probabilistic, while the probability related to advantage \( Adv^\text{HASH}_A(t_e) \) can be calculated using the random choices as given by \( A \) in execution time \( t_e \). The hashing function \( h(\cdot) \) is assumed to be resistant to collision, in case \( Adv^\text{HASH}_A(t_e) \leq \epsilon \) holds for adequately small \( \epsilon > 0 \).

2.2. Bio-Hashing

The bio-hashing, being one of the mechanisms to ensure three-factor authentication, complements the two-factor authentication framework with another biometric factor to boost the security. Many studies employing bio-hashing as a three-factor authentication mechanism can be witnessed lately [12, 41]. The sample from biometric scanning tends to behave differently each time it is collected. The bio-hashing function engenders a randomly generated, compressed set of codes by converting the finger-impression codes in so-called biocodes. The hamming distance assists in differentiating the set of various biocodes. Thus, the bio-hashing function is known for countering the de-synchronization problem that might result in capturing the biometric imprints [28].

3. A Review and Cryptanalysis of Chen and Lee’s Scheme

We employed a few notations to make the protocol comprehensible, as given in Table 1.

The Chen and Lee’s protocol includes a few of participating entities, particularly a trusted RC, users \( UI \) and numerous servers \( Sj \). Both users and servers get registered before hand of mutual authentication phase. For this purpose, \( Sj \) shares the secret \( PID \), with on confidential channel. The working of Chen and Lee’s protocol [11] is outlined next.

<table>
<thead>
<tr>
<th>Notations</th>
<th>Narration</th>
</tr>
</thead>
<tbody>
<tr>
<td>RC/UI/Sj</td>
<td>Registration Centre/User/Service provider</td>
</tr>
<tr>
<td>IDi/SID</td>
<td>Users’ identity/ Service provider’s identity</td>
</tr>
<tr>
<td>PWi</td>
<td>User’s password</td>
</tr>
<tr>
<td>PID</td>
<td>One-time pre-shared secret between RC/SJ and UI</td>
</tr>
<tr>
<td>x</td>
<td>RC’s master secret key</td>
</tr>
<tr>
<td>y</td>
<td>RC’s long secret key</td>
</tr>
<tr>
<td>BIOi</td>
<td>Biometric Identity of UI</td>
</tr>
<tr>
<td>H(.)</td>
<td>Bio-hash function</td>
</tr>
<tr>
<td>h(.)</td>
<td>A secure hash digest function</td>
</tr>
<tr>
<td>SC</td>
<td>Smart Card</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3.1. Design of Chen and Lee’s Protocol

The Chen and Lee’s protocol involves two procedures, i.e. 1) registration procedure and 2) login and authentication procedure as exhibited in Figure 1.

3.1.1. User Registration Process

The user completes its registration process by employing the succeeding steps with RC:

1. Initially, Ui selects its identity IDi, password PWi, and engenders a random number ϖ. Thereafter, it calculates \( h(\sigma \oplus PWi) \) and sends \( \{IDi, h(\sigma \oplus PWi)\} \) to registration centre.
2. Next, the registration centre calculates \( Ri=h(h(\sigma \oplus PWi)), Vi=h(Ri||h(x||y)), Pi=V_i \oplus h(\sigma \oplus PWi), Xi=h(IDi||x) \), \( Wi=X_i \oplus h(IDi||h(\sigma \oplus PWi)) \) and \( Fi=h(X_i) \). Subsequently, RC stores these contents in SC \( \{Wi, Pi, h(y), Fi, h()\} \) and submits to Ui.
Next, Ui receives smart card and adds the factor ϖ in it to finalize the registration.

### 3.1.3. Login and Authentication Procedure

We illustrate login and authentication procedure in this sub-section.

1. In login process, Ui is approved from the smart card to initiate the authentication process. For the said objective, Ui inserts its identity and password, i.e. \((ID_i\) and \(PW_i\)). This is followed by the smart card computation of \(X_i = Wi \oplus h(ID_i || h(\sigma \oplus PW_i))\), \(F_i^* = h(X_i)\) parameters. Then, the user compares the equation, i.e. \(F^* = F_i\). If it holds valid, then the user engenders a random number \(\beta_i\), and calculates \(V_i = Pi \oplus h(\sigma \oplus PW_i), Ri = h(\sigma \oplus PW_i)\), \(Bij = Ri \oplus h(h(y) || \beta_i || SID), TIDI = h(\sigma \oplus PW_i) \oplus h(X_i || Vi || \beta_i), Dij = Xi \oplus h(V_i || \beta_i || SID)\) and \(Ci = h(Ei || Vi || \beta_i)\). Thereafter, Ui submits the parameters \(m_i = \{TIDI, Bij, Dij, Ci, \beta_i\}\) to service provider.

2. The service provider receives \(m_i = \{TIDI, Bij, Dij, Ci, \beta_i\}\) and calculates \(Ri = Bij \oplus h(h(y) || \beta_i || SID), V_i = h(Ri || h(\sigma || y)), X_i = Dij \oplus h(V_i || \beta_i || SID), h(\sigma \oplus PW_i) = TIDI \oplus h(X_i || Vi || \beta_i)\) and \(Pi = Vi \oplus h(\sigma \oplus PW_i)\). Thereafter, it compares the equation, i.e. \(h(Pi || Vi || \beta_i) = Ci\). If valid, it creates a random number \(\beta_i\), calculates \(Qij = h(Pi || Vi || \beta_i || SID)\) and sends \(m_j = \{Qij, \beta_j\}\) towards user for the purpose of verification.

3. Then, Ui calculates \(h(Pi || \beta_i || Vi || SID)\) and also verifies the equation, i.e. \(h(Pi || \beta_i || Vi || SID) = Qij\). If this equality is valid, Ui calculates \(Qij' = h(Pi || \beta_i || Vi || SID)\) and submits the message \(m_j = \{Qij\}\) to service provider for further confirmation with \(\beta_j\) challenge.

4. The service provider receives \(m_j\) and computes \(h(Pi || \beta_j || Vi || SID)\). Next, the service provider verifies the equation, i.e. \(h(Pi || \beta_j || Vi || SID) = Qij'\). If this equality holds true, the service provider further computes the agreed session key with \(Ui\) as \(SK = h(Pi || \beta_i || \beta_j || Vi || SID)\).

### 3.2. Pitfalls in Chen and Lee’s Protocol

The Chen and Lee’s protocol has been found prone to stolen smart card threat, user masquerading threat, trace threat, key compromise impersonation threat. At the same time, the protocol bears an expensive password alteration process. Before recounting those pitfalls, we describe an attack model.

#### 3.2.1. Attack Model

In this study, an adversary \(A\) is assumed to be proficient \([4, 40, 42]\) in the following skills:

1. \(A\) could apply reverse engineering procedures on the stolen smart card contents and attempt to guess low-entropy secrets, like password or identity.

2. \(A\) could intercept the messages on insecure channel and manipulate by modifying or replaying the same contents.

3. \(A\) could be any legal user (insider) behaving maliciously.

#### 3.2.2. Cryptanalysis and Drawbacks

The details of reported attacks and other limitations in Chen and Lee’s scheme are given below:

**a. Stolen smart card threat**

An adversary \(A\) may trigger this attack if he/she could access the smart card contents which are not properly encrypted before storage. In our protocol, the smart card contains the parameters \(\{Wi, Pi, F_i, h(\sigma), h(y)\}\), while the messages that may be intercepted on public channel are \(m_i = \{TIDI, Bij, Dij, Ci, \beta_i\}\) and \(m_j = \{Qij\}\). Given that \(\beta_i\) and \(SID\) could be accessed on public channel, and \(h(y)\) could be approached from stolen card, the attacker may produce \(h(h(y) || \beta_i || SID)\) and approach \(R_i^*\) factor by calculating \(R_i^* = Bij \oplus h(h(y) || \beta_i || SID)\). Subsequently, owing to the accessibility of ‘\(\sigma\)’ random integer in smart card, \(A\) may initiate an offline-dictionary attack to guess the original password of \(Ui\). Now, the adversary attempts all dictionary strings as password \((PW_i)\) in computing \(R_i^* = h(\sigma \oplus PW_i)\) and verifying the equation, i.e. \(R_i^* = R_i\). Whenever the adversary finds the matching equation, the former will come to know the right password.

After having the password \(PW_i\) guessed, the adversary may compute \(Vi = h(\sigma \oplus PW_i) \oplus Pi\). Onwards, \(A\) may comfortably establish the legal session key on executing the hash function as \(h(Pi || \beta_i || Vi || SID)\). In this manner, the attacker may easily guess the identical (valid) session key \(SK\) after stealing the smart card, as constructed by the legitimate members. Hence the protocol is prone to stolen smart card threat.

**b. User impersonation threat**

The Chen and Lee’s protocol is prone to user spoofing attack, provided the smart card contents are available to adversary. By employing these contents, \(A\) may compute the legitimate password \(PW_i\) after following the steps as mentioned in sub-section 3.2.2(a).
The attacker may compute $V_i = Pi \oplus h(s) \oplus PW_i$ and $R_i = h(h(s) \oplus PW_i)$). Subsequently, A makes a guess of the ID$i$ string after checking the entire set of potential strings ID$i^*$ by verifying the equations, i.e., $X_i^* = Wi \oplus h(ID_i^* || h(s) \oplus PW_i)$ and $F_i = h(x_i^*)$, frequently. On positive matching, the legal identity (ID$i$) and $X_i^*$ are exposed. Thereafter, A engenders a random number $\beta_i$ and calculates $B_i = R_i \oplus h(h(y) \parallel \beta_i \parallel SID), TID_i = h(s) \oplus PW_i) \oplus (X_i \parallel Vi \parallel \beta_i), D_{ij} = X_i \oplus h(V_i \parallel \beta_i) \parallel SID)$ and $C_i = h(P_i \parallel Vi \parallel \beta_i)$. Ultimately, A designs authentication request as $m_z = \{TID_i, B_i, D_{ij}, C_i, \beta_i\}$ effectively.

c Trace attack

In this threat, the attacker could discover the session participants on the basis of distinguishing and comparing known parameters between different sessions of the same participants. In scheme [11], a privileged insider with known $h(y)$, could eavesdrop the request $m_z = \{TID_i, B_i, D_{ij}, C_i, \beta_i\}$ acting maliciously and try to trace the similarity between different sessions through $R_i$ after calculating $R_i = B_i \oplus h(h(y) \parallel \beta_i \parallel SID).$ The $R_i$ parameter is not changed among various sessions constructed between user and server, in case the user does not update his/her PW$i$ or $s$ parameters in SC. Thus, the scheme [11] is vulnerable to trace threat.

d Key-compromise impersonation threat (KCI)

In KCI attack, the attacker may use the guessed or stolen factor of some user to spoof as a service provider. The Chen and Lee’s protocol is vulnerable to KCI threat, if the data contents in user’s smart card are accessed by attacker. After accessing the password of $U_i$, as shown in Section 3.2.2(a), the attacker could impersonate as a service provider by designing a message $m_z = \{Q_i, \beta_i\}$ after generating a random integer $\beta_i$, and constructing $Q_i^* = h(P_i \parallel \beta_i \parallel V_i \parallel SID)$. This is because $P_i$ and $V_i$ factors could be produced by maneuvering the smart card factors as depicted in Section 3.2.2(b). The message $m_z$ is submitted to user, and will be duly verified by user, though fake. In this manner, a successful spoofing attack could be launched towards user in scheme [11].

e No session key security

In case the parameters $P_i$ and $V_i$ are recovered by the attacker out of data stolen from smart card, the adversary might calculate past session keys through eavesdropping $\beta_i, \beta_i$ and computing session key, i.e. $SK = h(P_i \parallel \beta_i \parallel \beta_i \parallel V_i \parallel SID)$.

f No direct password alteration

The Chen and Lee’s scheme claims that user does not consult RC to modify the password; nonetheless, the scheme [11] does not provide any procedure for altering PW$i$ without the involvement of RC. The alteration of the password needs updating $P_i = V_i \oplus h(s) \oplus PW_i)$, whenever the password PW$i$ is modified. Here, $R_i$ is employed for computing $V_i$, as $V_i = h(R_i \parallel h(x_i || y))$. Moreover, $R_i$ is a parameter that cannot be constructed without password, i.e. $R_i = h(h(s) \oplus PW_i)$). Consequently, $U_i$ will need to consult $S_j$ every instant to update $R_i$ since it does not possess the parameter $h(x_i || y)$. The above proof negates the claim of Chen and Lee’s scheme for the capability of password alteration without engaging RC.

4. The Proposed Model

The multi-server architecture involves three participating entities namely; registration centre (RC), user ($U_i$), and server or service provider ($S_j$). RC enables to register users and provide services from servers onwards. In initialization, the RC chooses its master secret key as $x$ and another long term secret integer as $y$. These two parameters are utilized in registering all users. RC calculates $h(x || y)$ and $h(y)$ and then shares these parameters with all legal service providers $S_j$, using a confidential channel. The contributed model also includes three sub-procedures, i.e., user registration procedure, login and authentication procedure, and password alteration method as illustrated below:

4.1. User Registration Procedure

To become a legal subscriber of the network, the user performs the succeeding steps with RC:

1. The user chooses the parameters as identity ID$i$, biometric $BIO_i$, password PW$i$, and creates a random integer $r$. Then, user further calculates $Y = H(ID_i || BIO_i), TPW = h(s) \oplus H(BIO_i || PW_i)$, and sends $\{ID_i, Y, TPW\}$ to RC to complete registration.

2. Next RC calculates $R_i = h(y || x), V_i = h(R_i || h(x || y)), E_i = V_i \oplus h(Y || TPW), Q_i = R_i \oplus h(x || y) \oplus h(y) \oplus TPW$, and $F_i = h(h(ID_i || TPW))$. Subsequently, it creates random integer $t$, and calculates $PID_i = h(ID_i || x) \parallel h(t) \parallel h(x || y)$ and $D_i = h(ID_i || x) \oplus h(ID_i || TPW)$. Later, the registration centre stores $(PID_i, D_i, E_i, F_i, Q_i, h(x), h(y))$ in smart card and sends towards $U_i$.

3. $U_i$ adds the parameter $s$ in smart card finally.
Figure 2
The proposed protocol (Registration and mutual authentication)

REGISTRATION STEPS:
1. Chooses $ID_i$, $PW_i$, $\sigma$
   
   Imprints $BIO_i$, Computes $Y = H(ID_i \| BIO_i)$, $TPW = h(\sigma \oplus H(BIO_i \| PW_i))$
   
   $\{ID_i, Y, TPW\}$

2. $R_i = h(Y \| x)$
   $V_i = h(R_i \| h(x \| y))$
   $E_i = V_i \oplus h(Y \| TPW)$
   $Q_i = R_i \oplus h(x \| y) \oplus h(y) \oplus TPW$
   $F_i = h(h(ID_i \| TPW))$

3. $U_i$ picks the smart card and inserts $\sigma$ as well in SC. $SC\{PID_i, Di, E_i, Fi, Qi, h(y), h()\}$

  生成随机数 $t$
   $PID_r = (h(ID_i \| x) \| h(t) \| h(y))$
   $Di = h(ID_i \| x)$
   $Di = h(ID_i \| TPW)$

MUTUAL AUTHENTICATION STEPS:
1. $U_i$ inputs $ID_i$, $PW_i$, and imprints $BIO_i$ in SC. Then calculates $TPW = h(\sigma \oplus H(BIO_i \| PW_i))$

2. $h(ID_i \| x)$
   $h(ID_i \| x) \| h(x \| y)$

3. $h(ID_i \| x)$
   $h(ID_i \| x)$

4. Verifies $SK_i = h(ID_i \| x) \| h(y) \| V_i \| S ID)$
   $h(SK_i) = h(ID_i \| x) \| h(y) \| V_i \| S ID)$
4.2. Login and Authentication Procedure

The mutual authentication phase between Ui and Sj is given below:

1. In this procedure, the user seeks authenticated access of services from server on account of RC. To serve the purpose, Ui inputs the identity IDi and password PWi into smart card while inputs biometric BIo/i into the scanner. Then smart card computes $TPW = h(σi ∘ H(BIo/i || PWi))$ and checks whether $Fi^* = h(h(IDi || TPW))$. If true, then Ui creates a random integer $βi$, and calculates $Y = h(IDi || BIo/i), h(IDi || x) = Di ∘ h(IDi || TPW), Bij = Qi ∘ h(SID || y) || βi$ $⊙ TPW ∘ h(y) ⊕ h(PID, || h(IDi || x)), Vi = Ei ∘ h(Y || TPW)$, and $TIDi = h(PIDi, || Vi || βi)$. Next, Ui sends the message $m1 = \{PIDi, TIDi, Bij, βi\}$ to Sj for verification.

2. $RC$, after receiving $m1 = \{PIDi, TIDi, Bij, βi\}$ computes $h(IDi || x) || h(t) = PIDi ∘ h(x || y), Ri = Bi ∘ h(SID || h(y)) || βi$ $⊙ TPW, h(IDi || x) ∘ h(x || y) || h(IDi || x)$ and $Vi = h( Ri || h(x || y))$. Next, $RC$ further computes $h(PIDi, || Vi || βi)$ and compares $TIDi = h(PIDi, || Vi || βi).$ If the equation is verified, $RC$ will create a random integer $t$ and calculate $PID ' = (h(IDi || x) || h(t)) ∘ h(x || y), Ti = PIDi ∘ h(PIDi, || h(IDi || x) || Vi)$. $Qi = h(h(IDi || x) || Ti || βi || βi || Vi || SID)$ after generating a random number $βi$. Ultimately, $RC$ sends the message $m2 = \{Qi, Ti, βi\}$ to Ui.

3. After receiving $m2$, $Ui$ computes $h(h(IDi || x) || βi || Vi || SID))$ and compares the equation $h(h(IDi || x) || Ti || βi || βi || Vi || SID)) = Qi$. If the equation is verified, $Ui$ further calculates $SKij = h(IDi || x) || βi || βi || Vi || SIDi, Qi ' = h(SKij || h(IDi || x) || βi || Vi || SID)$ and forwards $m3 = \{Qi '\}$. To server so that it may verify the $βi$-based challenge. At the same time, $Ui$ calculates $PIDi = Ti ∘ h(PIDi, || h(IDi || x) || Vi)$ and replaces the parameter PID with PIDi in SC.

4. The server (Sj), after receiving $m3$, calculates $SKij = h(IDi || x) || βi || βi || Vi || SID).$ Afterwards, it verifies the equation, i.e. $h(SKij || h(IDi || x) || βi || Vi || SID) = Qi'$. On successful verification, it constructs session key with user as $h(h(IDi || x) || βi || Vi || SID)$, finally. The details of the contributed protocol can be witnessed from Figure 2.

4.3. Password Alteration Mechanism

Ui may alter his/her password through initiating the password alteration steps, into another password (PWnew) without getting any assistance out of registration centre. These steps are illustrated below.

1. Initially, the user shall insert its SC into the reader for inputting identity (IDi*), password (PW* i) and imprinting its biometric factor BIo/* into the scanner device. Thereafter, the SC calculates $TPW = h(σi ∘ H(BIo/i || PWi))$ and checks $Fi^* = h(h(IDi || TPW)).$ If this equality holds, then user proceeds to the next step.

2. Afterwards, the smart card calculates $TPW = h(σi ∘ H(BIo/i || PWi))$ and computes $Vi = Ei ∘ h(H(IDi || BIo/i || TPW), Qi = Qi ∘ TPW, h(IDi || x) = Di ∘ h(IDi || TPW)$.

3. Subsequently, the user shall insert a new password (PWnew). The smart card then calculates $TPW = h(σi ∘ H(BIo/i || PWnew))$. $Ei^new = Vi ∘ h(H(IDi || BIoi) || TPW), Qi^new = Qi^new ∘ TPW, Di^new = h(IDi || x) ∘ h(IDi || TPW)$ and $TPW^new = h(H(IDi || TPW))$.

4. Next, the values Di, Ei, Fi, and Qi are replaced by $Di^new, Ei^new, Fi^new,$ and $Qi^new$ in the smart card.

5. Security Analysis

A comprehensive discussion on the security analysis of proposed model is provided in the following sub-sections.

5.1. Replay Attacks

In replay attacks, the intercepted messages are replayed without undergoing modifications to betray any legal member [1, 3, 17, 19, 44].

An attacker A, having access to factors {PIDi, ZIDi, Bi, βi, Qi, Ti, βi, Qi'} could try for replaying these contents in order to forge the legitimate participant. Nonetheless, the use of temporary novel parameters, such as $βi$ and $βi$, by the legitimate members, for every session, discourage the attacker for initiating an attack. In case the adversary attempts to replay $m3 = \{PIDi, ZIDi, Bi, βi\}$ towards server, the server could confirm the legitimacy of user in $m3$ in response to the $βi$-based challenge. Simultaneously, the user authenticates Sj in $m3$ to respond the $m3$-based βi challenge. Thus, the above discussion indicates towards a defense capability of the proposed model against replay attack.
5.2. Modification Attacks

The attacker could alter the intercepted parameters to resubmit to the intended party, in case a protocol is designed with least focus on resisting the modification attack.

If any adversary tries to alter the public parameters \( \{ \text{PID}, \ ZIDi, \ Bi, \ f, \ Qi, \ Ti, \ b, \ Qij \} \), \( A \) will be unable to reconstruct these contents \( \{ ZIDi, \ Bi, \ Qij, \ Qij' \} \) by creating fresh session variables, since the construction of these messages requires the knowledge of \( Vi \) and \( h(IDi \| x) \) which are only known to the legitimate participants. Hence, the legitimate participant will be able to detect any malicious participant. Therefore, the contributed scheme may easily thwart this threat.

5.3. Password or Secret Guessing Threat

The guessing attack is possible if the adversary tries to guess the password of user or some S\( j \)'s long term secret on account of intercepted parameters. In the proposed scheme, \( A \) can approach the factors \( \{ \text{PID}, \ ZIDi, \ Bi, \ f, \ Qi, \ Ti, \ b, \ Qij \} \) on little inspection of any public channel. Nevertheless, an attacker could not derive the password, since it is not utilized as a factor for the calculation of any parameter; hence it minimizes the chances of guessing the corresponding parameters.

5.4. Stolen-Verifier Threats

The adversary could embezzle with valuable data which are stored at server's end; and the database of U\( i \)'s secrets like passwords or other parameters, could be utilized to impersonate as the legitimate users. The proposed scheme provides mutual authentication without depending on database maintenance on S\( j \) or RC's end. This suggests that the stolen verifier attack is defeated in our scheme.

5.5. Offline-Dictionary Attack Based on Stolen Smart Card Contents

In this attack, the attacker attempts various combinations of dictionary contents after having the stolen smart card information [23, 46].

Using SC, an attacker might try to manipulate with its available contents of smart card i.e., \( \{ \text{PID}, \ Di, \ Ei, \ Fi, \ Li, \ Qi, \ h() \} \). For guessing the password from \( Ei, \ Fi \) and \( Qi \) parameters, \( A \) needs to know \( IDi, \ r \) and \( BIOi \) to guess \( PWi \) from \( TPW \), where \( TPW = h(r \oplus H(BIOi || PWi)) \). Consequently, the offline-dictionary attack using SC cannot be launched in polynomial time.

5.6. Session Key Security

This security feature warrants the knowledge of session key only to the known legitimate parties, such as user and service provider.

In the contribututed protocol, the established session key is designed as \( SK = h(h(IDi \| x)||f_i||f_j||Vi||SID) \). For constructing a valid session key, an attacker shall require \( h(IDi \| x) \) and \( Vi \) contents. If the user's identity is stolen or guessed by adversary, the latter may not be able to assemble \( h(IDi \| x) \) as the adversary does not possess the parameter \( x \). This stops the adversary from establishing a valid \( SK \), contrary to scheme [11]. Furthermore, \( A \) shall require \( Vi \) for creating a valid \( SK \); nonetheless, an attacker is not able to recover \( Vi \) as \( A \) does not possess \( TPW, \ BIOi \), and \( Ei \) factors. Therefore, there is much less of a chance for attacker to initiate this attack.

5.7. Known-Key Security

This feature ensures the confidentiality of private keys even if session key for a specific session gets revealed [2, 30].

Given that the agreed session key \( SK = h(h(IDi \| x)||f_i||f_j||Vi||SID) \) does not include \( U_i \)'s password \( PWi \) as a factor, though it bears \( h(IDi \| x) \), again the attacker may not derive \( x \) which is the high entropy master secret key of RC. Owing to this, the attacker might not recover the factors or parameters from an exposed session key. Thus, the contributed protocol is immune and fully complies with known-key security.

5.8. Mutual Authentication

The compliance to this feature lets the involved participants verify one another in the proposed scheme. In this scheme, both of the participants verify one another on account of \( h(IDi \| x) \) and \( Vi \). These both parameters are only accessible to adversary if the secrets of both RC and S\( j \) are exposed, and not otherwise. The attacker cannot recover \( h(IDi \| x) \) from \( PID_i \), by computing \( h(IDi \| x)||h(t)) = PID_i \oplus h(x||y) \), since it does not possess \( h(x||y) \). At the same time, the accessibility to \( Vi \) requires the information of either \( PWi \) and \( BIOi \), or \( h(x || y) \).

5.9. Anonymous Authentication

This feature lets the user communicate without exposing his/her identity [10, 32, 45]. The user submits the messages for authentication and gets verified
without declaring its true identity.

In the proposed protocol, the user sends his/her identity in the form of $PID_r = (h(ID_i || x) || h(t)) \oplus h(x || y)$, that is masked by using $t$, as assumed by server. The server recovers $h(ID_i || x)$ by taking XOR of $h(x || y)$ with $PID_r$ and then computing $h(ID_i || x)$ as a dynamic identity for additional calculation. This manner, our scheme fosters the element of anonymity to a particular user.

5.10. Immune from Key-Compromise Impersonation Threat

In such attack, an adversary could impersonate one participant of a particular session if it steals some key of another participant of the same session. The contributed protocol is immune of KCI threat in contrary to scheme [11], as the contents of stolen card will not help the attacker to get other constructive parameters, such as, $Vi$ and $(ID_i || x)$. Hence, the adversary cannot construct up-to-date $Qij$ parameter, and ultimately no KCI attack may be initiated.

5.11. Alteration of Password Without RC Involvement

The password could be comfortably updated without engaging RC, as contrary to Lee et al. and Chen and Lee, by adopting the procedure described in Section 3.4. Both of the schemes [11, 33] do not modify the password without RC engagement. As in scheme [11], the design of $Ri$ involves the password as a component, which is reused in the design of $Vi$, while $Vi$ is again used in the construction of $Ei$ for storing in SC [13, 49, 52]. The proposed protocol employs $BIOi$ for the construction of $Ri$ parameter, rather than $PW_i$, which enables the proposed scheme to update the password without RC involvement.

6. Formal Security Analysis

We demonstrate the robustness of key agreement, session key’s confidentiality and mutual authentication related features by using formal security analysis through Burrows-Abadi-Needham (BAN) logic [6] and random-oracle model (ROM). In this logic, we utilize few terms quite frequently, known as principals, keys and nonces which are described below.

The **principals** are the participating agents in an authentication protocol.

The **Keys** (symmetric) are utilized for encrypting the messages.

**Nonces** are the type of random secrets that are used only once.

Some notations related to BAN logic are defined as follows:

- $\psi \models \xi$: $\psi$ believes the statement $\xi$.
- $\psi \models \xi$: $\psi$ sees $\xi$. $\psi$ receives a message $\xi$ and could either read or replay it.
- $\psi \models \neg \xi$: $\psi$ once said $\xi$. Earlier the agent $\psi$ had sent a message $\xi$ and $\psi$ also believed $\xi$ when sent.
- $\psi \models \xi$: $\psi$ has jurisdiction over $\xi$; or $\psi$ enjoys authority over $\xi$ or it could be trusted.

### (\xi)_h: The formula $\xi$ is freshly created.

### (\xi)_r: The formula $\xi$ is used in combination with formulae $\Theta$.

### Rule 5: $\xi, \Theta$: $\xi$ or $\Theta$ being the part of message $\xi, \Theta$.

### Rule 6: $\xi, \Theta K$: $\xi$ or $\Theta$ is encrypted with key $K$.

- $\psi \leftrightarrow \psi'$: $\psi$ and $\psi'$ can securely contact using he shared key $K$.

Some rules particularly (Message meaning rule as Rule 1, nonce verification rule as Rule 2, jurisdiction rule as Rule 3, freshness conjunctenation rule as Rule 4, belief rule as Rule 5, and session keys rule as Rule 6) employed in BAN logic are stated below:

**Rule 1:**

$$\psi \models \xi \quad \psi \models K \psi' \quad \psi \models (\xi_\Theta) \\
\psi \models (\xi_\Theta)$$

**Rule 2:**

$$\psi \models (\xi) \quad \psi \models \psi' \models \xi \\
\psi \models \psi' \models \xi$$

**Rule 3:**

$$\psi \models \psi' \models \xi, \psi' \models \psi' \models \xi \\
\psi \models \xi$$

**Rule 4:**

$$\psi \models (\xi, \Theta) \\
\psi \models (\xi, \Theta)$$

**Rule 5:**

$$\psi \models (\xi) \quad \psi \models (\Theta) \\
\psi \models (\xi, \Theta)$$

**Rule 6:**

$$\psi \models (\xi) \quad \psi \models (\psi') \models \xi \\
\psi \models \psi' \leftrightarrow \psi'$$

The contributed work should meet the following targets for ensuring the security using BAN logic, under the indicated postulates:
To proceed, we first transform the communication messages into idealized form as given below:

\[ \text{IM}_i : \text{UI} \rightarrow \text{SID} : \langle \text{PID}_e, \text{ZIDI}, \text{Bij}, \beta_i : \{ (h(ID_i || x) || h(t_i) \}_{h(x || t_i)}, \langle \text{PID}_e, \beta_i \rangle_{VR}, \langle \text{RI} \rangle_{h(x || y), h(y), h(ID_i), \beta_i} \rangle \]

\[ \text{IM}_i : \text{S} \rightarrow \text{UI} : \langle \text{Qij}, \beta_i, \beta_j : \{ (\text{SID})_{h(ID_i), \beta_i} \} \rangle \]

Further, the following premises could be drafted to verify the strength of the proposed scheme:

\[ \text{Rule 1} \]
\[ \text{Rule 2} \]
\[ \text{Rule 3} \]
\[ \text{Rule 4} \]
\[ \text{Rule 5} \]

Next, the established idealized forms of the contributory protocol could be evaluated and tested in view of postulates as listed above.

Employing these notations, idealsizations and rules, we derive the following results:

Using the idealized forms, \( \text{IM}_i \) and \( \text{IM}_j \), we get:

\[ \text{IM}_i : \text{UI} \rightarrow \text{SID} : \langle (h(ID_i || x) || h(t_i) \rangle_{h(x || t_i)}, \langle \text{PID}_e, \beta_i \rangle_{VR}, \langle \text{RI} \rangle_{h(x || y), h(y), h(ID_i), \beta_i} \rangle \]

\[ \text{IM}_j : \text{UI} \rightarrow \text{Qij} : \{ (\text{SID})_{h(ID_i), \beta_i} \} \]

Further, the following premises could be drafted to verify the strength of the proposed scheme:

\[ \text{Rule 1} \]
\[ \text{Rule 2} \]
\[ \text{Rule 3} \]
\[ \text{Rule 4} \]
\[ \text{Rule 5} \]

Next, the established idealized forms of the contributory protocol could be evaluated and tested in view of postulates as listed above.

Employing these notations, idealsizations and rules, we derive the following results:

Using the idealized forms, \( \text{IM}_i \) and \( \text{IM}_j \), we get:

\[ \text{IM}_i : \text{UI} \rightarrow \text{SID} : \langle (h(ID_i || x) || h(t_i) \rangle_{h(x || t_i)}, \langle \text{PID}_e, \beta_i \rangle_{VR}, \langle \text{RI} \rangle_{h(x || y), h(y), h(ID_i), \beta_i} \rangle \]

\[ \text{IM}_j : \text{UI} \rightarrow \text{Qij} : \{ (\text{SID})_{h(ID_i), \beta_i} \} \]

Next, the established idealized forms of the contributory protocol could be evaluated and tested in view of postulates as listed above.

Employing these notations, idealsizations and rules, we derive the following results:

Using the idealized forms, \( \text{IM}_i \) and \( \text{IM}_j \), we get:

\[ \text{IM}_i : \text{UI} \rightarrow \text{SID} : \langle (h(ID_i || x) || h(t_i) \rangle_{h(x || t_i)}, \langle \text{PID}_e, \beta_i \rangle_{VR}, \langle \text{RI} \rangle_{h(x || y), h(y), h(ID_i), \beta_i} \rangle \]

\[ \text{IM}_j : \text{UI} \rightarrow \text{Qij} : \{ (\text{SID})_{h(ID_i), \beta_i} \} \]

Next, the established idealized forms of the contributory protocol could be evaluated and tested in view of postulates as listed above.

Employing these notations, idealsizations and rules, we derive the following results:

Using the idealized forms, \( \text{IM}_i \) and \( \text{IM}_j \), we get:

\[ \text{IM}_i : \text{UI} \rightarrow \text{SID} : \langle (h(ID_i || x) || h(t_i) \rangle_{h(x || t_i)}, \langle \text{PID}_e, \beta_i \rangle_{VR}, \langle \text{RI} \rangle_{h(x || y), h(y), h(ID_i), \beta_i} \rangle \]

\[ \text{IM}_j : \text{UI} \rightarrow \text{Qij} : \{ (\text{SID})_{h(ID_i), \beta_i} \} \]

Next, the established idealized forms of the contributory protocol could be evaluated and tested in view of postulates as listed above.

Employing these notations, idealsizations and rules, we derive the following results:

Using the idealized forms, \( \text{IM}_i \) and \( \text{IM}_j \), we get:

\[ \text{IM}_i : \text{UI} \rightarrow \text{SID} : \langle (h(ID_i || x) || h(t_i) \rangle_{h(x || t_i)}, \langle \text{PID}_e, \beta_i \rangle_{VR}, \langle \text{RI} \rangle_{h(x || y), h(y), h(ID_i), \beta_i} \rangle \]

\[ \text{IM}_j : \text{UI} \rightarrow \text{Qij} : \{ (\text{SID})_{h(ID_i), \beta_i} \} \]
3 Calls Reveal oracle on input $Q\delta_i$ to produce $(h(ID_i || x), \delta_i, \beta_i, \gamma_i, \delta_i, Vi, SID)$ as $(h(ID_i || x) || \delta_i || \beta_i || \gamma_i || Vi || SID) \leftarrow$ reveal1 $(Q\delta_i)$
4 Calls Reveal oracle on input $h(ID_i || x)$ to produce $(ID_i', x')$ as $(ID_i || x') \leftarrow$ reveal1 $(h(ID_i || x))$
5 Calls Reveal oracle on input $F_i$ to produce $(ID_i, TPW')$ as $(ID_i || TPW') \leftarrow$ reveal1 $(F_i)$
6 Computes $K' = R_i \oplus h(x || y) = Q_i \oplus h(y) \oplus TPW'$
7 Eavesdrops the login request message $m_1 = \{ PID_i, ZID_i, B_i, \delta_i \}$, where $PID_i = (h(ID_i || x), t) \oplus h(x || y), ZID_i = h(PID_i, \delta_i, BI, Ti = PID_i', h(SID || h(y)) || \delta_i) \oplus h(PID_i, \delta_i, BI, Ti = PID_i', h(SID || h(y)) || \delta_i)$.
8 Computes $ZID_i' = K' \oplus h(h(SID || h(y)) || \delta_i) \oplus h(PID_i, \delta_i, BI, Ti = PID_i', h(SID || h(y)) || \delta_i)$
9 If $(ZID_i' \neq ZID_i)$ Then
10 Accept $ID_i$ as the true identity of the user $U_i$.
11 Return 1 (True)
12 Else
13 Return 0 (False)
14 End if

**Theorem 1.** If a one-sided hash function acts closely to some randomly behaving oracle, the contributed protocol shall remain protected of malicious adversary in case the latter attempts to capture the user’s identity.

**Proof.** In this proof, any shrewd attacker $A$, who approaches the publicly available message parameters as $(PID_i, ZID_i, BI, \delta_i, \beta_i, TI, Bi, \delta_i')$, might use the oracle Reveal1 to implement algorithm $ALG \text{hash}_{SPMSAC}$. The probability for the success of $ALG \text{hash}_{SPMSAC}$ amounts to Sussp1=Pr.2 $[ALG \text{hash}_{SPMSAC} = 1]-1$, where $Pr[E_{\text{str}}]$ represents the probability of an event $E_{\text{str}}$. The advantage function for algorithm (experiment) $ALG \text{hash}_{SPMSAC}$ is referred to as $Adv_{\text{fun}, 1} \text{hash}_{SPMSAC}(t_{str}, q_{\text{str}}) = \max_{\delta} [Sussp1 \text{hash}_{SPMSAC}]$, with execution time $t_{str}$ and the corresponding random query $q_{\text{str}}$ as maximized on adversary $A) [15-16, 17]$. We could safely refer to the contributed protocol as secure against the attacker $A$ so it may not recover the true identity $ID_i'$, provided $Adv_{\text{fun}, 1} \text{hash}_{SPMSAC}(t_{str}, q_{\text{str}}) \leq \rho$ for any adequately small $\rho > 0$. According to the above testing algorithm, if the attacker $A$ is capable enough to invent a one-sided hash function $h(.)$, and deciphering the oracle, it might recover the valid legal $ID_i'$ and eventually wins the game. Nonetheless, keeping in view the above definition, this would be computationally impractical to reverse the hash function, as $Adv_{\text{fun}, 1} \text{hash}_{SPMSAC}(t_{str}) \leq \rho$ for any adequately small $\rho > 0$.

**Algorithm 2.** $ALG_{2 \text{hash}_{SPMSAC}}$

1 Attacker recovers information from smart card using power analysis as $SC \{ PID_i, Di, E_i, Fi, Qi, hi) \}$. 
2 Attacker intercepts message $m_2 = \{ Q_i, Ti, \beta_i \}$ in the authentication phase, where $Q_i = h(h(ID_i || x) || \beta_i || \beta_i \ || Vi || SID), Ti = PID_i', h(PID_i, \ || II || Vi)$. 
3 Calls Reveal oracle on input $Q_i \rightarrow$ produce $(h(ID_i || x), \beta_i, \beta_i, Vi, SID)$ as $(h(ID_i || x) || \beta_i || \beta_i || Vi || SID) \leftarrow$ reveal1 $(Q_i)$
4 Calls Reveal oracle on input $h(ID_i || x)$ to produce $(ID_i || x')$ as $(ID_i || x') \leftarrow$ reveal1 $(F_i)$
5 Calls Reveal oracle on input $F_i$ to produce $(ID_i, TPW')$ as $(ID_i || TPW') \leftarrow$ reveal1 $(F_i)$
6 Computes $K' = R_i \oplus h(x || y) = Q_i \oplus h(y) \oplus TPW'$
7 Eavesdrops the message $m_2 = \{ PID_i, ZID_i, B_i, \delta_i \}$ in the authentication phase, where $PID_i = (h(ID_i || x), t) \oplus h(x || y), ZID_i = h(PID_i, \delta_i, BI, Ti = PID_i', h(SID || h(y)) || \delta_i) \oplus h(PID_i, \delta_i, BI, Ti = PID_i', h(SID || h(y)) || \delta_i)$
8 Computes $Y = h(ID_i \ || BI)\delta_i$
9 Calls Reveal oracle on inputting $H(ID_i \ || BI)\delta_i$ to produce $(ID_i, BI)\delta_i$ as $(ID_i || BI)\delta_i \leftarrow$ reveal1 $(H(ID_i \ || BI)\delta_i)$
10 Computes $R_i' = h(H(ID_i \ || BI)\delta_i || x')', h(x || y) = R_i' \ominus K_i', Vi = h(R_i' || h(x || y))$
11 Calculates session key as $SK_i' = h(h(ID_i || x') || \beta_i || \beta_i \ || VI || SID)$
12 Compute $ZID_i' = h(PID_i || h(\delta_i, BI, Ti = PID_i', h(SID || h(y)) || \delta_i \ || PID_i, \delta_i, BI, Ti = PID_i', h(SID || h(y)) || \delta_i))$
13 If $(ZID_i' \neq ZID_i)$ Then
14 Accept $SK_i'$ as the validated session key $SK_i'$ among participants $U_i$ and $S_i$
15 Return 1 (True)
16 Else
17 Return 0 (False)
18 End if

**Theorem 2.** If a one-sided hash function acts closely to some randomly behaving oracle, the contributed protocol shall remain protected of malicious adversary in case the latter attempts to intercept the parameters on insecure channel and compute a valid session key $SK_i$.
Table 2
A comparison of schemes on the basis of security features

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Ensuring anonymity</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Supports mutual authentication</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Immune to Insider Attack</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Immune to Offline password-guessing threat</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Immune to Stolen smart card threat</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Immune to Impersonation threat</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Immune to KCI threat</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Supports session key security</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Immune to trace attack</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
<tr>
<td>Reparability</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Efficient Password Modification</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table 3
The computational cost of schemes

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Login &amp; Authentication phase</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Server side</td>
<td>7 T_h</td>
<td>9 T_h</td>
<td>8 T_h</td>
<td>8 T_h</td>
<td>8 T_h</td>
</tr>
<tr>
<td>User side</td>
<td>9 T_h</td>
<td>10 T_h</td>
<td>10 T_h</td>
<td>11 T_h</td>
<td>11 T_h+3 T_H</td>
</tr>
<tr>
<td>RC</td>
<td>0 T_h</td>
<td>5 T_h</td>
<td>0 T_h</td>
<td>0 T_h</td>
<td>0 T_h</td>
</tr>
<tr>
<td>Total</td>
<td>16 T_h</td>
<td>24 T_h</td>
<td>18 T_h</td>
<td>19 T_h</td>
<td>19 T_h+3 T_H</td>
</tr>
<tr>
<td>Computation delay (s)</td>
<td>0.008</td>
<td>0.012</td>
<td>0.009</td>
<td>0.0095</td>
<td>0.0395</td>
</tr>
<tr>
<td>Energy (µJ)</td>
<td>12.16</td>
<td>14.44</td>
<td>18.24</td>
<td>13.68</td>
<td>16.72</td>
</tr>
</tbody>
</table>

Proof. In this proof, the attacker A having approached the publicly available parameters as \{PID, ZID_i, Bij, β_i, Qij, T_i, β_i, Qij\}, might use the oracle Reveal1 to implement algorithm ALG2^{HASH}_{SPMSAC}. The probability for the success of ALG2^{HASH}_{SPMSAC} amounts to Susps2=Pr.2[ALG2^{HASH}_{SPMSAC} = 1] - 1, where Pr[E_{α2}] characterizes probability for an event E_{α2}. The advantage function for algorithm ALG2^{HASH}_{SPMSAC} is referred to as Adv_fun^{HASH}_{SPMSAC}(t_{α2}, q_{Ry2}) = \max_{α} Susps2^{HASH}_{SPMSAC}, with the execution time t_{α2}, while the corresponding random query q_{Ry2} is maximized on adversary (A) [15-16, 27]. We could safely refer to the contributed protocol as protected against the adversary A so it might not derive the convincing session key SK_{α}, provided Adv_fun^{HASH}_{SPMSAC}(t_{α2}, q_{Ry2}) ≤ ρ for any adequately small ρ > 0 [24-25, 48]. According to this testing algorithm, if the adversary A is able enough to reverse the one-sided hash function h(·), and decipher the oracle, it might recover the valid session key SK_{α}, for that ID_i, and eventually wins the game. Nonetheless, keeping in view the above definition, this would be computationally impractical to invert hash function [20-22, 36-37, 50] Adv_fun^{HASH}_{SPMSAC}(t_{α2}) ≤ ρ for any adequately small ρ > 0.
7. Comparison and Performance Evaluation

In this section, we evaluate the strength for proposed protocol with other MSA-based protocols [11, 18, 33, 38]. Table 2 shows the security features and the analysis of resistance to various threats for various schemes, which signifies the contributed scheme as a resilient authenticated key agreement in contrary to previous schemes. For comparing the costs, we depict the hash-digest operation with $T_h$ and bio-hashing with $T_H$ and ignoring XOR function due to a quite negligible cost as shown in Table 3. The comparison in Table 2 is shown for Liao and Wang [38], Hsiang and Shih [18], Lee et al. [33], Chen and Lee [11], and our proposed scheme. Therefore, in view of the current performance evaluation, we infer that the proposed model is quite more secure than Liao and Wang, Hsiang and Shih, Lee et al., and Chen and Lee’s schemes. All of these schemes are based on light weight hashed based symmetric cryptography. The contributed protocol bears a bit higher computation cost than Liao and Wang, and Lee et al., and Chen and Lee’s schemes, but provides more security. Moreover, our scheme achieves the required security objectives in less cost than Hsiang and Shih’s scheme. Adding a bit extra and negligible cost, the proposed protocol is immune to insider attack, password guessing attack, stolen smart card attack, impersonation and trace attacks as compared to previous schemes. Comparing on the same lines and taking the computation delay of hash function as 0.0005s and bio-hash operation as 0.01s, the cost of Liao and Wang, Lee et al., Chen and Lee, Hsiang and Shih, and proposed scheme amounts to 0.008s, 0.009s, 0.0095s, 0.012s, and 0.395s, respectively. Furthermore, the protocols may also be analyzed on account of energy consumptions by taking the energy cost of computation for SHA-1 as 0.76$\mu$J against single byte [43]. Following this, the energy consumption for the Liao and Wang’s, Lee et al.’s, Chen and Lee’s, Hsiang and Shih’s and contributed protocol amounts to 12.16, 13.68, 14.44, 18.24, and 16.72$\mu$J, respectively. As obvious from Table 3, the proposed protocol has a bit more cost than other related schemes; this is for the reason that it makes a use of bio-hash function, which increases the cost of the proposed protocol from 0.01s to 0.395s. This paper makes the proper use of biometric input by employing bio-hash function, unlike previous schemes. Hence, in view of the current performance evaluation, we can safely deduce that our proposed protocol is secure enough as compared to other schemes being analyzed, and achieves this objective in almost an equivalent cost.

8. Conclusion

The multi-server authentication robustness is considered as a crucial requisite of the existing remote authentication paradigm. Much of the research efforts can be witnessed to strengthen multi-server authentication protocols, lately. This paper critically examines the Chen and Lee’s multi-server authentication protocol. The Chen and Lee’s protocol has been found recently susceptible to few attacks. Its cryptanalysis suggests the three ways where the Chen and Lee’s protocol could be attacked or termed as inefficient. The Chen and Lee’s scheme was found defenseless to impersonation attack, trace attack, stolen smart card attack exposing session key, key-compromise impersonation attack and inefficient password modification. The proposed study identified these attacks and also demonstrated an improved version countering the identified threats. This paper is complemented with formal security analysis and performance evaluation analysis among different schemes.

References


