An improved lightweight multi-server authentication scheme with automated analysis

Azeem Irshad[1], Shehzad Ashraf Chaudhary[1], Saru Kumari[2], Muhammad Usman[3], Khalid Mahmood[1], Muhammad Shahzad Faisal[1]

irshadazeem2@gmail.com, shahzad@iiu.edu.pk, saryusiirohi@gmail.com, musman@qau.edu.pk , khalid_mahmood@iiu.edu.pk, ch_muhammad@hotmail.com

1 *Computer Science department, International Islamic University, Islamabad*

2 *Chaudhary Charan Singh University, Meerut 250004 Uttar Pradesh, India*

3 *Department of Computer Science, Faculty of Natural Science, Quaid-I-Azam University, Islamabad, Pakistan*

Abstract.
Multi-server authentication complies with the up-to-date requirements of internet services and latest applications. The multi-server architecture enables the expedient authentication of subscribers on an insecure channel for the delivery of services. The users rely on a single registration of a trusted third party for the procurement of services from various servers. Recently, Chen and Lee, Moon et al. and Wang et al. presented multi-server key agreement schemes, which are found to be vulnerable to many attacks according to our analysis. The Chen and Lee scheme was found susceptible to impersonation attack; trace attack, stolen smart card attack exposing session key, key-compromise impersonation attack and inefficient password modification. The Moon et al. is susceptible to stolen card attack leading to further attacks, i.e. identity-guessing, key-compromise impersonation attack, user impersonation attack, and session keys disclosure. While, Wang et al. is also found to be prone to trace attack, session-specific temporary information attack, key-compromise information attack, and privileged insider attack leading to session key disclosure and user impersonation attacks. We propose an improved protocol countering the indicated weaknesses of these schemes in an equivalent cost. Our scheme demonstrates automated and security analysis based on BAN logic, and also presents the performance evaluation for related schemes.

Keywords: Multi-server authentication, remote authentication, biometrics, attacks,

1. INTRODUCTION

Multi-server authentication fulfills the modern-age requirements of internet-based services, in comparison with single-server authentication. Multi-server authentication (MSA) enables the verification of users for various services out of a single registration. The MSA environment is beneficial to both users and servers equally, since the users are relieved of memorizing multiple passwords that would, otherwise, be required for each service it registers. At the same time, the MSA environment relieves the servers of performing separate registrations for every user. The MSA architecture involves three participating entities, i.e. user, servers (also termed as service providers), and registration centre (RC). In the initialization stage, the RC being a trusted third party registers the users and servers employing confidential paths. Thereafter, the users could get the stipulated services directly from servers after mutual authentication phase on insecure channel. Alternatively, the trust, in MSA environment, transfers from RC towards user and servers.

The first simple authentication scheme was presented by Lamport in 1981 [1]. Then, these schemes evolve from password-based schemes to smart-card [2], biometric-based schemes and ultimately towards multi-server authentication schemes [3-6]. Since a decade, we witnessed several multi-server authentication techniques. Yet, the practical implications call for presenting more computationally efficient and secure MSA protocols. In this connection Li et al. [3] presented a pioneer multi-server authentication scheme for neural networks. However, according to Lin et al. [4], the Li et al. scheme takes much time to train neural networks, and presented an improved protocol embedding ElGamal digital signature and geometric features on the Euclidean plane. Next, Juang [5] proposed a symmetric cryptography-based MSA scheme, but it has scalability issues due to the maintenance of verifier table on the end of server for every user. Afterwards, Tsaur [6] put forward a remote user-based authentication protocol relying on RSA cryptography and Lagrange interpolating polynomial. It is worthy to note that there have been presented many public key cryptography (PKC) -based schemes for MSA [6, 11-23] though, the symmetric key schemes are still preferable for low-end mobile devices with scarce resources. Following the pace on symmetric crypto-based light weight protocols, Chang and Lee [7] presented another MSA based scheme which was found exposed to insider attacks, and server and RC spoofing attacks [8]. Liao and Wang [8], afterwards, presented a remote user dynamic ID based authentication scheme for MSA framework. Then, Hsiang and Shih [9] found the scheme [8] to be vulnerable for masquerading and insider attacks and also presented an improved scheme. Lee et al. [10] found that [9] does not provide mutual authentication, and presented its own improved protocol. However, Chen and Lee [24] found that [10] does not provide smart card based two-factor security, and suffers masquerading attack. Besides, that scheme utilize an inefficient password updating procedure that involves RC each time, the password is changed. After discovering weaknesses in [10], Chen and Lee scheme also presented an improved scheme. After a careful analysis of the Chen and Lee's protocol [24], we observe that the scheme is prone to stolen smart card attack that may further lead towards password and session key disclosure. The scheme is also susceptible to impersonation and trace attacks. Besides, the protocol [24] undergoes a faulty password modification procedure. Recently, Moon et al. [25] and Wang et al. [26] presented multi-server authenticated key agreement schemes, which are found to be prone to many attacks according to our analysis. The Moon et al. is prone to privileged insider attack, identity-guessing attack, and session key disclosure. While, Wang et al. is found to be vulnerable to trace attack, session specific temporary information attack, key-compromise information attack and privileged insider attack. The current study work reviews Chen and Lee, Moon et al. and Wang et al. schemes [24-26] with the demonstration of working and cryptanalysis. Finally it presents an improved protocol version including formal security analysis. Moreover, the protocol has been incorporated by automated tool analysis and BAN logic-based security analysis.

The section 2 relates to preliminaries defining hash function and bio-hashing. Section 3 takes into account the reviews of Chen and Lee scheme, Moon et al. and Wang et al. schemes. Section 4 discusses the proposed model. The section 5 presents informal security analysis. Section 6 exhibits automated analysis, formal analysis and performance evaluation. Section 7 summarizes the paper findings.

2. PRELIMINARIES

The preliminary section describes properties of hash and bio-hashing functions as used in the proposed contribution.

2.1 Multi-server authentication architecture

In Multi-server authentication (MSA) architecture [49-50, 56-59], the users get registered through a centralized control centre. Thereafter, the users may get services of authorized service providers without re-registration. However, the users must perform mutual authentication procedure to qualify for service provision. Unlike, single server authentication, the MSA architecture relieves the subscribers of registrations from multiple service providers separately. The MSA environment embraces three interacting entities, that is, user (Ui), service providers (Sj), and registration centre (RC) as shown in figure 1. The RC acting as a centralized control centre, registers all subscribers and servers on confidential channels in initializing phase. This lets the subscribers to get the services from servers either directly without getting RC engaged, or indirectly by engaging RC in mutual authentication phase. Alternatively, we can say that trust is transferred from RC towards all entities subject to RC, since, the former acts as a trusted third party to authenticate the entities (users and servers).

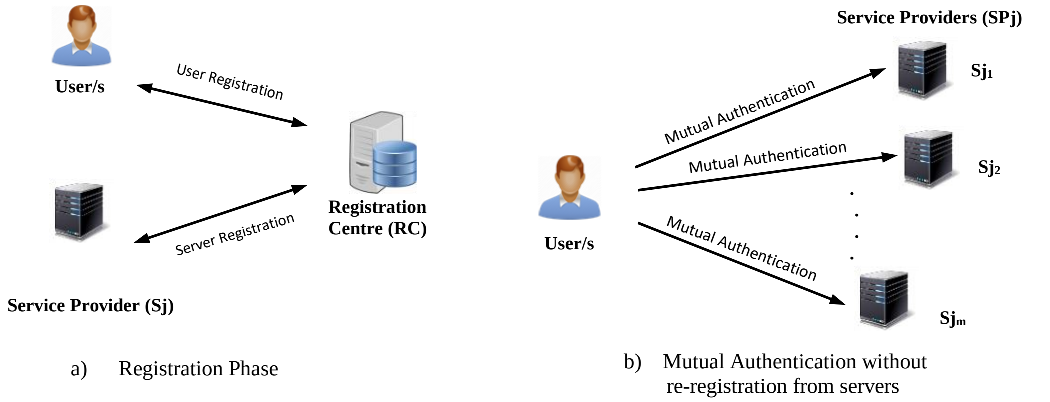

Figure 1. MSA-architecture

2.2 Hash function

we describe the properties of a secure one-way hash function, i.e. h:$\{0,1\}^* \rightarrow \{0,1\}^\ell$, where ℓ represents a secure length, that generates a y' string of fixed length as output, by taking a variable length string x' as input, i.e., $y' = h(x')$, as following:

1. It is a hard problem to modify the message m without modifying the digest $h(m)$.
2. It is intractable to create a message m that generates $h(m)$ as preimage resistance.
3. It is intractable to find the numbers m_1 and m_2 such that m_1 is not equal to m_2 while $h(m_1)$ equates $h(m_2)$ simultaneously.

2.3 Bio-hashing

The biometric parameters *BIOi* behaves somehow, in a different manner, every time these are collected. The biohash *H(.)* function generates a compact set of codes, for the user after bringing randomness, by introducing random salt in the function. Alternatively, *H(.)* transforms the extracted finger or facial codes *F* along with the random salt into biocodes *B*, while the hamming distance is used to distinguish the two biocodes. In this manner, the use of bio-hashing may comfortably thwart the de-synchronization attack, along with other attacks [52-53].

2.4 Attack model

An attacker is supposed to be having following capabilities [27-35].

1. An attacker may steal the smart card contents by power analysis and reverse engineering procedures.
2. An adversary may intercept, eavesdrop, modify, and replay messages over a public channel.
3. An adversary might be an insider i.e. legitimate user or a server having malicious intentions.
4. An adversary might guess a low entropy password and identity of a user.

3. WORKING AND LIMITATIONS IN CHEN AND LEE's, MOON *ET AL.*'s, AND WANG *ET AL.*'s SCHEMES

Since, the three multi-server authentication schemes, i.e. Chen and Lee, Moon et al. and Wang et al. share a single property of a secret key sharing, in which a registration authority shares a single secret with all service providers. On the basis of that shared secret, the service providers verify the authenticity of a subscriber. All of the three schemes have been reviewed in this study work. The working and cryptanalysis details for these schemes are described below:

3.1 Chen and Lee scheme

This section presents the design and limitations of Chen and Lee scheme [24] as illustrated below:

3.1.1 Working of Chen and Lee scheme

In Chen and Lee scheme, a trusted RC registers the servers *Sj* by issuing a unique secret PID_j using secure channel. The Chen and Lee scheme comprises three phases, i.e. Registration phase, Login and Authentication phase, as depicted in Figure 2.

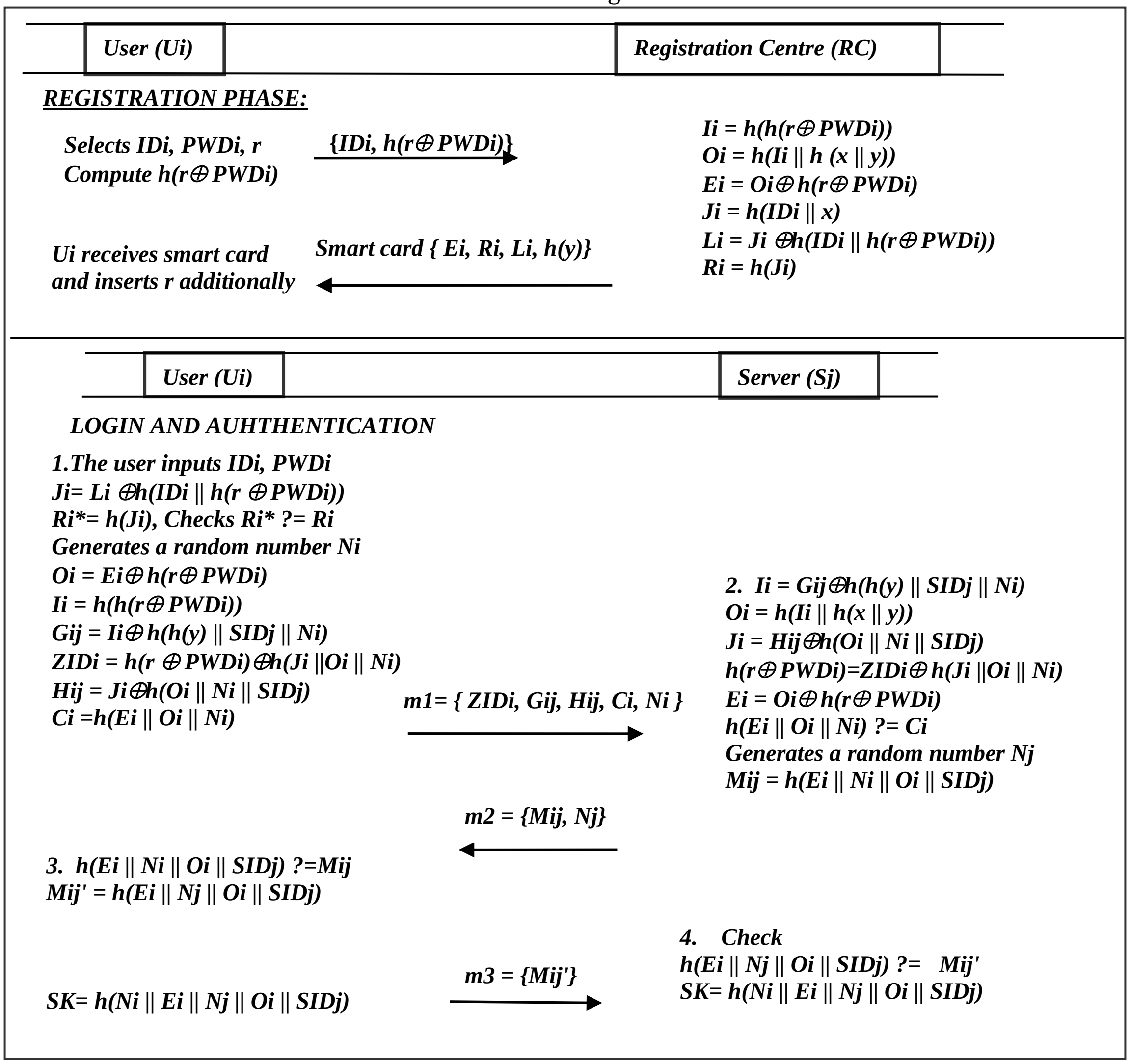

Figure 2. Chen and Lee model Registration, Login and Authentication phase

a) The Registration Phase

The user *Ui* gets registered by adopting the under-mentioned procedure with registration centre:

1. Firstly, the user selects *IDi*, *PWDi*, and also generates a random integer *r*. Then, it computes *h(r⊕ PWDi)*, and submits {*IDi*, *h(r⊕ PWDi)*} to RC for the purpose of registration.
2. *RC*, then computes *Ii = h(h(r⊕PWDi))*, *Oi = h(Ii||h(x||y))*, *Ei= Mi⊕h(r⊕PWDi)*, *Ji = h(IDi ||x)*, *Ji=h(IDi||x)*, *Li =Ji⊕h(IDi||h(r⊕ PWDi))* and *Ri = h(Ji)*. Next, *RC* stores in smart card *{Ei, Ri, Li, h(y)}* and sends towards user.
3. *Ui* gets smart card and in addition, stores the parameter *r* in it.

b) The Login and Authentication procedure

1. In login phase the user gets authenticated access from *Sj* through *RC*. For this reason

the user inputs its identity IDi and password $PWDi$. Then SC computes $Ji= Li \oplus h(IDi \,||\, h(r \oplus PWDi))$, $Ri^*= h(Ji)$, and checks the equation $Ri^* ?= Ri$. If true, then generates a random integer Ni to further compute $Oi = Ei \oplus h(r \oplus PWDi)$, $Ii= h(h(r \oplus PWDi))$, $Gij = Ii \oplus h(h(y) \,||\, SIDj \,||\, Ni)$, $ZIDi = h(r \oplus PWDi) \oplus h(Ji \,||Oi \,||\, Ni)$, $Hij = Ji \oplus h(Oi \,||\, Ni \,||\, SIDj)$, $Ci =h(Ei \,||\, Oi \,||\, Ni)$ and sends the message $m1= \{ ZIDi, Gij, Hij, Ci, Ni \}$ to RC.

2. Sj receives the request $m1= \{ ZIDi, Gij, Hij, Ci, Ni \}$ and computes $Ii= Gij \oplus h(h(y) \,||\, SIDj \,||\, Ni)$, $Oi = h(Ii|| \, h(x \,||\, y))$, $Ji = Hij \oplus h(Oi \,||\, Ni \,||\, SIDj)$, $h(r \oplus PWDi)= ZIDi \oplus h(Ji \,||\, Oi \,||\, Ni)$ and $Ei = Oi \oplus h(r \oplus PWDi)$. Next, it compares the equation $h(Ei \,||\, Oi \,||\, Ni) ?= Ci$. If it holds true, it further generates a random integer Nj to compute $Mij = h(Ei \,||\, Ni \,||\, Oi \,||\, SIDj)$ and submits the message $m2 = \{Mij, Nj\}$ to Ui to proceed for authentication.
3. Next, the user Ui constructs $h(Ei \,||\, Ni \,||\, Oi \,||\, SIDj)$ and then compares the equation $h(Ei \,||\, Ni \,||\, Oi \,||\, SIDj) ?=Mij$. If true then computes $Mij' = h(Ei \,||\, Nj \,||\, Oi \,||\, SIDj)$ and this $m3 = \{Mij'\}$ to RC for final verification with Nj based challenge.
4. The RC receives $m3$ and checks equality $h(Ei \,||\, Nj \,||\, Oi \,||\, SIDj) ?= Mij'$ after computing $h(Ei \,||\, Nj \,||\, Oi \,||\, SIDj)$. If the match occurs, it finally develops the session key with user as $SK= h(Ni \,||\, Ei \,||\, Nj \,||\, Oi \,||\, SIDj)$.

3.1.2 Weaknesses in Chen and Lee scheme.

The Chen and Lee scheme is found susceptible to stolen card attack, user impersonation attack, trace attack, key-compromise impersonation attack and costly password modification phase as described below.

a) Stolen smart card Attack

An attacker Ă could launch a stolen smart card attack, if it happens to approach the card accidentally [51]. As the smart card bears the $\{Li, Ei, Ri, h(y)\}$ and the publicly available messages are $m1= \{ ZIDi, Gij, Hij, Ci, Ni \}$, $m2 = \{Mij, Nj\}$ and $m3 = \{Mij'\}$. Since Ni and $SIDj$ are publicly accessible, and $h(y)$ could be approached from stolen card. Then, an adversary could construct $h(h(y) \,||\, Ni \,||\, SIDj)$ and access the Ii^* parameter by computing $Ii^* = Gij \oplus h(h(y) \,||\, Ni \,||\, SIDj)$. Next, due to the availability of *'r'* random number in SC, it could launch an offline dictionary attack for guessing the right password. It tries all dictionary combinations of $PWDi^*$ and match with $Ii^* = h(h(r \oplus PWDi^*)$ repeatedly by computing and checking the equation $h(h(r \oplus PWDi^*) ?= Ii^*$. Wherever it matches, there comes the right password for adversary.

After guessing the password $PWDi$ it may compute $h(r \oplus PWDi)$, and Oi' by performing $Oi'= h(r \oplus PWDi) \oplus Ei$. Next, it could easily generate the legitimate session key by implementing the hash function as $h(Ni \,||\, Ei \,||\, Nj \,||\, Oi' \,||\, SIDj)$. This way, an adversary guesses the shared session key SK between the participants by stealing the smart card. Hence the scheme is susceptible to stolen card attack.

b) User impersonation Attack

The Chen and Lee scheme is susceptible to user impersonation attack, subject to the availability to SC contents. Using SC contents Ă may construct a valid $PWDi$ according to the procedure defined above. Next, Ă computes $Oi = Ei \oplus h(r \oplus PWDi)$ and $Ii = h(h(r \oplus PWDi))$. Next, it guesses IDi by trying all of the possible strings IDi^* using these two statements, $Ji^*= Li \oplus h(IDi^*||h(r \oplus PWDi))$ and $Ri ?= h(Ji^*)$, repeatedly. If the equality hits, then the valid IDi and Ji^* are located. Next, it assumes a random number Ni and computes $Gij = Ii \oplus h(h(y)||Ni \,||SIDj)$, $ZIDi = h(r \oplus PWDi) \oplus h(Ji \,||Oi|| \, Ni)$, $Hij =Ji \oplus h(Oi \,||\, Ni \,||\, SIDj)$ and $Ci =h(Ei||Oi||Ni)$. Finally, it constructs login request message $m1= \{ZIDi, Gij, Hij, Ci, Ni \}$ successfully.

c) Trace Attack

In a trace attack, an adversary may trace the consistency among various sessions created between the same participants in different periods of time. In Chen and Lee, a malicious insider, having the knowledge of $h(y)$, may intercept the message $m1=\{ ZIDi, Gij, Hij, Ci, Ni \}$ and attempt to find the symmetry among various sessions by finding Ii after computing $Ii = Gij \oplus h(h(y) \,||\, SIDj \,||\, Ni)$. The Ii parameter remains the same for all sessions established between the Ui and Sj, until the PWi or r are changed in smart card. Hence, the Chen and Lee scheme is susceptible to trace attack.

d) Key-Compromise Impersonation attack (KCI)

In this attack, an adversary may use the recovered or stolen secret parameter of a user to masquerade it as a server. The Chen and Lee scheme is susceptible to KCI attack, once the smart card contents are stolen by an attacker. After password recovery of user, as shown in sub-section *3.1.2* (*a*), the adversary may easily masquerade as server by constructing the message *m2 = {Mij, Nj}* after generating a random number *Nj*, and computing *Mij** as *Mij* =h(Ei||Nj||Oi|| SIDj)*. Since, *Ei* and *Vi* parameters can be constructed by manipulating SC parameters as shown in section *3.1.2 (b)*. This message *m2* will be sent towards user, which will be duly verified by user, though fake. In this manner, a successful masquerading attack can be initiated against user in Chen and Lee scheme.

e) No session key security

Once, the parameters *Ei* and *Oi* are recovered by an adversary using stolen smart card contents, it may compute all previous session keys by intercepting *Ni*, *Nj* and constructing the session key as *SK= h(Ni || Ei || Nj || Oi || SIDj).*

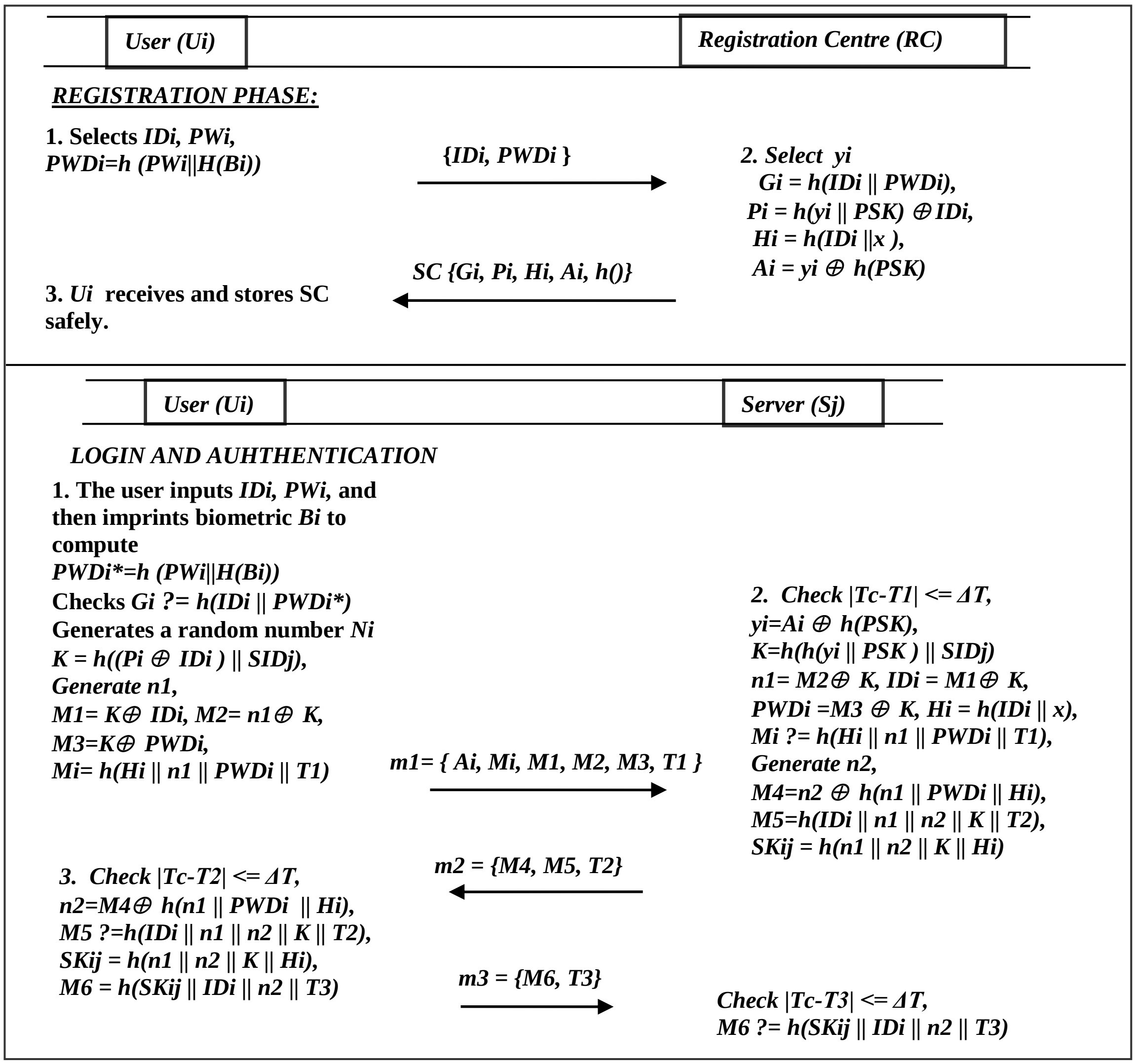

Figure 3. Moon et al. registration and login & authentication phase

f) No direct password modification

The author claims that *Ui* does not resort to *RC* for changing the password, however, the current Chen and Lee scheme has no way for the user to modify the *PWDi* without engaging *RC*. The password modification involves the update of *Ei = Oi ⊕ h(r⊕PWDi)*, every time the *PWDi* is changed. While, *Ri* is used for the construction of *Oi*, as *Oi = h(Ii || h (x || y))*. Additionally, the parameter *Ii* is a function of the password as *Ii = h(h(r⊕ PWDi))*. Hence, the *Ui* will have to resort to *RC* each time, for Ii update for not having the knowledge of $h(x||y)$. This proof nullifies the author's claim of modifying the password without *RC's* engagement.

3.2 Moon et al. scheme

The Moon et al. scheme presents an improved biometric multi-server authentication protocol after finding attacks in Lu et al. [46]. This section presents the design and limitations in Moon et al. scheme [25] as depicted below:

3.2.1 Protocol design of Moon et al.

In Moon et al., the RC registers Sj by sending secret parameter PSK and secret number x using a confidential channel. The scheme comprises two phases notably, Registration and Login & Authentication phase, as depicted in figure 3.

a) The Registration procedure

The user enrolls with RC for registration to perform the under-mentioned steps:

1. Ui selects IDi, PWi, and calculates $PWDi=h(PWi||H(Bi))$. Then, it submits the request {$IDi, PWDi$} towards RC.
2. RC, after receiving {IDi, $PWDi$}, generates yi and computes $Gi = h(IDi \,||\, PWDi)$, $Pi = h(yi \,||\, PSK) \oplus IDi$, $Hi = h(IDi \,||x$ and $Ai = yi \oplus h(PSK)$, Then, it stores Gi, Pi, Hi, Ai in smart card. Next, it forwards the updated smart card towards Ui.

b) Login and Authentication procedure

1. In this phase the user initiates the procedure for having authenticated access from Sj directly. To serve the purpose, the Ui proceeds to input its IDi, PWi and imprint biometric Bi. Next, the smart card computes $PWDi^{*}=h\ (PWi||H(Bi))$ and checks the equation $Vi\ ?=\ h(IDi \,||\, PWDi^{*})$. If it holds true, then further constructs $K=\ h((Pi\oplus IDi)||\ SIDj)$ and defines a random integer as $n1$, and further computes $M1= K\oplus IDi$, $M2= n1\oplus\ K$, $M3=K\oplus\ PWDi$ and $Mi= h(Hi \,||\, n1 \,||\, PWDi \,||\, T1)$. Next, it submits the message $\{Ai, Mi, M1, M3, M2, T1\ \}$ towards Sj.
2. Sj receives the request and checks $|Tc\text{-}T1| <= \Delta T$. If the difference is less than threshold, it further computes $yi=Ai \oplus\ h(PSK)$, $K=h(h(yi \,||\, PSK\ ) \,||\, SIDj)$, $n1= M2\oplus\ K$, $IDi = M1\oplus\ K$ and $PWDi =M3 \oplus\ K$, $Hi = h(IDi \,||\, x)$. Next, it verifies the equality for $Mi\ ?=\ h(Hi \,||\, n1 \,||\, PWDi \,||\, T1)$. If true, then it validates the user, and further generates $n2$, and computes $M4=n2 \oplus h(n1 \,||\, PWDi \,||\, Hi)$, $M5=h(IDi \,||\, n1 \,||\, n2 \,||\, K \,||\, T2)$ and $SKij = h(n1 \,||\, n2 \,||\, K \,||\, Hi)$. Next, it sends the message $\{M4, M5, T2\}$ towards Ui.
3. Next, the Ui receives the message $\{M4, M5, T2\}$ and matches the timestamp difference against threshold. If it is valid, then it computes $n2=M4\oplus\ h(n1 \,||\, PWDi \,||\, Hi)$, and verifies $M5\ ?=h(IDi \,||\, n1 \,||\, n2 \,||\, K \,||\, T2)$. It validates Sj on the positive match. Next, it computes $SKij = h(n1 \,||\, n2 \,||\, K \,||\, Hi)$, and $M6 = h(SKij \,||\, IDi \,||\, n2 \,||\, T3)$. Finally it sends $M6$ towards Sj for further verification.
4. Sj receives the message and matches the timestamp with threshold. If it holds true, further computes and verifies the equality $M6\ ?=\ h(SKij \,||\, IDi \,||\, n2 \,||\, T3)$ to finally validate the user.

3.2.2 *Weaknesses in Moon et al.*

The Moon et al.'s protocol has been discovered as susceptible to identity guessing attack, and once identity is guessed, the user becomes vulnerable to many sorts of other attacks, e.g., impersonation attack and session keys guessing attack.

a) Stolen smart card leading to Identity guessing Attack

The identity IDi of a user, being a low entropy string just like a low entropy password, can be guessed in polynomial amount of time by adopting the following procedure.

1. An adversary may extract the contents $\{Gi, Pi, Hi, Ai, h()\}$ of a stolen smart card by using differential power analysis [54]. At the same time it may also intercept the messages $M1, M2$ and $M3$, i.e $M1= K\oplus\ IDi$, $M2= n1\oplus\ K$, $M3=K \oplus PWDi$. Next, it may attempt many combinations of the selected IDi^{*} and compute the following parameters.

$$K^{*} = M1\oplus\ IDi^{*} \quad (1)$$

$$n1^{*}= M2\oplus\ K^{*} \quad (2)$$

$$PWDi^{*} =K^{*}\oplus\ M3 \quad (3)$$

$$h(Hi \,||\, n1^{*} \,||\, PWDi^{*} \,||\, T1)\ \ ?= Mi \quad (4)$$

The adversary keeps checking different combinations of IDi^{*} until the equation (4) holds.

Once a valid *IDi* string is guessed, it might easily compute other parameters as well, i.e. *K*, *n1* and *PWDi*. After guessing these parameters, an adversary might be in a strong position to launch user impersonation attack, Key-Compromise Impersonation attack (server masquerading attack), and may even recover all previous session keys as elaborated below.

b) User impersonation attack

In case, an adversary accesses the *IDi*, *PWDi*, *Hi* and *K* parameters as described above, it may launch user impersonation attack by constructing a new authentication request message $m1= \{Ai, Mi, M1, M2, M3, T1\}$ by generating a novel random secret n_a and computing $M1= K\oplus IDi$, $M2= n_a \oplus K$, $M3=K\oplus PWDi$ and $Mi= h(Hi \| n_a \| PWDi \| T1)$. Next, it submits the message $\{Ai, Mi, M1, M3, M2, T1\}$ towards *Sj*. Following all the steps as defined in sub-section *3.2.1(b)*, it may construct the final verification message $\{M6\}$ and send towards *Sj* impersonating *Ui*, which will be verified by *Sj* successfully, however fake.

c) Key-Compromise Impersonation attack

An adversary, after guessing and accessing the parameters $\{IDi, PWDi, Hi, K\}$ may construct the message $m2 = \{M4, M5, T2\}$, whereas $M4= n_b \oplus h(n1 \| PWDi \| Hi)$, $M5=h(IDi \| n1 \| n_b \| K \| T2)$, n_b is fresh random number and *T2* is new timestamp. After constructing the message, it may forward to the legitimate user impersonating as a server and will be successfully verified by the user, however fake.

d) Session key security failure

After guessing and computing the parameters $\{IDi, PWDi, Hi, K\}$, an adversary may compute past session keys *SKij* by capturing the earlier messages and computing *n1* and *n2*, i.e.

$$n1 = M2\oplus K \qquad (5)$$

$$n2 = M4\oplus h(n1 \| PWDi \| Hi) \qquad (6)$$

In this way, it may construct all previous session keys by computing $SKij = h(n1 \| n2 \| K \| Hi)$. Hence, the Moon et al.'s protocol is prone to session key security attack.

3.3 Wang et al. protocol

The Wang et al. depicts an improved biometric multi-server authentication protocol after finding drawbacks in Mishra et al. [47]. This section presents the design and limitations of Wang et al. protocol as illustrated below:

3.3.1 Protocol design of Wang et al.

The Wang et al. protocol is composed of two phases, i.e. Registration and Login & authentication phase, as depicted in Figure 4. The server gets registered through *RC* using a shared secret *PSK* on a secure channel.

a) The Registration Phase

In registration phase, the user performs registration procedure with *RC* by adopting the following steps:

1. The *Ui* inputs its identity *IDi*, password *PWi*, imprints *Bi*. Thereafter, it calculates $Gen(Bi)\rightarrow (R_i, P_i)$, $RPBi =h(PWi \| R_i)$ and submits $\{IDi, RPWi\}$ to RC on a secure channel.
2. *RC*, initially stores $<IDi, Ni =1>$ in its credential table for maintaining the status of non-revoked subscriber, which may be updated to $<IDi, Ni =0>$, whenever *Ui* wants to revoke its registration in future. Next, *RC* calculates $Wi = h(IDi \|x\|T_r)$, $Xi = RPBi \oplus h(Wi)$, $Yi = Xi \oplus h(PSK)$, $Zi = PSK \oplus Wi \oplus h(PSK)$, and $Qi = h(IDi \| RPBi)$, while T_r represents registration time. Finally, *RC* stores *Xi*, *Yi*, *Zi*, *Qi* in *SC*, and forwards to user by using a confidential channel. Where *PSK* is a shared secret among *RC* and all servers.
3. The user receives smart card and stores P_i in it finally.

b) Login & Authentication Phase

1. In this phase, the user seeks verified access of servers directly without *RC*. To meet the objective, *Ui* enters its identity *IDi*, password *PWi*, then it imprints *Bi* to compute $Rep(B_i, P_i)\rightarrow(R_i)$. Then, it constructs $RPBi =h(PWi\|R_i)$ and verifies the equality $h(IDi \| RPBi) ?= Qi$. If it holds true, then it further computes $h(PSK)= Xi \oplus Yi$, generates a random number *r1*, and computes $CIDi = IDi \oplus h(r1)$, $M1= RPBi \oplus r1 \oplus h(PSK)$ and $M2= h(CIDi \| r1 \| RPBi \| SIDj \| Ti)$. Finally, it sends the message $m1= \{CIDi, M1, M2, Xi, Zi, Ti\}$ using insecure channel towards Sj for verification.

2. Sj, after receiving the message, checks the difference of timestamps against the threshold by comparing $Ti - Tj <= \Delta T$. If true, then it further computes $Wi = PSK \oplus Zi \oplus h(PSK)$, $RPBi = Xi \oplus h(Wi)$, $r1 = RPBi \oplus M1 \oplus h(PSK)$ and verifies the equality for $h(CIDi \,||\, r1 \,||\, RPBi \,||\, SIDj \,||\, Ti) ?= M2$. If it holds true, then it generates $r2$ and computes $SKij = h(CIDi \,||\, SIDj \,||\, r1 \,||\, r2)$, $M3 = r2 \oplus h(CIDi \,||\, r1) \oplus h(PSK)$ and $M4 = h(SIDj \,||\, r2 \,||\, CIDi)$. Next, it submits $\{SIDj, M4, M3\}$ using insecure channel.
3. After receiving the message, the user computes $r2 = M3 \oplus h(CIDi \,||\, r1) \oplus h(PSK)$, $SKij = h(CIDi \,||\, SIDj \,||\, r1 \,||\, r2)$, $r1 = Xi \oplus M1 \oplus h(PSK)$. Then, it matches equality for $h(SIDj \,||\, r2 \,||\, CIDi\,) ?= M4$. If does not match, it aborts the session. Otherwise, it further computes $M5 = h(SKij \,||\, r1 \,||\, r2)$ and sends the message $m3 = \{M5\}$ towards Sj for verification.
4. Sj receives the message $M5$, and computes and verifies the equality for equation $M5 ?= h(SKij \,||\, r1 \,||\, r2)$. If it is true, it validates the user as a legitimate subscriber, and establishes the session key $SKij = h(CIDi \,||\, SIDj \,||\, r1 \,||\, r2)$ with it.

User (Ui) | **Registration Centre (RC)**

REGISTRATION PHASE:

1. Selects *IDi, PWi,* Imprints *Bi,*
Gen (Bi)→ (R_i, P_i)
Selects IDi, PWi
Computes RPBi =h(PWi || Ri)

{IDi, RPBi } →

Stores for revocation <IDi, Ni =1>,
Wi= h(IDi || x || T_r),
Xi= RPBi ⊕ h(Wi),
Yi= Xi ⊕ h(PSK),
Zi= PSK ⊕ Wi ⊕ h(PSK),
Qi= h(IDi || RPBi),
Stores Xi, Yi, Zi, Qi in Smart card

← ***SC { Xi, Yi, Zi, Qi, h()}***

5.*Ui* receives the *SC* and stores P_i additionally.

User (Ui) | **Server (Sj)**

LOGIN AND AUHTHENTICATION

1. The user inputs *IDi, PWi,* and imprints *Bi*
Rep (Bi, P_i)→ (R_i)
Computes RPBi =h(PWi || R_i),
Qi=h(IDi || RPBi), h(PSK)= Xi ⊕ Yi,
generates r1
CIDi = IDi ⊕ h(r1),
M1= RPBi ⊕ r1 ⊕ h(PSK),
M2= h(CIDi || r1 || RPBi || SIDj || Ti)

m1= { CIDi, M1, M2, Xi, Zi, Ti } →

2. Ti –Tj <= ΔT,
Retrieves Wi = PSK ⊕ Zi ⊕ h(PSK),
RPBi =Xi ⊕ h(Wi)
r1=RPBi ⊕ M1 ⊕ h(PSK)
h(CIDi || r1 || RPBi || SIDj || Ti) ?= M2
Generates r2
SKij =h(CIDi || SIDj || r1 || r2)
M3 = r2 ⊕ h(CIDi || r1) ⊕ h(PSK)
M4=h(SIDj || r2 || CIDi)

← ***m2 = {SIDj, M3, M4}***

3. r2= M3 ⊕ h(CIDi || r1) ⊕ h(PSK),
SKij =h(CIDi || SIDj || r1 || r2)
r1=Xi ⊕ M1 ⊕ h(PSK),
h(SIDj || r2 || CIDi) =M4
M5 = h(SKij || r1 || r2)

m3 = {M5} →

4. M5 ?= h(SKij || r1 || r2)

Figure 4. Wang et al. registration and login & authentication phase

3.3.2 Weaknesses in Wang et al. scheme.

The Wang et al. protocol has been found susceptible to trace attack, session specific temporary information attack, Key-Compromise Impersonation attack and insider attack. The details of the attacks are described below.

a) Trace Attack

An adversary may distinguish a particular user among other users, and identify its location on the basis of intercepted public parameters *{Xi, Zi}* which remains uniform in all authentication requests. Since, all authentication requests of a particular user at different locations are bound to contain the parameters *{Xi, Zi}*, the locations can be linked and traced with the occurrence of common parameters by the adversary. If an adversary is privileged insider, having the values *IDi* and *RPBi*, it may easily compute *CIDi*, *M1* and *M2*. In this manner, it may comfortably trace the linkages between *IDi* and locations, where the authentication requests were originated.

b) Session-specific temporary information Attack

In Wang et al. protocol, if a single session-specific temporary random number is accidentally exposed, then the adversary may recover not only current session key but all previous session keys in the following ways.

i. Exposure of current session key

If a single session-specific temporary random number *r1* is exposed, then a malicious insider (adversary) having access to *h(PSK)* may compute the current session by adopting the following steps.

- The adversary computes $r2 = M3 \oplus h(CIDi \mid\mid r1) \oplus h(PSK)$, assuming the adversary intercepts the parameter *CIDi* for the current session.
- Next, the current session key may be constructed by computing $SKij = h(CIDi \mid\mid SIDj \mid\mid r1 \mid\mid r2)$.

ii. Exposure of previous session keys

Once, *r1* is exposed, then the adversary having access to *h(PSK)* may recover all previous session keys by adopting the following steps.

- It computes $RPBi = M1 \oplus r1 \oplus h(PSK)$ out of a disclosed single session-specific variable *r1*.
- Next, it may compute other session-specific numbers $r1_j$ and $r2_j$, while $j = 1.....n$. (where *n* represents the number of sessions up to which the adversary could recover the variables and had intercepted the messages), for instance,

$$r1_j = RPBi \oplus M1 \oplus h(PSK) \qquad (7)$$

$$r2_j = M3 \oplus h(CIDi \mid\mid r1_j) \oplus h(PSK) \qquad (8)$$

- Next, it may compute the session key of corresponding computed parameters, $r1_j$ and $r2_j$, i.e.

$$SKij = h(CIDi \mid\mid SIDj \mid\mid r1_j \mid\mid r2_j) \qquad (9)$$

c) Key-Compromise Information Attack (KCI)

In Wang et al. protocol, an adversary on the compromise of a single session-specific random number once, may launch KCI attack and masquerade as a server by adopting the following steps.

1. According to sub-section 3.2.2 (b), on the compromise of *r1* random integer, the malicious insider may compute *RPBi.* Next it may compute $r1_j$ from another intercepted user's authentication request message *M1*, i.e.

$$r1_j = RPBi \oplus M1 \oplus h(PSK) \qquad (10)$$

2. Then, it further computes $M3 = r2 \oplus h(CIDi \mid\mid r1_j) \oplus h(PSK)$ and $M4 = h(SIDj \mid\mid r2 \mid\mid CIDi)$, while *r2* is a fresh random integer. Next, it sends the message *m2={SIDj, M3, M4}* towards *Ui* to masquerade as a server *Sj*.
3. That fake message will be successfully verified by the user, and the later will be treating the adversary as a valid server.

d) Insider attack, leading to session keys exposure

An insider, having the *RPBi* parameter which might be acquired during user registration procedure, may compute all previous session keys for that user, of which *RPBi* is recovered, by adopting the following procedure.

1. Since, the parameter *h(PSK)* is known to every user, hence any compromised user may disclose that parameter, which may be approached by an adversary. Further, the adversary may compute $r1 = RPBi \oplus M1 \oplus h(PSK)$ and $r2 = M3 \oplus h(CIDi \mid\mid r1) \oplus h(PSK)$.
2. Ultimately, it may construct the corresponding session key by computing $SKij = h(CIDi \mid\mid SIDj \mid\mid r1 \mid\mid r2)$.

4. PROPOSED MODEL

The multi-server environment comprises three participating entities, i.e. user (*Ui*), server (*Sj*), and the registration centre (*RC*). *RC* defines two secrets; one is master secret x and another simple secret y. Next, it computes $h(x||y)$ and shares with all the legal service providers *Sj*, using a confidential channel. Some symbols that describe the proposed model are depicted in Table I.

Table I: Description of notations

Notations	Description
Ui	i[th] User
IDi/PWDi	User's identity and password
Sj, SIDj	Server, Server's identity
RC	Registration centre
x , y	*RC's* master key and secret key
H(.)	Bio-hashing function
h(.)	a secure hash digest function
$E_k()/D_k()$:	Symmetric encryption/decryption
SKij	Session Key shared between Ui and S_j
$\|\|/\oplus$:	Concatenation, XOR function

The proposed model comprises three stages, i.e., user registration, login & authentication, and password update phase as described under:

4.1 The Registration Phase

In registration phase the user performs the under-mentioned steps with registration centre as following:

1. First, the user inputs the parameters *IDi*, *BIOi*, *PWi*, and generates random numbers r_1 and r_2. It then, computes $Y= h(H(BIOi) ||IDi || r_2)$, $TPW=h(PWi ||H(BIOi))$, and sends $\{IDi, Y, TPW\oplus r_1\}$ to registration centre for registration.
2. Then, *RC* constructs $A= h(IDi || x)$, $Vi = h(A || h(x || y))$, $W'= TPW \oplus r_1 \oplus Vi$, $Di'=A\oplus h(IDi ||Y)$ and $Fi = h(h(IDi ||Y))$. Further, it generates a random integer t and constructs $PIDi =E_{h(x || y)} (A || h(t))$. Next, *RC* stores in smart card $\{PIDi , Di', Fi, W', h()\}$ and submits to *Ui*.
3. *Ui* receives the *SC* and computes $W= W'\oplus r_1$, $Di=Di' \oplus h(IDi ||r_2)$ and $B_r= H(BIOi) \oplus r_2$. Then, it replaces W' with W, Di' with Di, and stores B_r in *SC* finally. The smart card now contains $\{PIDi , Di, Fi, W, B_r , h()\}$.

4.2 Login & Authentication Phase

1. In this phase, the user seeks authorized access to services from *Sj* through *RC*. To meet this objective, *Ui* inputs *IDi*, *BIOi*, *PWi*. Next, the smart card calculates $Y = h(H(BIOi) ||IDi || r_2)$ after extracting r_2 from B_r, and compares $Fi^* ?= h(h(IDi ||Y))$. If it holds true, then further computes $TPW=h(PWi || H(BIOi))$, $Vi =TPW\oplus W$ and $A= Di \oplus h(IDi || r_2) \oplus h(IDi ||Y)$. Then, it generates a random number *Ni*, and computes $ZIDi = h(PIDi || Vi || A || Ni)$. Next, it sends the message $m1=\{PIDi, ZIDi, Ni\}$ to Sj for verification.
2. *RC* receives the request $m1= \{ PIDi, ZIDi, Ni \}$ and computes $(A || h(t))= D_{h(x || y)}(PIDi)$, $Vi = h(A || h(x || y))$ and compares $ZIDi ?= h(PIDi || Vi || A || Ni)$. If the equation holds true, it generates random integer t' and *Nj*. Then, it calculates $PIDi' =E_{h(x || y)} (A || h(t'))$, $Ti = PIDi' \oplus h(PIDi || A || Vi)$ and $Mij = h(A|| Ti || Ni ||Nj|| Vi || SIDj)$. Finally, it submits the message $m2 = \{Mij, Ti, Nj\}$ to *Ui*.
3. After getting *m2*, *Ui* calculates $h(A || Ti || Ni || Nj || Vi || SIDj))$ and compares *Mij*. If it holds true, then further computes $SKij= h(A|| Ni || Nj || Vi || SIDj)$, $Mij' = h(SKij || A || Nj || Vi || SIDj)$ and sends the message $m3 = \{Mij'\}$ to *Sj* for final verification with *Nj* based challenge.. Besides, it also computes $PIDi' =Ti \oplus h(PIDi || A || Vi)$ and replace *PID* with *PIDi'* in its smart card.
4. The *Sj* receives *m3* and computes $SKij= h(A|| Ni || Nj || Vi || SIDj)$. Then, it checks the equality $h(SKij || A || Nj || Vi || SIDj) ?= Mij'$. If the above verification holds true, it establishes the session key with *Ui* as $h(A|| Ni || Nj || Vi || SIDj)$. We have highlighted some salient differences of our proposed scheme in Figure 5.

4.3 Password modification

The user updates its password by invoking this procedure, into fresh password PWi^{new} without seeking any help from *RC*. Its steps are given below:

1. The user puts its smart card into the SC reader and also inputs identity *IDi** along with password *PWi**. Then, it imprints the biometric identity *BIOi** into the scanner. Thereafter, the smart card calculates $Y=h(H(BIOi)||IDi||r_2)$ after extracting r_2, and

User (Ui)	*Registration Centre (RC)*

REGISTRATION PHASE:

1. Selects *IDi, PWi,* r_1, r_2,

Imprints *BIOi*, Computes
TPW=h(PWi || H(BIOi)),
Y=h(H(BIOi) ||IDi || r_2***)***

$\{IDi, Y, TPW \oplus r_1\}$ →

2. ***A= h(IDi || x)***
Vi = h(A || h(x || y))
$W' = TPW \oplus r_1 \oplus Vi$
Generate random number *t*
$PIDi = E_{h(x\,||\,y)}(A\,||\,h(t))$,
$Di' = A \oplus h(IDi\,||Y)$
Fi = h(h(IDi ||Y))

← $\{PIDi, Di', Fi, W', h()\}$
Secure channel

3. *Ui* computes $W = W' \oplus r_1$, $Di = Di' \oplus h(IDi\,||r_2)$, $B_r = H(BIOi) \oplus r_2$ **and replaces *W'* with *W*, *Di'* with *Di*, stores** B_r **in SC now containing** $\{PIDi, Di, Fi, W, B_r, h()\}$

User (Ui)	*Server (Sj)*

LOGIN AND AUHTHENTICATION

1. The user inputs *IDi, PWi,* and imprints *BIOi* in *SC* Then computes $Y = h\,(H(BIOi)\,||IDi\,||\,r_2)$ **after extracting** r_2 **from** B_r

Checks $Fi^* ?= h(h(IDi\,||Y))$
Generates *Ni*
Computes TPW=h(PWi || H(BIOi)),
$Vi = TPW \oplus W$,
$A = Di \oplus h(IDi||r_2) \oplus h(IDi\,||Y)$,
ZIDi = h(PIDi || Vi || A || Ni)

$m1 = \{PIDi, ZIDi, Ni\}$ →

2. $(A\,||\,h(t)) = D_{h(x\,||\,y)}(PIDi)$,
Vi = h(A || h(x || y))
ZIDi ?= h(PIDi || Vi ||A|| Ni)
Generates *t', Nj*
$PIDi' = E_{h(x\,||\,y)}(A\,||\,h(t'))$,
$Ti = PIDi' \oplus h(PIDi\,||\,A||\,Vi)$,
Mij = h(A|| Ti || Ni ||Nj || Vi || SIDj)

← $m2 = \{Mij, Ti, Nj\}$

3. h(A || Ti ||Ni ||Nj|| Vi || SIDj)) ?=Mij
SKij= h(A || Ni || Nj || Vi || SIDj),
Mij' = h(SKij || A || Nj || Vi || SIDj)

$PIDi' = Ti \oplus h(PIDi\,||\,A||\,Vi)$
Replace *PIDi* with *PIDi'* in *SC*

$m3 = \{Mij'\}$ →

4. Check
SKij= h(A || Ni || Nj || Vi || SIDj),
h(SKij || A || Ni || Vi || SIDj) ?= Mij'

Shared session key =SKij= h(A || Ni || Nj || Vi || SIDj)

Figure 5. Proposed model (Registration, and Login & authentication)

compares $Fi^* ?= h(h(IDi \| Y))$. If true, then moves to the next step.

2. The *SC*, then computes $TPW=h(PWi \| H(BIOi))$ and $Vi = TPW \oplus W$.
3. Next, the *Ui* inputs a new password PWi^{new} and the SC further computes $TPW^{new}=h(PWi^{new} \| H(BIOi))$, $W^{new} = Vi \oplus TPW^{new}$.
4. Next, the value W is replaced with W^{new} in the smart card.

5. SECURITY ANALYSIS

This section comprises automated security verification using ProVerif tool [55] and security analysis using BAN logic [47-48] as following.

5.1 Automated Security Verification

The objective of any automated security verification tool is to analyze the strength of an authentication protocol for any threat. ProVerif [52] is deemed to be as one of the powerful tools by the academia to judge the reliability of authentication schemes' robustness against threats. ProVerif works on widely familiar applied π calculus which supports a great deal with different cryptographic primitives like encryption/ decryption, digital signatures, one-way hash-based and Diffie-Helman-based operations etc. In order to test the efficacy of our scheme, we have analyzed and tested the results of the protocol in ProVerif automated tool.

We begin the tool testing process, first, by defining the two channels used among the Ui, Sj and RC entities as, a private channel *SCh* and a public channel *PCh*.

```
(*** Channels ***)
free SCh: channel [private].     (*Confidential Channel*)
free PCh: channel.               (*Open/insecure Channel*)
```

The constants and variables as used in the proposed scheme are given as follows.

```
(*** Constants and Variables ***)
free IDi : bitstring.
free SIDj : bitstring.
free x : bitstring [ private ] .
free y : bitstring [ private ] .
```

The constructors H, h, XOR and CONCAT are defined as Bio-hashing, one-way hash functions, exclusive or and concatenation, respectively. We define an equation (XOR) to utilize the property of exclusive or, i.e. XOR(XOR(u,v),v) = u. The security primitives, i.e. constructors, destructors, and equations for the proposed scheme have been modeled in ProVerif as follows.

```
(*** Constructor ***)
fun h(bitstring ) :bitstring .
fun H(bitstring ) :bitstring .
fun XOR(bitstring, bitstring): bitstring.
fun ENC(bitstring, bitstring): bitstring.
fun CONCAT( bitstring ,bitstring):bitstring .


(** Destructors & related Equations **)
equation forall u: bitstring, v: bitstring; XOR (XOR(u,v) ,v)=u.
reduc forall w: bitstring, key: bitstring; DEC (ENC (w, Pub), Prs)=w.
```

We have modeled two events for each of the entities (Ui and Sj). The start and end events for Ui are beginUserUi(bitstring) and endUserUi(bitstring), while the same events for Sj are beginServerSj (bitstring) and endServerSj(bitstring). The authenticity of our protocol can be verified by checking the associated relationship between either of the participant's beginning and ending events. These events are described as follows.

```
(** Events **)
event beginUserUi ( bitstring ) .
event endUserUi ( bitstring ) .
event beginServerSj ( bitstring ) .
event endServerSj ( bitstring ) .
```

We have defined three distinct processes *UserUi*, *RegistrationCentreRC*, and *ServerSj* to model the three entities i.e. *Ui*, *RC* and *Sj*, respectively. The process *UserUi* initially sends the computed parameters (IDi, Y, TPW') using secure channel SCh towards *ServerSj*. Likewise, after receiving the (xPIDi, xDi', xFi, xW') message, the *UserUi* process further computes W, Di and B_r, and stores all parameters in smart card. In mutual authentication phase, the *UserUi* process compares xFi and Fi' after computing Fi'. If it holds true, it further computes TPW, Vi, A, ZIDi. Next, using PCh, it sends (xPIDi, ZIDi, Ni) towards *ServerSj* process. Afterwards, the same process, *UserUi* receives (xMij, xTi, xNj) and computes xMij' and compares with xMij. If both parameters are equivalent, then computes SKij, Mij' and sends Mij' towards *ServerSj* process. Next, it recovers PIDi' and replaces with PIDi in smart card.

```
(*********************** p r o c e s s e s *********************)
( **************** User Ui *************** )
let UserUi=
( ****** Registration * )
new r1: bitstring;
new r2: bitstring;
let TPW= h(CONCAT(PWi, H(BIOi)), b) in
```

```
let TPW'=XOR(TPW, r1) in
let Y=h(CONCAT(H(BIOi), IDi, r2)) in
out (SCh , (IDi, Y, TPW')) ;
in (SCh, (xPIDi : bitstring, xDi':bitstring, xFi:bitstring , xW':bitstring,)) ;
let W=XOR(xW', r1) in
let Di=XOR(xDi', h(CONCAT(IDi, r2))) in
let Br=XOR(H(BIOi), r2) in

( ******** Login and Authentication * )
event beginUserUi ( IDi ) ;
let r2= XOR(Br, H(BIOi)) in
let Y=h(CONCAT(H(BIOi), IDi, r2)) in
let Fi'=h(h(CONCAT(IDi, Y))) in
if (xFi=Fi') then
new Ni:bitstring;
let TPW= h(CONCAT(PWi, H(BIOi)), b) in
let Vi= XOR(TPW, W) in
let A=XOR(Di, h(CONCAT(IDi, r2)), h(CONCAT(IDi, Y))) in
let ZIDi = h(CONCAT(xPIDi, Vi, A, Ni)) in
out (PCh, (xPIDi, ZIDi, Ni)) ;
in (PCh, (xMij : bitstring, xTi:bitstring, xNj:bitstring)) ;
let xMij'=h(CONCAT(A, xTi, Ni, xNj, Vi, SIDj)) in
if (xMij' = xMij) then
let SKij= h(CONCAT(A, Ni, xNj, Vi, SIDj)) in
let Mij'=h(CONCAT(SKij, A, xNj, Vi, SIDj)) in
let PIDi'=XOR(xTi, h(CONCAT(xPIDi, A, Vi))) in
let PIDi = PIDi' in
out (PCh, (Mij')) ;
event endUserUi (IDi)
else
0.
```

The *RegistrationCentreRC* process receives the parameters (xIDi, xY, xTPW') from *UserUi* process on a secure channel SCh, computes A and Vi, and sends the tuple (Wi', PIDi, Di', Fi) towards *UserUi* using SCh channel. Likewise, for registering server it computes XY= h(CONCAT(x, y)) in, and sends to any new process *ServerSj* to complete the registration process.

```
(******************* Registration Centre (RC) *******************)
let RegistrationCentreRC =
( ******* User Registration * )
in (SCh , xIDi : bitstring, xY: bitstring, xTPW': bitstring ) ;
let A=h(CONCAT(IDi, x)) in
let Vi=h(CONCAT(A, XY)) in
let W'=XOR(TPW', Vi) in
new t : bitstring ;
let PIDi = ENC(CONCAT(A, h(t)), XY) in
let Di' = XOR(CONCAT(A, h(CONCAT(IDi, Y))) in
let Fi = h(h(CONCAT(IDi, Y))) in
out (SCh, (PIDi , Di', Fi, W')) ;
( ******* Server Registration * )
let XY = h(CONCAT(x, y)) in
out (SCh , ( XY )) ;
0 .
```

The *ServerSj* process receives the parameter xXY during server registration process. During mutual authentication phase, the *ServerSj* process receives the tuple (xxPIDi, xZIDi, xNi) from *UserUi* process and computes A, Vi and ZIDi for comparing ZIDi with xZIDi. If it holds true, it further generates t' and computes PIDi', Ti and Mij. Then, it sends the tuple (Mij, Ti, Nj) using public channel towards UserUi. The same process, after receiving xMij', computes SKij, Mij'' and compares xMij' against Mij''. If it holds true, then it validates the *UserUi* as a valid process and the developed session key SKij, and proceeds for verifying the message authenticity on its end.

```
(******************* Server (Sj) *******************)
( * Server Sj * )
let ServerSj=
( ********* Registration * )
in (SCh , ( xXY : bitstring )) ;

( ********* Login and Authentication * )
event beginServerSj ( SIDj ) ;
in (PCh, (xxPIDi, xZIDi, xNi)) ;
let A = DEC( xxPIDi ,xXY) in
let Vi = h(CONCAT(A, xXY)) in
let ZIDi'=h(CONCAT(xxPIDi, Vi, A, xNi)) in
if (ZIDi' = xZIDi) then
new t' : bitstring ;
new Nj : bitstring ;
let PIDi' = ENC(CONCAT(A, h(t')), xXY) in
let Ti=XOR(PID', h(CONCAT(xxPIDi, A, Vi)) in
let Mij=h(CONCAT(A, Ti, xNi, Nj, Vi, SIDj)) in
out (PCh, (Mij, Ti, Nj)) ;
in (PCh,(xMij': bitstring)) ;
let SKij = h(CONCAT(A, xNi, Nj, Vi, SIDj)) in
let Mij''=h(CONCAT(SKij, A, Nj, Vi, SIDj)) in
if (Mij'' = xMij') then
event endServerSj ( SIDj )
else
0 .
```

The three principals or participants are agreed for an unbounded number of parallel sessions, hence the three processes are deemed to be in replication as shown below.

```
process
( ( ! UserUi ) | ( ! RegistrationCentreRC) | ( ! ServerSj ) )
```

We define the under mentioned queries for testing the security and correctness of the proposed protocol.

```
(** Queries **)
free SK: bitstring [ private ] .
query attacker (SK) .
query id : bitstring ; inj event ( endUserUi ( id )) ==> inj
event ( beginUserUi ( id )) .
query id : bitstring ; inj event ( endServerSj ( id ) ) ==> inj
event ( beginServerSj ( id )) .
```

The following three results have been obtained after the implementation of above mentioned queries in this simulation.

RESULT inj-event(endServerSj(id)) ==> inj-event(beginServerSj(id)) is true.
RESULT inj-event(endUserUi(id_1890)) ==> inj-event(beginUserUi(id_1890)) is true.
RESULT not attacker (SK[]) is true.

The computed results (1) and (2) indicate clearly that all of the three processes started and ended successfully, while the result (3) verifies that the adversary's query failed to expose the session key generated by the processes during the authentication phase.

5.2 Informal security discussion

The security analysis of proposed scheme is described below:

5.2.1 Replay Attacks

An adversary Ă, having access to intercepted parameters *{PIDi, ZIDi, Ni, Mij, Ti, Nj, Mij'}* might attempt to replay the message to deceive any legal user or server. However, the use of newly created session parameters by the legal participants like *Ni* and *Nj,* each time a session is established, debars Ă to launch a replay attack. If an attacker replays the message *m1= {PIDi, ZIDi, Ni}* towards *Sj,* the later verifies the authenticity of *Ui* in the *m3* message received in the last, in response to the *Nj* based challenge. i.e. if there is *Nj* in m3 along with other parameters, it validates the user. At the same time, the *Ui* confirms the authenticity of Sj in the *m2* message, in response to the *Ni* based challenge in *m1*. i.e. if there is *Ni* in m2 along with other parameters, it validates the server. Hence, the proposed scheme could foil a replay attack successfully.

5.2.2 Modification Attacks

If an adversary attempts to modify the intercepted messages *{PIDi, ZIDi, Ni, Mij, Ti, Nj, Mij'},* it may not be able to construct the parameters *{ZIDi, Gij, Mij, Mij'}* by generating novel session variables, in view of the fact that the construction of these messages require the knowledge of *Vi* and *A.* While, these parameters are only known to the legitimate participants, and the later can easily detect any malicious participant. Therefore, Ă may not be able to launch modification attack. Hence, the proposed scheme could foil a modification attack successfully.

5.2.3 Offline-password guessing Attack

This attack can be initiated when an attacker attempts to obtain a Ui's password on account of publicly available parameters [44-46]. In proposed scheme, an adversary may intercept the messages and access the parameters *{PIDi, ZIDi, Ni, Mij, Ti, Nj, Mij'}* after careful observation of public channel. Nonetheless, the adversary may not be able to recover the *PWi,* since *PWi* is not used in any transcript that could offer the adversary any chance to guess Ui's password. Likewise, using stolen smart card contents *{PIDi, Di, Fi,* B_r *, W, h()}* , the adversary needs *H(BIOi)* and *Vi* parameters to guess *PWi* from *W.* Hence, the offline guessing attack using smart card cannot be initiated in polynomial time in proposed scheme.

5.2.4 Stolen Verifier Attacks

An attacker might get some precious information that is stored on server's end; and if it also maintains the database of *Ui*'s information like passwords or other shared secrets, and utilize it to impersonate as the legal users, this is termed as stolen verifier attack [41-43].
The proposed scheme does not keep any storage database on the part of *Sj* or *RC* that is an essential requirement for an attacker to launch such an attack.

5.2.5 Stolen Smart Card attack

As we see in sub-section 5.3, that an attacker can never extract password using stolen smart card contents in polynomial time. In view of this fact, the attacker might not be able to construct an up-to-date *ZIDi* parameter for authentication request message except replaying it, which is detected by the server in the third run. Therefore, the adversary evidently cannot initiate any sort of impersonation or masquerading attack. Hence, the stolen smart card contents do not lead to other attacks in our scheme.

5.2.6 Session Key Security

In proposed scheme, for constructing a valid session key *SK= h(A|| Ni || Nj || Vi || SIDj),* an adversary needs to access *A* and *Vi*. Even, if the *IDi* of the user is exposed, still an adversary may not construct the parameter *A=h(IDi || x)* that prevents Ă from generating *SK*, unlike Chen and Lee and Moon et al. schemes. Furthermore, Ă also needs *Vi* to construct *SK,* however, an adversary cannot derive or construct *Vi* without the knowledge of *h(x||y), TPW* or *H(BIOi)* parameters. Hence, there are little chances for an adversary to generate a valid session key by any of the means.

5.2.7 Known-Key Security

The known-key security maintains the security of private keys or secrets of involved participants in a session, once the session key is compromised [36-40]. Since, the session key *SK= h(A || Ni || Nj || Vi || SIDj)* is not a function of Ui's password *PWi*. Although, it contains the parameter *A*, but Ă cannot guess *x*: *RC's* private key, out of it as being a large bit random integer. Hence, an adversary cannot guess the secrets out of any session key revealed. Therefore, the proposed scheme has been quite secure for the known-key security.

5.2.8 Mutual Authentication

The mutual authentication ensures that the participants authenticate one another during the same protocol. In proposed scheme, the involved participants authenticate one another on the basis of factors *A* and *Vi*. Both of these parameters are not easily accessible to an adversary, which is only possible with the disclosure of *RC* and *Sj* secrets. An adversary cannot decrypt *PIDi* to access *A* by computing *(A||h(t))* = $D_{h(x||y)}$ *PIDi*, for not having *h(x||y)*. In addition, the access to *Vi* requires the knowledge of either *BIOi* and *PWi*, or *h(x || y)*, which is not possible under normal conditions.

5.2.9 Anonymous Authentication

The anonymous authentication is meant for hiding the identity of user from outsiders during mutual authentication process. In proposed scheme, Ui submits its identity in the form of *PIDi*=$E_{h(x||y)}$ *(A|| h(t))*, which is masked under the guise of *t* secret, as generated by *Sj*. The *Sj* recovers the dynamic identity parameter *A* by decrypting *PIDi* through *h(x || y)*, which is utilized in further computation. In this manner, the proposed scheme provides anonymity to user *Ui*.

5.2.10 Resists against Key-Compromise Impersonation attack (KCI)

Our proposed scheme is resistant to KCI attack in comparison with Chen et al, Moon et al. and Wang et al. protocols, since the stolen smart contents can never help adversary in extracting the other useful parameters, for instance, *Vi* and *A*. Even if the parameter *A* is approached by an adversary by some means, it will not be able to compute *Vi*, which further requires access to *h(x || y)*. Hence, the adversary cannot construct a legitimate *Mij* parameter by maliciously acquiring the parameter *A*, and in return, no key-compromise impersonation or server masquerading attack is possible against *Ui*.

5.2.11 Password modification without RC participation

In proposed scheme, the password can be easily modified by adopting the procedure as described in section 4.3, without having interaction with registration centre, unlike Chen and Lee, schemes. The Chen and Lee scheme cannot update the password without the engagement of registration centre. As in Chen and Lee the construction of *Ri* requires the use of password *PWi*, which is further used in the construction of *Vi* parameter, and in turn used in *Ei* parameter to be stored in smart card. Nonetheless, our scheme is capable of updating smart card parameter '*W'* in accordance with the newly modified password *PWi*, without the involvement of registration centre.

5.3 Security analysis using BAN logic

This section demonstrates the security proof of proposed technique using Burrows-Abadi-Needham logic (BAN) logic [44-45], which is a model that proves the protocol's robustness related to mutual authentication between participants, key distribution to those participants, and resistance to session key exposure. In this logic, we employed principals, keys and nonces as defined below.

Principals, acting as agents, participate in a protocol.
Keys are meant for encryption using symmetric crypto- primitives.
Nonces, in messages, are used to counter replay attacks.
Some notations, as used in this proof, are described as under:

$\mathfrak{A} \mid\equiv \mathfrak{Q}$: $\mathfrak{A}$ believes $\mathfrak{Q}$.
$\mathfrak{A} \lhd \mathfrak{Q}$: $\mathfrak{A}$ sees $\mathfrak{Q}$ after receiving it.
$\mathfrak{A} \mid\sim \mathfrak{Q}$: $\mathfrak{A}$ once said $\mathfrak{Q}$. i.e. In history, $\mathfrak{A}$ had transmitted $\mathfrak{Q}$ and $\mathfrak{A}$ believed that when sent.
$\mathfrak{A} \Rightarrow \mathfrak{Q}$: $\mathfrak{A}$ has jurisdiction over $\mathfrak{Q}$ and can behave as an authority over $\mathfrak{Q}$ that might be trusted.
$\sharp(\mathfrak{Q})$: The message $\mathfrak{Q}$ is produced fresh and not replayed.
$(\mathfrak{Q})_{\mathfrak{C}}$: The formulae $\mathfrak{Q}$ is used in combination with formulae $\mathfrak{C}$.
$(\mathfrak{Q}, \mathfrak{C})$: $\mathfrak{Q}$ or $\mathfrak{C}$ being the part of message $(\mathfrak{Q}, \mathfrak{C})$.
$\mathfrak{A} \xleftrightarrow{K} \mathfrak{A}'$: $\mathfrak{A}$ and $\mathfrak{A}'$ can securely contact using the shared key K.
$\langle \mathfrak{Q}, \mathfrak{C} \rangle_K$: $\mathfrak{Q}$ or $\mathfrak{C}$ is encrypted using the key K.

We state few logical postulates (rules) as used this logic analysis, in the following:

$\Re1$. Message meaning postulate: $\frac{\mathfrak{A}|\equiv\mathfrak{A}\overset{K}{\leftrightarrow}\mathfrak{A}',\ \ \mathfrak{A}\lhd\langle\mathfrak{Q}\rangle_{\mathfrak{C}}}{\mathfrak{A}|\equiv\mathfrak{A}'\ |\sim\ \mathfrak{Q}}$

$\Re2$. Nonce verification postulate: $\frac{\mathfrak{A}|\equiv\sharp(\mathfrak{Q}),\ \ \mathfrak{A}|\equiv\mathfrak{A}'\ |\sim\ \mathfrak{Q}}{\mathfrak{A}|\equiv\mathfrak{A}'\ |\equiv\ \mathfrak{Q}}$

$\Re3$. Jurisdiction postulate: $\frac{\mathfrak{A}|\equiv\mathfrak{A}'\Rightarrow\mathfrak{Q},\ \ \mathfrak{A}|\equiv\mathfrak{A}'\ |\equiv\ \mathfrak{Q}}{\mathfrak{A}|\equiv\ \mathfrak{Q}}$

$\Re4$. Freshness conjuncatenation postulate: $\frac{\mathfrak{A}|\equiv\sharp(\mathfrak{Q})}{\mathfrak{A}|\equiv\sharp(\mathfrak{Q},\ \mathfrak{C})}$

$\Re5$. Belief postulate: $\frac{\mathfrak{A}|\equiv(\mathfrak{Q}),\ \ \mathfrak{A}|\equiv(\mathfrak{C})}{\mathfrak{A}|\equiv(\mathfrak{Q},\ \mathfrak{C})}$

$\Re6$. Session keys postulate: $\frac{\mathfrak{A}|\equiv\sharp(\mathfrak{Q}),\ \ \mathfrak{A}|\equiv\mathfrak{A}'\ |\equiv\ \mathfrak{Q}}{\mathfrak{A}|\equiv\mathfrak{A}\overset{K}{\leftrightarrow}\mathfrak{A}'}$

The proposed model should meet the following goals to strengthen its security using BAN logic, given the above postulates and assumptions.

Goal1 : $Sj\ |\equiv Sj \xleftrightarrow{SK}$ Ui

Goal2 : $Sj\ |\equiv Ui\ |\equiv Sj \xleftrightarrow{SK}$ Ui

Goal3 : $Ui\ |\equiv Sj \xleftrightarrow{SK}$ Ui

Goal4 : $Ui\ |\equiv Sj\ |\equiv Sj \xleftrightarrow{SK}$ Ui

To proceed in this proof, first, the exchange messages need to be transformed into idealized form as depicted below.

M1: $Ui \rightarrow Sj$: $PIDi, ZIDi, Ni$: $\{\langle A\ ||\ h(t)\rangle_{h(x\ ||\ y)}, \langle A, PIDi, Ni\rangle_{Vi}, Ni\}$

M2: $Sj \rightarrow Ui$: Mij, Ti, Nj: $\{\langle Ti, Ni, Nj, SIDj\rangle_{A,\ Vi}, Ti, Nj\}$

M3: $Ui \rightarrow Sj$: Mij': $\{\langle SKij, Nj,\ SIDj\rangle_{A,\ Vi}\}$

Next, the following premises could be established to proceed further in logic proof.

A1 : $Ui\ \ |\equiv\ \sharp Ni$

A2 : $Sj\ \ |\equiv\ \sharp Nj$

A3 : $Ui\ \ |\equiv$ Sj $\xleftrightarrow{A,\ Vi)}$ Ui

A4 : $Sj\ \ |\equiv$ Sj$\xleftrightarrow{A,\ Vi)}$ Ui

A5 : $Ui\ \ |\equiv$ Sj $|\equiv$ Ui $\xleftrightarrow{A,\ Vi)}$ Sj

A6 : $Sj\ \ |\equiv$ Ui $|\equiv$ Ui $\xleftrightarrow{A,\ Vi)}$ Sj

A7 : $Ui\ \ |\equiv$ Sj $\Rightarrow$ Mij

A8 : $Sj\ \ |\equiv$ Ui $\Rightarrow$ Mij'

Further, the designed idealized forms (M_1, M_2 and M_3) of the proposed model could be evaluated and tested, considering the above narrated premises and postulates.
By using the above notations, rules, premises and idealizations, we get to the following derivations:

Considering M1 and M3 of the idealized form:

M1: $Ui \rightarrow Sj$: $PIDi, ZIDi, Ni$: $\{\langle A\ ||\ h(t)\rangle_{h(x\ ||\ y)}, \langle A, PIDi, Ni\rangle_{Vi}, Ni\}$

M3: $Ui \rightarrow Sj$: Mij': $\{\langle SKij, Nj,\ SIDj\rangle_{A,\ Vi}\}$

By applying seeing rule, we get

S1: $Sj \lhd PIDi, ZIDi, Ni$: $\{\langle A\ ||\ h(t)\rangle_{h(x\ ||\ y)}, \langle A, PIDi, Ni\rangle_{Vi}, Ni\}$

S2: $Sj \lhd Mij'$: $\{\langle SKij, Nj,\ SIDj\rangle_{A,\ Vi}\}$

According to S1, S2, A3 and $\Re1$, we say

S3: $Sj\ |\equiv\ Ui \sim PIDi, ZIDi, Ni$: $\{\langle A\ ||\ h(t)\rangle_{h(x\ ||\ y)}, \langle A, PIDi, Ni\rangle_{Vi}, Ni\}$

S4: $Sj\ |\equiv\ Ui \sim \{\langle SKij, Nj,\ SIDj\rangle_{A,\ Vi}\}$

According to S3, S4, A1, $\Re4$ and $\Re2$, we say

S5: $Sj\ |\equiv\ Ui\ |\equiv\ \{\langle A\ ||\ h(t)\rangle_{h(x\ ||\ y)}, \langle A, PIDi, Ni\rangle_{Vi}, Ni\}$

S6: $Sj\ |\equiv\ Ui\ |\equiv\ \{\langle SKij, Nj,\ SIDj\rangle_{A,\ Vi}\}$

According to S5, S6, A4, A8 and $\Re3$, we get

S7: $Sj\ |\equiv\ \{\langle A\ ||\ h(t)\rangle_{h(x\ ||\ y)}, \langle A, PIDi, Ni\rangle_{Vi}, Ni\}$

S8: $Sj\ |\equiv\ \{\langle SKij, Nj,\ SIDj\rangle_{A,\ Vi}\}$

Using S7, S8, A4, ($SK = h(A\ ||\ Ni\ ||\ Nj\ ||\ Vi\ ||\ SIDj)$) and $\Re6$, we get

S9: $Sj\ |\equiv\ Sj \xleftrightarrow{SK} Ui$ **(Goal 1)**

According to S9, A6 we apply $\Re6$ as

S10: $Sj\ |\equiv\ Ui\ |\equiv\ Sj \xleftrightarrow{SK} Ui$ **(Goal 2)**

Further, we consider M2 in idealized form:

M2: $Sj \rightarrow Ui$: Mij, Ti, Nj: $\{\langle Ti, Ni, Nj, SIDj\rangle_{A,\ Vi}, Ti, Nj\}$
By applying seeing rule, we get

S11: $Ui \lhd Mij'$: $\{\langle Ti, Ni, Nj, SIDj\rangle_{A,\,Vi}, Ti, Nj\}$

According to S11, A4 and $\Re 1$, we can say

S12: $Ui \mid\equiv Sj \sim \{\langle Ti, Ni, Nj, SIDj\rangle_{A,\,Vi}, Ti, Nj\}$

According to S12, A2, $\Re 4$ and $\Re 2$, we say

S13: $Ui \mid\equiv Sj \mid\equiv \{\langle Ti, Ni, Nj, SIDj\rangle_{A,\,Vi}, Ti, Nj\}$

According to S13, A3, A7 and $\Re 3$, we get

S14: $Ui \mid\equiv \{\langle Ti, Ni, Nj, SIDj\rangle_{A,\,Vi}, Ti, Nj\}$

From S14, A3, ($SK = h(A \,||\, Ni \,||\, Nj \,||\, Vi \,||\, SIDj)$), and $\Re 6$, we get

S15: $Ui \mid\equiv Sj \xleftrightarrow{SK} Ui$ **(Goal 3)**

According to S15, A5, we apply $\Re 6$ as

S16: $Ui \mid\equiv Sj \mid\equiv Sj \xleftrightarrow{SK} Ui$ **(Goal 4)**

Based on the above logical analysis, we could infer that the proposed model adheres to mutual authentication property that leads to the establishment of a mutually shared session key *SK* between Ui and Sj.

6. PERFORMANCE EVALUATION

In this section, we evaluate the performance of proposed model with other multi-server authentication protocols, in terms of resistance against threats. The Table II depicts the analysis of security features for various protocols including Chen and Lee [24], Wang et al. [26], Moon et al. [25], which indicates our proposed scheme as a robust authentication protocol against those contemporary schemes. According to Table II, all of these three schemes [24-26] are found vulnerable to impersonation attack, KCI and trace attack. Besides, the Moon et al. does not provide anonymity and resistance to identity guessing and stolen smart card attacks. The Wang et al. fails to provide resistance to privileged insider attack and session-specific temporary information attack. Likewise, Chen and Lee could not provide session key security and resistance to stolen smart card attack and offline-password guessing attacks.

Table II: Comparison of security-based features

	Chen and Lee [24]	**Wang et al. [26]**	**Moon et al. [25]**	**Ours**
Anonymity	Yes	Yes	No	Yes
Mutual Authentication	Yes	No	No	Yes
Resist privileged insider Attack	Yes	No	Yes	Yes
Resist Offline password guessing attack	No	Yes	Yes	Yes
Resist Stolen smart card attack	No	Yes	No	Yes
Resists Impersonation attack	No	No	No	Yes
Resists Key-compromise impersonation attack	No	No	No	Yes
Session key security	No	Yes	No	Yes
Resist Trace attack	No	No	No	Yes
Resist session-specific temporary information attack	Yes	No	Yes	Yes
Resists Identity guessing attack	Yes	Yes	No	Yes
Efficient Password Modification	No	Yes	Yes	Yes

Table III. Operations cost comparison

		Chen and Lee	**Wang et al.**	**Moon et al.**	**Ours**
Login & Authentication phase	Server side	$8T_h$	$6T_h$	$7T_h$	$10T_h$
	User side	$11T_h$	$8T_h$	$9T_h$	$11T_h$
Total		$19T_h$	$14\ T_h$	$16T_h$	$21T_h$
Computation cost (ms)		*0.043*	*0.032*	*0.036*	*0.048*
Energy (μJ)		*14.44*	*10.64*	*12.16*	*15.96*

For comparing the costs, in Table III, we represent hash operation with T_h while overlooking XOR function due to its insignificant cost. Hence, considering the given performance analysis, we can infer that our proposed technique is more secure than Wang et al., Moon et al., and Chen and Lee, schemes. All of these protocols are based on light-weight SHA-1 hash-digest operations. The proposed scheme sustains a bit higher cost than Wang et al. and Moon et al. et al. schemes, and lower cost than Chen and Lee, however the proposed scheme provides more security than those schemes. In fact, all of these schemes can be regarded as light-weight, since hash-digest is regarded as a negligible operation in higher cost crypto-primitives, i.e. scalar point multiplication, exponentiation and bilinear operations. Therefore, all of these schemes can be regarded as equivalent in terms of computational cost. However, the immunity of our scheme against most of the identified threats turns the scale in its favor as shown in Table II.

According to Klinic [33], the hash operation assumes to take *0.0023ms* time delay. Considering this, the cost of Chen and Lee, Wang et al., Moon et al., and proposed scheme amounts to *0.043ms*, *0.032ms*, *0.036* and *0.048ms*, respectively. Likewise, the schemes may be evaluated on the basis of energy requirements by taking the cost of SHA-1 as 0.76μJ for the computation of a single byte [54]. In this regard, the energy cost for the Chen and Lee, Wang et al., Moon et al., and proposed schemes will amount to $14.44\ \mu J$, $10.64\ \mu J$, $12.16\ \mu J$ and $15.96\mu J$, respectively. Hence, considering the above performance evaluation, we can deduce that the proposed protocol is more secure than all schemes as analyzed, in almost an equivalent cost.

7. CONCLUSION

This study reviews three multi-server authentication schemes, Chen and Lee, Wang et al., and Moon et al. aimed at maximizing the security in minimum cost. The Chen and Lee scheme was found susceptible to impersonation attack, trace attack, stolen smart card attack exposing session keys, key-compromise impersonation attack and inefficient password modification. The Wang et al. scheme does not provide resistance to trace attack, session-specific temporary information attack, key-compromise information attack, and privileged insider attack leading to session key disclosure and user impersonation attacks. The Moon et al. is prone to stolen smart card attack leading to further attacks, i.e. identity-guessing attack, user impersonation attack, key-compromise impersonation attack, and session keys disclosure. The proposed scheme presented its contribution with an improved version countering the identified threats. Besides, the proposed work incorporates logic-based security analysis and the performance evaluation with contemporary schemes.

REFERENCES

[1] Lamport, L. (1981). Password authentication with insecure communication. Communications of the ACM, 24(11), 770–772.
[2] Chang, C. C., & Wu, T. C. (1991). Remote password authentication with smart cards. IEE Proceedings E: Computers and Digital Techniques, 138, 165–168.
[3] Li LH, Lin IC, Hwang MS. A remote password authentication scheme for multiserver architecture using neural networks. IEEE Transactions on Neural Networks 2001; 12(6):1498–1504.
[4] Lin IC, Hwang MS, Li LH. A new remote user authentication scheme for multi-server architecture. Future Generation Computer Systems 2003; 1(19):13–22.
[5] Juang WS. Efficient multi-server password authenticated key agreement using smart cards. IEEE Transactions on Consumer Electronics 2004; 50(1):251–255.
[6] Tsaur WJ, Wu CC, Lee WB. An enhanced user authentication scheme for multi-server internet services. Applied Mathematics and Computation 2005; 170:258–266.
[7] Chang CC, Lee JS. An efficient and secure multiserver password authentication scheme using smart cards. IEEE Proceeding of the International Conference on Cyberworlds, Tokyo, Japan, Nov. 2004; 417–422.
[8] Liao YP, Wang SS. A secure dynamic ID based remote user authentication scheme for multi-server environment. Computer Standards & Interfaces 2009; 31(1):24–29.
[9] Hsiang HC, Shih WK. Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment. Computer Standards &Interfaces 2009; 31(6):1118–1123.
[10] Lee CC, Lin TH, Chang RX. A secure dynamic ID based remote user authentication scheme for multiserver environment using smart cards. Expert Systems with Applications 2011; 38(11):13863–13870.
[11] He, D., & Wang, D. (2015). Robust biometrics-based authentication scheme for multiserver environment. IEEE Systems Journal, 9(3), 816-823.
[12] Li, X., Niu, J., Kumari, S., Islam, S. H., Wu, F., Khan, M. K., & Das, A. K. (2016). A novel chaotic maps-based

user authentication and key agreement protocol for multi-server environments with provable security. Wireless Personal Communications, 89(2), 569-597.
[13] He, D., Zeadally, S., Kumar, N., & Wu, W. (2016). Efficient and anonymous mobile user authentication protocol using self-certified public key cryptography for multi-server architectures. IEEE Transactions on Information Forensics and Security, 11(9), 2052-2064.
[14] Li, X., Ma, J., Wang, W., Xiong, Y., & Zhang, J. (2013). A novel smart card and dynamic ID based remote user authentication scheme for multi-server environments. Mathematical and Computer Modelling, 58(1), 85-95.
[15] Irshad, A., Sher, M., Chaudhary, S. A., Naqvi, H., & Farash, M. S. (2016). An efficient and anonymous multi-server authenticated key agreement based on chaotic map without engaging Registration Centre. The Journal of Supercomputing, 72(4), 1623-1644.
[16] Li, X., Xiong, Y., Ma, J., & Wang, W. (2012). An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards. Journal of Network and Computer Applications, 35(2), 763-769.
[17] Irshad, A., Sher, M., Ahmad, H. F., Alzahrani, B. A., Chaudhry, S. A., & Kumar, R. (2016). An improved Multi-server Authentication Scheme for Distributed Mobile Cloud Computing Services. KSII Transactions on Internet and Information Systems (TIIS), 10(12), 5529-5552.
[18] Chaudhry, S. A., Naqvi, H., Farash, M. S., Shon, T., & Sher, M. (2015). An improved and robust biometrics-based three factor authentication scheme for multiserver environments. The Journal of Supercomputing, 1-17.
[19] Irshad, A., Sher, M., Nawaz, O., Chaudhry, S. A., Khan, I., & Kumari, S. (2016). A secure and provable multi-server authenticated key agreement for TMIS based on Amin et al. scheme. Multimedia Tools and Applications, 1-27
[20] Jiang, P., Wen, Q., Li, W., Jin, Z., & Zhang, H. (2015). An anonymous and efficient remote biometrics user authentication scheme in a multi server environment. Frontiers of Computer Science, 9(1), 142-156.
[21] Irshad, A., Sher, M., Chaudhry, S. A., Xie, Q., Kumari, S., & Wu, F. (2017). An improved and secure chaotic map based authenticated key agreement in multi-server architecture. Multimedia Tools and Applications, 1-38.
[22] Chaudhry, S. A. (2016). A secure biometric based multi-server authentication scheme for social multimedia networks. Multimedia Tools and Applications, 75(20), 12705-12725.
[23] Irshad, A., Ahmad, H. F., Alzahrani, B. A., Sher, M., & Chaudhry, S. A. (2016). An efficient and anonymous Chaotic Map based authenticated key agreement for multi-server architecture. KSII Transactions on Internet and Information Systems (TIIS), 10(12), 5572-5595.
[24] Chen, Chi-Tung, and Cheng-Chi Lee. A two-factor authentication scheme with anonymity for multi-server environments. Security and Communication Networks 8.8, 2015: 1608-1625.
[25] Moon, J., Choi, Y., Jung, J., & Won, D. (2015). An Improvement of Robust Biometrics-Based Authentication and Key Agreement Scheme for Multi-Server Environments Using Smart Cards. PloS one, 10(12), e0145263.
[26] Wang, C., Zhang, X., & Zheng, Z. (2016). Cryptanalysis and Improvement of a Biometric-Based Multi-Server Authentication and Key Agreement Scheme. Plos one, 11(2), e0149173.
[27] Chang CC, Lee CY. A smart card-based authentication scheme using user identify cryptography. International Journal of Network Security 2013; 15(2):139–147.
[28] Tsai JL. Efficient multi-server authentication scheme based on one-way hash function without verification table. Computers & Security 2008; 27(3-4):115–121.
[29] Arshad, H., & Nikooghadam, M. (2015). Security analysis and improvement of two authentication and key agreement schemes for session initiation protocol. The Journal of Supercomputing, 71(8), 3163-3180.
[30] Nikooghadam, M., Jahantigh, R., & Arshad, H. (2016). A lightweight authentication and key agreement protocol preserving user anonymity. Mult. Tools and Applications, 1-23.
[31] Arshad, H., & Nikooghadam, M. (2014). Three-factor anonymous authentication and key agreement scheme for telecare medicine information systems. Journal of medical systems, 38(12), 136.
[32] Mir, O., & Nikooghadam, M. (2015). A secure biometrics based authentication with key agreement scheme in telemedicine networks for e-health services. Wireless Personal Communications, 83(4), 2439-2461.
[33] Kumari, S., Chaudhry, S. A., Wu, F., Li, X., Farash, M. S., & Khan, M. K. (2017). An improved smart card based authentication scheme for session initiation protocol. Peer-to-Peer Networking and Applications, 10(1), 92-105.
[34] Farash, M. S., Kumari, S., & Bakhtiari, M. (2016). Cryptanalysis and improvement of a robust smart card secured authentication scheme on SIP using elliptic curve cryptography. Multimedia Tools and Applications, 75(8), 4485-4504.
[35] Kumari, S., Wu, F., Li, X., Farash, M. S., Jiang, Q., Khan, M. K., & Das, A. K. (2016). Single round-trip SIP authentication scheme with provable security for Voice over Internet Protocol using smart card. Multimedia Tools and Applications, 75(24), 17215-17245.
[36] Kilinc, H. H., & Yanik, T. (2014). A survey of SIP authentication and key agreement schemes. Communications Surveys & Tutorials, IEEE, 16(2), 1005-1023.
[37] Li, X., Niu, J., Khan, M. K., & Liao, J. (2013). An enhanced smart card based remote user password

authentication scheme. Journal of Network and Computer Applications, 36(5), 1365-1371.
[38] Wu, F., Xu, L., Kumari, S., & Li, X. (2015). A novel and provably secure biometrics-based three-factor remote authentication scheme for mobile client–server networks. Computers & Electrical Engineering, 45, 274-285.
[39] Kumari, S., Li, X., Wu, F., Das, A. K., Arshad, H., & Khan, M. K. (2016). A user friendly mutual authentication and key agreement scheme for wireless sensor networks using chaotic maps. Future Generation Computer Systems, 63, 56-75.
[40] Chaudhry, S. A., Farash, M. S., Naqvi, H., Kumari, S., & Khan, M. K. (2015). An enhanced privacy preserving remote user authentication scheme with provable security. Security and Communication Networks, 8(18), 3782-3795.
[41] Kumari, S., Khan, M. K., & Li, X. (2014). An improved remote user authentication scheme with key agreement. Computers & Electrical Engineering, 40(6), 1997-2012.
[42] Qi Jiang, Fushan Wei, Shuai Fu, Jianfeng Ma, Guangsong Li, Abdulhameed Alelaiwi. Robust extended chaotic maps-based three-factor authentication scheme preserving biometric template privacy. Nonlinear Dynamics, 2016, 83(4), 2085-2101.
[43] Kumari, S., & Khan, M. K. (2014). Cryptanalysis and improvement of 'a robust smart-card-based remote user password authentication scheme'. International Journal of Communication Systems, 27(12), 3939-3955.
[44] Qi Jiang, Jianfeng Ma, Xiang Lu, Youliang Tian. An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks. Peer-to-Peer Networking and Applications, 2015, 8 (6): 1070-1081.
[45] Kumari, S., Khan, M. K., & Atiquzzaman, M. (2015). User authentication schemes for wireless sensor networks: A review. Ad Hoc Networks, 27, 159-194.
[46] Qi Jiang, Jianfeng Ma, Guangsong Li, Xinghua Li. Improvement of robust smart-card-based password authentication scheme. International Journal of Communication Systems, 2015, 28(2):383-393.
[47] M. Burrows, Abadi, M., & Needham, R. (1990). A logic of authentication. ACM Transactions on Computer Systems, 8(1), 18–36. doi:10.1145/77648.77649.
[48] M. Burrows, 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.
[49] Lu YR, Li LX, Yang X, Yang YX. (2015) Robust biometrics based authentication and key agreement scheme for multi-server environments using smart cards. PLoS One. 10(5):e0126323. doi: 10.1371/journal.pone.0126323 PMID: 25978373
[50] Mishra D, Das AK, Mukhopadhyay S. A secure user anonymity-preserving biometric-based multiserver authenticated key agreement scheme using smart cards. Expert Systems with Applications. 2014; 41(18): 8129–8143. doi: 10.1016/j.eswa.2014.07.004
[51] Qi Jiang, Jianfeng Ma, Fushan Wei. On the Security of a Privacy-Aware Authentication Scheme for Distributed Mobile Cloud Computing Services. IEEE Systems Journal, 2016. DOI: 10.1109/JSYST.2016.2574719
[52] 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.
[53] Lumini, A., and Loris N. An improved Bio-hashing for human authentication. Pattern recognition 40.3 (2007): 1057-1065.
[54] Kocher, P., Jaffe, J., and Jun, B., Differential power analysis. In: Advances in Cryptology CRYPTO 99, Lecture Notes in Computer Science, vol. 1666, pp. 388–397, 1999.
[55] Blanchet B, Cheval V, Allamigeon X, Smyth B. ProVerif: Cryptographic protocol verifier in the formal model. (Available at: http://prosecco.gforge.inria.fr/personal/bblanche/proverif/).
[56] Li, C. T., Lee, C. C., Weng, C. Y., & Fan, C. I. (2015). A Secure Dynamic Identity Based Authentication Protocol with Smart Cards for Multi-Server Architecture. J. Inf. Sci. Eng., 31(6), 1975-1992.
[57] Chen, T. Y., Lee, C. C., Hwang, M. S., & Jan, J. K. (2013). Towards secure and efficient user authentication scheme using smart card for multi-server environments. The Journal of Supercomputing, 66(2), 1008-1032.
[58] Li, C. T., Lee, C. C., Weng, C. Y., & Fan, C. I. (2013). An Extended Multi-Server-Based User Authentication and Key Agreement Scheme with User Anonymity. KSII Transactions on Internet & Information Systems, 7(1).
[59] Lee, C. C., Lou, D. C., Li, C. T., & Hsu, C. W. (2014). An extended chaotic-maps-based protocol with key agreement for multiserver environments. Nonlinear Dynamics, 76(1), 853-866.