Cryptanalysis and improvement of a Multi-server Authentication protocol by Lu et al.

Azeem Irshad[1], Muhammad Sher[1], Bander A. Alzahrani[2], Aiiad Albeshri[2], Shehzad Ashraf Chaudhry[1], Saru Kumari[3]

[1] Department of Computer Science & Software Engineering, International Islamic University, Islamabad
[e-mail: irshadazeem2@gmail.com, m.sher@iiu.edu.pk, shahzad@iiu.edu.pk]
[2]Faculty of Computing & Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia
[e-mail: baalzahrani@kau.edu.sa, aalbeshri@kau.edu.sa]
[3]Chaudhary Charan Singh University, Meerut 250004 Uttar Pradesh, India
[email : saryusiirohi@gmail.com]
*Corresponding author: Azeem Irshad

Received November 21, 2016; revised April 12, 2017; accepted November 2, 2017; published January 31, 2018

Abstract

The increasing number of subscribers and demand of multiplicity of services has turned Multi-Server Authentication (MSA) into an integral part of remote authentication paradigm. MSA not only offers an efficient mode to register the users by engaging a trusted third party (Registration Centre), but also a cost-effective architecture for service procurement, onwards. Recently, Lu et al.'s scheme demonstrated that Mishra et al.'s scheme is unguarded to perfect forward secrecy compromise, server masquerading, and forgery attacks, and presented a better scheme. However, we discovered that Lu et al.'s scheme is still susceptible to malicious insider attack and non-compliant to perfect forward secrecy. This study presents a critical review on Lu et al.'s scheme and then proposes a secure multi-server authentication scheme. The security properties of contributed work are validated with automated Proverif tool and proved under formal security analysis.

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

1. Introduction

Multi-server authentication is synonymous to overhead efficiency as it minimizes the cost whenever a client needs to access the services of multiple servers (wired and wireless) in a network. Earlier, the wired and wireless subscribers had to remember numerous passwords to procure the services of several service providers. Likewise, each service provider had to register the users and store their verifiers independently in its local repository; that procedure was inefficient for both participants. The concept is almost based upon single-sign-on where a single authentication relieves the user of multiple registrations from various servers, but with few differences. As in multi-server paradigm, the user requires to get authenticated with a server or servers each time it wants to take service out of it, using the same password and factors [1]. The authentication based on remote communication frequently involves the form of multi-server authentications that further dictates the efficiency and robustness of these techniques.

In the last decade, several multi-server authentication techniques can be seen in the literature. However, there is need to bridge more gaps in the designing of multi-server protocols. Initially, Lamport [2] gave an idea for remote authentication over an insecure network. However, the necessary condition of maintenance of a stored verifiers' database on the server's end was taken as a serious flaw due to malicious tendencies of an attacker to misuse it. Afterwards, many related authentication schemes were presented [3-5], that were exposed to many known dictionary attacks. Thereafter, numerous biometric authentication schemes are presented [6-8]. Majority of those schemes were proposed for single server environment, that puts a restriction on the number of services, a network provides. Afterwards, different smart card schemes were proposed [9-11], based on random numbers and cost efficient hash-function based schemes. Meanwhile Tsai [12] presented a multiserver authentication model. Later, Li et al. [10] found few drawbacks in [12], and came with an enhanced dynamic ID-oriented multi-server authentication model. Xue et al. [13] exposed the flaws in Li et al. and presented a new scheme. Onwards, Lu et al. found three attacks in Xue et al. i.e., off-line guessing attack, masquerading attack, and a malicious insider attack, and came with another new scheme. On the biometric side, Yang et al. and Yoon et al. [13, 17] introduced multi-server protocols. Subsequently, He [18] proved the vulnerability of both schemes for stolen card threat; insider and impersonation attack. He [18] also presented an enhanced version of protocols. Chuang et al. [20], in return proposed another smart card based biometric multiserver protocol. The scheme was confronted by Mishra et al. [21] along with the introduction of three possible attacks of DOS attack, misrepresentation and stolen card attack. Mishra et al., then proposed an anonymous authentication model with improved security features. Afterwards, He et al. [24] presented another multi-server authentication scheme. Chuang et al [31], again found forgery, masquerading, and perfect forward secrecy compromise attack in He et al, and proposed an incrementally improved robust scheme. However, the scheme is found susceptible to two threats again, i.e., lack of perfect forward secrecy, and malicious insider attack in Lu et al. [32]. The current research is based on reviewing Lu et al.'s work. Afterwards, we will present an improved scheme countering the identified threats. The security will be analyzed and performance evaluated, finally.

The section 2 defines the preliminaries including hash function, fuzzy extractor and elliptic curve essentials. Section 3 illustrates the working and review for Lu et al.'s model. The section 4 shows the proposed scheme. Section 5 demonstrates the automated security verification. While, section 6 explains the formal security analysis and section 7 demonstrates performance evaluation. The last one presents the conclusive summary.

2. Preliminaries

We describe a few preliminaries to assist the layman readers, such as, hash function, elliptic curve cryptography (ECC) and fuzzy extractor, in the following:

2.1 Hash Function

A one-sided hash function $h: \{0, 1\}^* \rightarrow Z_q^*$, should bear the following four attributes:

1. The one-sided hash function h produces a message digest of pre-determined length after getting a random string as input.
2. Given $h(\rho)=\varphi$, it is hard to take inverse $h^{-1}(\varphi)$ and recover ρ;
3. Given ρ, it is not viable to evaluate ρ', such that $\rho' \neq \rho$, and $h(\rho') = h(\rho)$;
4. Furthermore, it is computationally not viable to locate a pair ρ, ρ' given $\rho' \neq \rho$, and $h(\rho') = h(\rho)$.

2.2 Elliptic Curve essentials

The curve E_c could be defined as an array of multiple points over a prime field (F_q), on a singular elliptic curve [36]:

$$t^2 \bmod q = (s^3 + ls + f) \bmod q \quad (1)$$

where $l, f, s, t \in F_q$ and $(4l^3 + 27f^2) \bmod q \neq 0$. Assuming an elliptic curve point $\mathcal{E}(s, t)$ w.r.t (1), where as, $\mathcal{F}(s, -t)$ is negative of the point $\mathcal{E}$. Here, we take two different points, i.e. $\mathcal{E}$ (s_1, t_1) and $\mathcal{F}$ (s_2, t_2) on (1), where as the line ln acting as tangent of (1) (subject to $\mathcal{E}$ equals $\mathcal{F}$), embrace $\mathcal{E}$ and $\mathcal{F}$ crossing the curve (1) from $-\mathcal{G}(s_3, -t_3)$ and its reflection in relation to x-axis is $\mathcal{G}$ (s_3, t_3), that is, $\mathcal{E} + \mathcal{F} = \mathcal{G}$. The array of points constituting E_c/F_q, together with *point at infinity* (O), form an additive-elliptic curve cyclic group $G_q = \{(s, t) : s, t \in F_q$ and $(s, t) \in E_c/F_q\} \cup \{O\}$. Moreover, an ECC-point multiplication on G_q can be represented as $\zeta . \mathcal{E} = \mathcal{E} + \mathcal{E} + \ldots + \mathcal{E}$ (ζ times), given the point $\mathcal{E} \in G_q$ with order ϖ, while ϖ being a positive integer and $\varpi . \mathcal{E} = O$.

2.3 Fuzzy extractor

Fuzzy extractor converts the captured data (biometric stream) into randomized homogenous strings, termed as a biometric key [33-34]. These keys assist in proving the authenticity of any source generating the message. This fuzzy extractor enables the construction of a stadardized pattern of random string β_i, with the noisy biometric parameter *BIOi* along with a helper string γ_i. The working of fuzzy extractor is based on two key operations, i.e. *Gen* and *Rep*. The operation *Gen*, a probabilistic generation fucntion, produces two fixed size binary strings, one is $\beta_i \in \{0, 1\}^l$ while another is helper string $\gamma_i \in \{0, 1\}^*$. The string β_i is kept secret, where as γ_i is exposed to public. To recover β_i, the deterministic reproduction operation *Rep* is utilized with input arguments containing biometric parameter *BIOi** and helper string γ_i. For more description on fuzzy extractors, some further references [33-34] could be explored.

3. REVIEW OF LU ET AL. SCHEME

The protocol design of Lu et al.'s scheme is illustrated below:

3.1 Working of Lu et al.'s protocol

The working of Lu et al.'s scheme [32] includes registration, login & authentication sub-sections, as shown in **Fig. 1**. To describe the working of Lu et al.'s scheme, few notations are listed in **Table 1**.

Table 1. Notations description

Notations	Description
Ui, Sj, RC:	User (i^{th}) , Server (j^{th}), Registration Centre
IDi, PWi:	Identity and password of Ui
Gen():	Generate fuzzy extractor
Rep():	Reproduce fuzzy extractor
BIOi:	Biometric imprint
x:	Ui's private key
Pub/Prs:	Public and private key of Sj
PSK:	Shared key between RC and Sj
T_1-T_4:	Timestamps
n_1, n_2:	Temporary session variables
h(.):	a secure hash digest function
⊕, ||	XOR, Concatenation

3.1.1 Initialization Phase

In this phase, the proposed model dedicates a trusted RC for registration purpose, while reserves *n* number of trusted servers Sj to furnish services to the users. All servers perform registraion through RC with sharing a secret *PSK* employing a confidential channel.

3.1.2 Registration Phase

In registration phase, the user performs registration with RC so that it may qualify for services offered through various servers. The user registraion phase incorporate the following steps:

1. The Ui sends *IDi, h(PWi|| Ni)* by computing and assuming a random number Ni to RC, using a confidential channel [43]. The RC receives *{IDi, h(PWi|| Ni)}*, computes *Ri=h(IDi || h(PWi || Ni))* and sends *{Ri, h(PSK)}* to Ui by storing in smart card.
2. The Ui now computes *Xi= h(PSK)⊕ x*, and *Bi=Ni⊕ H(BIOi)*, and stores *{Xi, Bi}* also in smart card (SC). Now the SC contains the *{Ri, Xi, Bi, h()}*, finally.

3.1.3 Login and Authentication Phase

1. In login phase, the user uses its smart card for authenticated access to the services offered by servers. To serve the purpose, Ui inputs its *IDi*, *PWi* and *BIOi* and computes *Ni=Bi⊕ H(BIOi)* and checks the equality *Ri?=h(IDi || PWi || Ni)*. On successful verification, the

SC allows the Ui to proceed for next procedure. Now the Ui generates n_1, $M_1=E_{pub}\{IDi, n_1, h(PWi||Ni)\}$, and $M_2=h((Xi\oplus x)||n_1||h(PWi||Ni))$. Next, Ui sends $\{M_1, M_2\}$ towards Sj.

2. In the authentication phase the Sj receives parameters and computes $\{IDi, n_1, h(PWi || Ni)\}$ by decrypting M_1 i.e., $D_{Prs}\{M_1\}$. Then, Sj computes $h(h(PSK)||n_1||h(PWi||Ni))$ and checks the equality $M_2 ?= h(h(PSK) || n_1 || h(PWi||Ni))$. On successful authentication it generates n_2, then computes $M_3=n_2\oplus h(n_1||IDi||h(PWi||Ni))$, $skji =h(n_1||n_2||h(PWi||Ni))$, and $M_4 =h(IDi|| n_1 || skji ||h(PWi||Ni))$. Now Sj sends the message $\{M_3, M_4\}$ to Ui.
3. Ui receives the message and calculates $n_2= M_3\oplus h(n_1||IDi||h(PWi||Ni))$, $skij =h(n_1||n_2||h(PWi||Ni))$, and checks the equality $M_4 ?=h(IDi || n_1 || skij ||h(PWi||Ni))$. If validated, then Ui computes $M_5=h(skij || IDi ||n_2 || h(PWi||Ni))$, and sends the message $\{M_5\}$ to Sj.
4. Sj receives $\{M_5\}$, and verifies the equality after computing M_5, i.e., $M_5 ?=h(skji || IDi ||n_2 ||h(PWi||Ni))$. On successful verification, it establishes the session key $skij =skji =h(n_1||n_2||h(PWi||Ni))$ with Ui, finally.

3.2 Lu et al. scheme's cryptanalysis.

The Lu et al. scheme is found to be prone for two attacks, that is, perfect forward secrecy incompliance and malicious insider (impersonation) attack.

3.2.1 Non-compliance to perfect forward secrecy

The forward secrecy non-compliance attack may be initiated by an attacker, if the private keys of legitimate participants are revealed. This may lead to the computation of all previous session keys for a particular user Ui. If the private key 'P_{rs}' gets leaked accidentally, the adversary may recover all previous session keys *SK* from intercepted messages by following the undermentioned steps.

1. After approaching the factor 'P_{rs}', Ⱥ could easily decrypt M_1 and recover $IDi, n_1, h(PWi || Ni)$.
2. Next, the adversary can easily get n_2 from M_3 by performing $n_2= M_3\oplus h(n_1||IDi||h(PWi||Ni))$. Since, the Session key is based on n_1, n_2 and $h(PWi||Ni)$ i.e., $skij =skji =h(n_1||n_2||h(PWi||Ni))$, following this, the previous session keys can easily be constructed by using the extracted values from the stored open message parameters {M_1 and M_3}.

3.2.2 Malicious insider attack

The Lu et al. scheme fails to differentiate among legitimate users, already registered on valid basis, during authentication phase. A malicious legal user (insider), having the knowledge of shared key between RC and Ui, i.e., $h(PSK)$, may launch an attack easily without even stealing the smart card details. Even, a malicious insider without intercepting the message parameters $\{M_1, M_2, M_3, M_4,$ and $M_5\}$ may launch this attack. In Lu et al.'s protocol, Sj has no mechanism for verifying the authenticity of $h(PWi||Ni)$, or binding the identity IDi with $h(PWi||Ni)$, given that it does not maintain any password verifier database. Neither, RC makes a use of *PSK* as a function in its computations during registration phase that renders the server Sj devoid of performing any positive verification. Hence, any malicious insider having the knowledge of $h(PSK)$ may launch an insider attack and easily compute M_1 and M_2 by computing $M_1=E_{pub}\{IDi^*, n_1^*, h(PWi^* ||Ni^*)\}$ and $M_2=h(h(PSK) || n_1^*||h(PWi^* ||Ni^*))$ after generating

random values *(IDi*, n_1*, PWi** and *Ni*)*. The Sj authenticates Ui, only on the basis of *h(PSK)* owned by all users in the system, which is a serious flaw in the scheme.

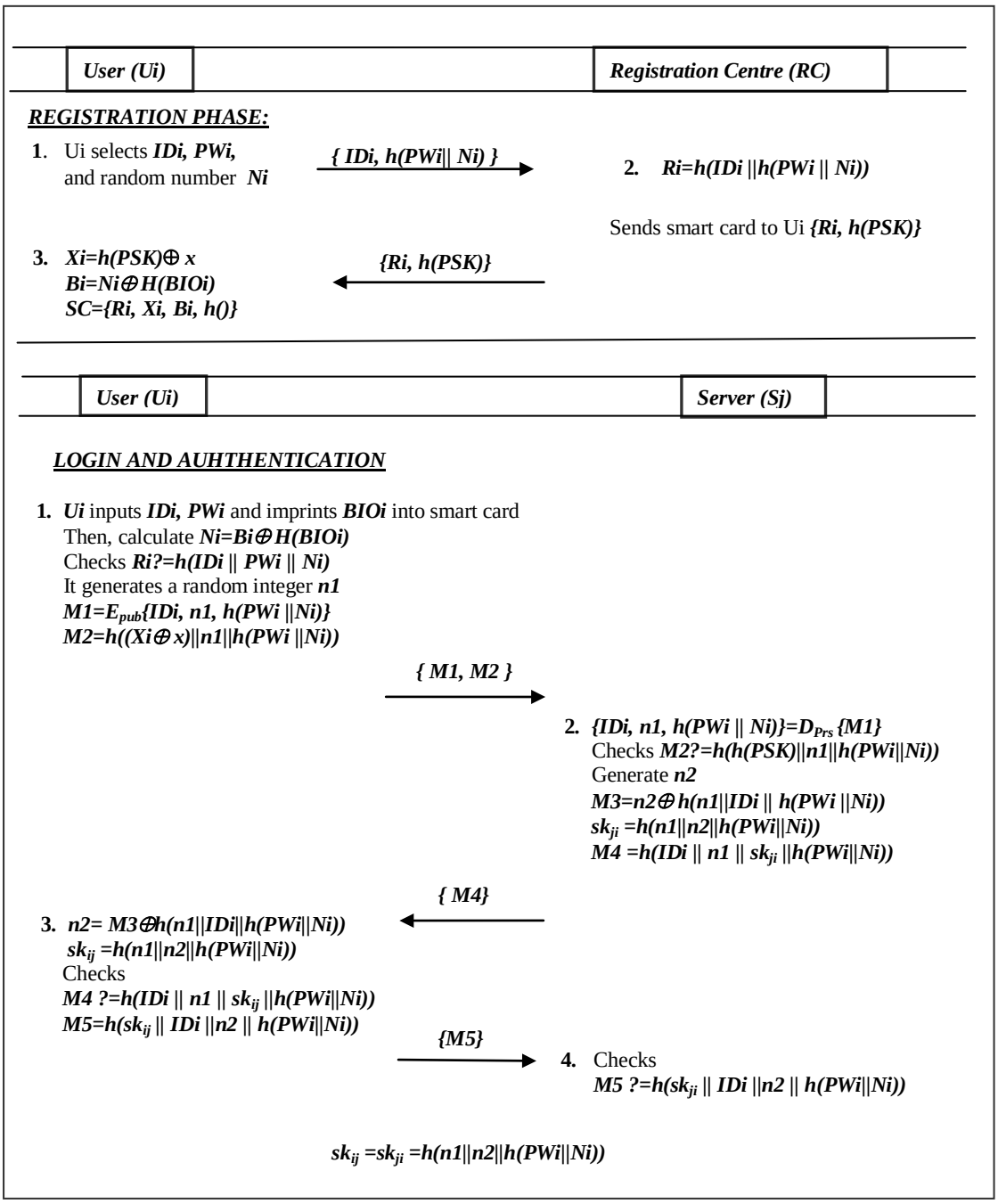

Fig. 1. Registration, login & authentication of Lu et al.'s protocol

4. PROPOSED MODEL

The discovered impersonation and forward secrecy violation attacks in Lu et al.'s scheme was the motivation for presenting an improved model. The proposed work is a smart-card based scheme that takes the user's fingerprint as biometric input into the SC to proceed with the login and authentication phase. The proposed model architecture comprises many subscibers (users), a trustworthy registration centre, and various service providers. The server Sj performs registration with RC through sharing a high entropy secret key *PSK*, over a confidential channel, which precedes the user registration procedure. The proposed model comprises three phases, i.e., 1) user registration phase 2) login & authentication phase (mutual authentication), and 3) password modification phase.

4.1 User Registration Phase

In this phase, Ui registers with RC by adopting the following steps:

1. The Ui generates two random numbers ω and Ni. Then, it calculates $\underline{TPWi}=h(IDi \,||h(\, PWi||\, Ni))$ and $\omega \oplus \underline{TPWi}$. Next, it submits IDi and $\omega \oplus \underline{TPWi}$ to RC, using a secure channel.
2. The RC receives $\{IDi, \omega \oplus \underline{TPWi}\ \}$, computes $Di=h(IDi \,||\, h(PSK))$ and $Ci' = Di \oplus \omega \oplus \underline{TPWi}$, and sends $\{Ci'\}$ to Ui after storing in SC.
3. Next, the user imprints biometric $BIOi$ and computes $Gen(BIOi) \rightarrow (\beta_i, \gamma_i)$. Next, it further calculates $Ri=h(IDi \,||\, h(PWi \,||Ni))$, $Bi=Ni \oplus h(\beta_i)$ and $Ci= \omega \oplus Ci'$ Next, it stores the parameters in SC which now contains $\{Ri, Ci, Bi, \gamma_i\ h()\}$, finally.

4.2 Login and authentication phase

1. In login stage, Ui inputs IDi, PWi into SC for verifying its authenticity to avail services of Sj. Next, the user imprints $BIOi$ and computes $Rep\,(BIOi^*, \gamma_i) \rightarrow \beta_i$, $Ni=Bi \oplus h(\beta_i)$. Then, it verifies the equality for $Ri \stackrel{?}{=} h(IDi \,||\, PWi \,||\, Ni)$. On successful verification, the SC allows Ui to proceed for login phase. Then, it further calculates $TPWi =h(IDi \,||h(\, PWi||\, Ni))$. Next, Ui generates a random integer n_1, and calculates $Di = Ci \oplus TPWi$, $M_1=E_{pub}\{IDi, n_1P, h(PWi \,||Ni)\}$, and $M_2=h(h(Di) \,||\, n_1P \,||h(PWi \,||Ni)||\, T_1)$. Finally, it sends $\{M_1, M_2, T_1\}$ to Sj.
2. In the authentication phase the Sj receives parameters and checks the equality $T_2-T_1 > \Delta T$, ΔT being the threshold for timestamp. If the difference surpasses threshold ΔT, Sj terminates the session, otherwise computes $\{IDi, n_1P, h(PWi \,||Ni)\}$ by decrypting M_1 i.e., $D_{Prs}\{M_1\}$. Next, Sj computes $Di^*=h(IDi \,||\, h(PSK))$, $M_2^*= h(Di^*|| \, n_1P \,||h(PWi||Ni)||T_1)$. Now, it compares the equality $M_2^* \stackrel{?}{=} M_2$. On successful verification, it generates n_2, and computes $M_3=n_2P \oplus h(PWi||Ni)$ and $M_4 =h(IDi \,||\, n_2P \,||\, h(PWi||Ni)||\, T_3)$. The Sj sends $\{M_3, M_4, T_3\}$ to Ui, and calculates the session key as $h(n_1n_2P \,||\, h(PWi||Ni) \,||\, IDi\,)$, finally.
3. Ui , after receiving the message from Sj, compares timestamp against the threshold $T_4-T_3 > \Delta T$. If this is true, aborts the session. Otherwise it constructs $n_2P = M_3 \oplus h(PWi \,||\, Ni)$ and $M_4^* =h(IDi \,||\, n_2P \,||\, h(PWi||Ni)||T_3)$. Now it checks the equality for $M_4^* \stackrel{?}{=} M_4$, if the match fails, it aborts the session, otherwise constructs the session key as $h(n_1n_2P \,||\, h(PWi||Ni) \,||\, IDi\,)$.

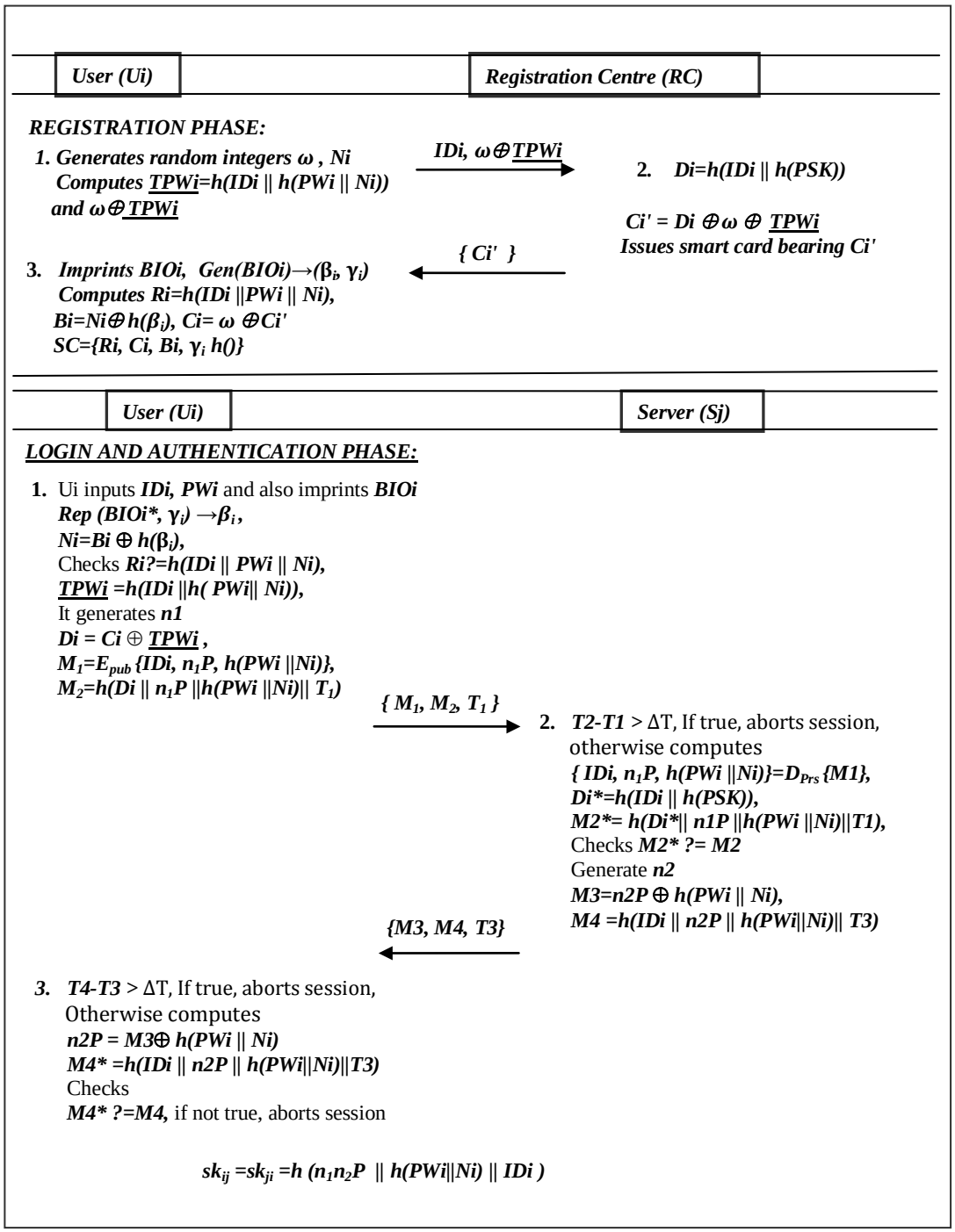

Fig. 2. Proposed Authentication Protocol

4.3 Password update phase

Ui could update its password into a novel password, i.e. PWi^{new} by initiating a procedure, which does not require any interaction with RC [14-16, 19, 22-23]. The procedure is mentioned below:

1. Ui gives as input, its identity (*IDi)*, password (*PWi)* in SC and imprints the biometric pattern *BIOi* into a device.
2. Then, SC computes *Rep (BIOi*,* γ_i*)* $\rightarrow \beta_i$ and $Ni^*=Bi \oplus h(\beta_i)$. Next, it checks whether $Ri^* ?= h(IDi \,||\, PWi \,||\, Ni^*)$ holds, if it does not hold true, the SC refuses to proceed for changing password; Otherwise, Ui inputs a new password PWi^{new}, SC computes a new parameter $Ri^{new} = h(IDi \,||\, PWi \,||\, Ni)$ and would replace *Ri* with Ri^{new} to finalize the password modification.

5. Automated Tool Security Verification

The objective for security verification using an automated tool is to analyze the proposed scheme's immunity against a malicious adversary. ProVerif [41, 42] has been accepted as one of the effective tools by the research academia to test the protocols' immunity against attacks, privacy, and session key secrecy. ProVerif is based on widely accepted applied π calculus which is capable of supporting different cryptographic primitives like one-way operations, digital signatures, encryption, Diffie-Helman etc. We make a thorough analysis for measuring the efficiency and security of contributed scheme using Pro Verif simulation.

We proceed in this simulation after defining the two channels such as *SeCh*: a private channel, and *PbCh*: a public channel, among the entities RC, Sj and Ui.

```
free SeCh: channel [private].
free PbCh: channel.
```

Some variables and constants are used in the contributed model as demonstrated under:

```
const P: bitstring.
free IDi: bitstring.
free g: bitstring.
free PWi: bitstring [private].
free x: bitstring [private].
free Pub: bitstring.

free Prs: bitstring [private].
free PSK: bitstring [private].
free BIOi: bitstring [private].
```

We employed some constructors in the simulation as CONCAT, XOR, h, ECPM, ENC, and Exp are defined as concatenation, exclusive-OR, one-sided hash, elliptic curve-based scalar point multiplication, asymmetric key encryption, and exponentiation function, respectively. Here, for asymmetric decryption, we define DEC that executes decryption using a different key. Gen and Rep are the fuzzy extractor functions. Another function XOR is used for exclusive-OR, i.e. XOR(XOR(c,d),d)=c. The authentication primitives including constructors and destructors in this simulation are modeled for the contributed scheme as follows.

```
fun Gen(bitstring): bitstring.
fun Rep(bitstring, bitstring):  bitstring.
fun h(bitstring): bitstring.
fun XOR(bitstring, bitstring):bitstring.
fun CONCAT(bitstring, bitstring):bitstring.
fun ENC(bitstring, bitstring):bitstring.
fun ECPM(bitstring, bitstring):bitstring.
fun Exp(bitstring, bitstring):bitstring.

equation forall c:bitstring,d:bitstring; XOR(XOR(c,d),d)=c.
reduc forall o: bitstring, key: bitstring; DEC(ENC(o, Pub),Prs)=o.
```

We initiate the simulation modeling by creating two events for each of the participant i.e.Ui and Sj. The events, beginUserUi(bitstring) and endUserUi(bitstring) represent start and end events for Ui, whereas the events, beginServerSj(bitstring) and endServerSj(bitstring) represent the same for Sj. The contributed scheme's authenticity may be evaluated by analyzing the corresponding link between start and end events for any participant. We describe these events as given below.

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

The three different processes are modeled, i.e. *RegistrationCentreRC*, *UserUi*, and *ServerSj* against RC, Ui and Sj, respectively. First, the *UserUi* process forwards the parameters IDi, PWi' on confidential channel SeCh to *ServerSj* process. Next, after getting xCi, *UserUi* further calculates Ri and Bi. In mutual authentication phase, *UserUi* compares Ri and Ri' after calculating Ri'. It further computes PWi'' and Di, which are used in the construction of M1 and M2. The *UserUi* then submits the message (M1, M2, T1) to *ServerSj* using the open channel PbCh. Finally, the *UserUi* , after receiving the message (M3, M4, T3) from SeverSj, computes n2P', M4' and compares xM4' with M4' and validates the ServerSj process, otherwise, aborts the session. Then it proceeds for calculating the session key SK.

```
let UserUi=
new Ni:bitstring;
new w:bitstring;
let Pub=Exp(g,Prs) in
let PWi' = h(CONCAT(IDi,h(CONCAT(PWi,Ni)))) in
let W= XOR(w, PWi') in
```

```
out (SeCh,(IDi, W));
in(SCh,(xCi*:bitstring));
let (u: bitstring, v: bitstring) = Gen(BIO_i) in
let Ri = h(CONCAT(IDi,(PWi,Ni))) in
let Ci= XOR(w,xCi*) in
let Bi= XOR(Ni,h(u)) in
event beginUserUi(IDi);
new BIO_i*: bitstring;
let u* = Rep(BIO_i*, v) in
let Ni'= XOR(Bi, h(u*)) in
let Ri' = h(CONCAT(IDi,(PWi,Ni'))) in
if (Ri = Ri') then
let PWi'' = h(CONCAT(IDi,h(CONCAT(PWi,Ni')))) in
new n1:bitstring;
let Di = XOR(xCi,PWi'') in
let M1 = ENC(CONCAT(IDi,(ECPM(n1,P),h(CONCAT(PWi,Ni')),Di)),Pub) in
new T1:bitstring;
let M2 = h(CONCAT(h(Di),(ECPM(n1,P),h(CONCAT(PWi,Ni')),T1))) in
out(PbCh,(M1,M2,T1));
in(PbCh,(xM3:bitstring,xM4:bitstring,xT3:bitstring));
let n2P' = XOR(xM3,h(CONCAT(PWi,Ni'))) in
let M4' = h(CONCAT(IDi,(n2P',h(CONCAT(PWi,Ni')),xT3))) in
if(xM4 = M4') then
let SK = h(CONCAT(ECPM(n1,n2P'),(h(CONCAT(PWi,Ni')),IDi))) in
event endUserUi(IDi)
else
0.
```

The process *RegistrationCentreRC* gets xIDi and xPWi' parameters from *UserUi* using a confidential channel SeCh, and calculates Di and Ci= XOR(Di, xPWi'). Finally, it sends Ci towards *UserUi* process using SeCh.

```
let RegistrationCentreRC=
in (SeCh,(xIDi:bitstring,xW:bitstring));
let Di = h(CONCAT(xIDi,h(PSK))) in
let Ci* = XOR(Di,xW) in
out(SeCh,(Ci*));
0.
```

The *ServerSj* process receives the parameters xM_1, xM_2 and xT_1 from *UserUi* process for verifying authenticity. Next, it decrypts xM1 using its private key (Prs), and recovers < xIDi, xn_1P, xhPN, xDi > tuple. Next, it computes Di', M_2' and compares xM_2 against M_2'. If found true, then it validates the *UserUi* process, otherwise aborts the session. Following the positive verification of *UserUi* process, it further computes M_3 by doing XOR (ECPM(n_2,P),h(xhPN)) and M_4. Then this process sends the message (M_3, M_4, T_3) towards *UserUi* using a public channel. Then the *UserUi* process verifies the authencity of received message, finally.

```
let ServerSj=
event beginServerSj(Pub);
in(PbCh,(xM1:bitstring,xM2:bitstring,xT1:bitstring));
let (xIDi:bitstring,xn1P:bitstring,xhPN:bitstring,xDi:bitstring)=DEC(xM1,Prs) in
let Di' = h(CONCAT(xIDi,h(PSK))) in
new T2:bitstring;
let M2' = h(CONCAT(h(Di'),(xn1P,xhPN,T2))) in
if (xM2 = M2') then
new n2:bitstring;
let M3 = XOR(ECPM(n2,P),h(xhPN)) in
new T3:bitstring;
let M4 = h(CONCAT(IDi,(ECPM(n2,P),xhPN,T3))) in
out(PbCh,(M3,M4,T3));
event endServerSj(Pub)
else
0.
```

The three principals get agreed to an unrestricted number of sessions in parallel, for these processes will be in replication as depicted below.

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

We define the following queries to test the security and correctness of the proposed protocol.

```
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 results are the outcome of implementing above queries in this simulation.

```
RESULT inj-event(endServerSj(id)) ==> inj-event(beginServerSj(id)) is true.
RESULT inj-event(endUserUi(id_1580)) ==> inj-event(beginUserUi(id_1580)) is true.
RESULT not attacker(SK[]) is true.
```

The first two results clearly identity that these processes begin and also end successfully, whereas the third outcome suggests that the attacker-based query is unable to output the session key as constructed among the processes during mutual authentication procedure.

6. SECURITY ANALYSIS

This section demonstrates security analysis of contributed scheme in the follow up of discovered threats in Lu et al.'s protocol. This analysis depicts that the proposed work is resistant to all of the threats as listed in **Table 2**, particularly malicious insider attack and perfect forward secrecy compromise as posed to Lu et al. scheme. The **Table 2** and III list the functionality comparison and the computational cost of different schemes. Before delving into

an informal security analysis, first, we define few terms [36] as used in the analysis.

Definition 1

An Elliptic Curve Computational Diffie–Hellman problem (ECCDHP) is acknowledged as: Given σ G's generator $P \epsilon E_q$, σP, $\mu P \epsilon E_q$, it is infeasible to compute $\sigma\mu P \epsilon E_q$ without having the knowledge of $\sigma \epsilon Z_P^*$ or $\mu \epsilon Z_P^*$.

Definition 2

The Elliptic Curve Discrete Logarithm Problem (ECDLP) is acknowledged as: Given a point $\Theta=\sigma P$ on Elliptic Curve, it is intractable to figure out the scalar σ, given $\Theta \epsilon E_q$ and $P \epsilon E_q$.

The informal security analysis of the contributed model is elaborated as follows:

Proposition 1. *The proposed scheme thwarts replay attack.*

In replay attacks, the attacker replays the intercepted messages at any time to misrepresent the legitimate entities.

Proof. An adversary $\mathbb{A}$ having the intercepted contents $\{M_1, M_2, T_1, M_3, M_4, T_3\}$ may attempt to replay the parameters to betray the legitimate participants. Nonetheless, the use of T_1 and T_3 timestamps by either of the participants debars an adversary to launch such attack. Since, if $\mathbb{A}$ will employ the same timestamp for generating a fake login request, it fails the threshold. On the other hand if adversary generates its own message with a new timestamp, it fails the equality check $M_2^* \stackrel{?}{=} M_2$, subsequently. Hence, this fact proves the above proposition that proposed scheme can successfully thwart a replay attack.

Proposition 2. *The contributed protocol is resistant to Man-in-the-Middle attack.*

In this threat, a silent intermediary manipulates the communication on both ends by replaying or constructing messages. In this attack, the legal participants believe erroneously that these are talking to an intended recipient.

Proof. In contributed scheme, the adversary is not able to initiate MiTM attack, since none of the intermediaries could apporach the parameters included in a message i.e., IDi, $h(PWi \,||Ni)$, and Di, using a public channel. Hence, it may not be able to construct the original message $M_1=E_{pub}\{IDi, n_1P, h(PWi \,||Ni)\}$, and $M_2=h(Di \,||\, n_1P \,||h(PWi \,||Ni)||\, T_1)$ with a fresh timestamp, that makes the server believe that $\mathbb{A}$ is a legitimate entity.

Proposition 3. *Our scheme prevents modification threat.*

The modification attacks could be initiated by an attacker if it resends the message after reconstructing it in an unauthorized manner towards a legal entity.

Proof. If any adversary tries to modify the intercepted messages $\{M_1, M_2, T_1, M_3, M_4, T_3\}$, it may not change the messages M_1-M_4 except T_1 and T_3 parameters. However, an adversary may not be able to perform any timestamp based update in M_1-M_4 messages, since the adversary's modification might be successful in passing the timestamp threshold check ΔT, nevertheless, it may not be able to pass the $M_2^* \stackrel{?}{=} M_2$ and $M_4^* \stackrel{?}{=} M_4$ checks. Hence, it proves the proposition that the proposed scheme prevents modification attacks.

Proposition 4. *The contributed work is resistant to password-guessing attack.*

Proof. The offline-password guessing attack [26-30] may be tried when an attacker attempts to extract or compute Ui's password by employing the intercepted messages *{ M_2, M_4}* or stolen smart card information i.e. *{Ri, Ci, Bi}*. However, adversary may not be able to guess PWi from *M2=h(Di || n1P ||h(PWi ||Ni)|| T1)* or *M4 =h(IDi || n_2P || h(PWi||Ni)|| T_3)*, by the reason of not having access to *Di*, n_1P, *IDi*, *Ni* and n_2P parameters. At the same time, password computation or dictionary guessing attack, from *Ri=h(IDi || h(PWi || Ni)), Ci = Di* ⊕ *TPWi* or *Bi=Ni*⊕ *h(βi)* is not possible until *Ni* and *IDi* factors are known. Where, *Ni* can only be extracted with the imprinted biometric *BIOi* and extracted parameter *βi*, while an access to the latter is definitely an unfeasible task. Hence, the above proposition is proved.

Proposition 5. *The contributed work is immune to stolen-verifier attack.*

Proof. An attacker might steal secret information stored on a server's repository and exploit it for some malicious purpose, however, only if the server maintains the repository of users' verifiers and shared secrets. While, the contributed protocol foregoes the maintenance of any kind of database on server's or RC's end, which makes the proposed scheme naturally immune to stolen-verifier attack. Hence, this fact sustains the above proposition.

Proposition 6. *The proposed scheme foils the offline dictionary attack, in case the user's smart card gets stolen.*

In offline dictionary threat, the adversary after stealing SC, attempts to utilize the extracted contents for guessing low-entropy secrets by inputting all possible combinations from the dictionary.

Proof. An adversary after making away with the stolen contents of smart card, may attempt to use for some malicious purpose. Nevertheless, those contents including *Ri, Ci, Bi* serve to be useless, given that *PWi* recovery or dictionary guessing attack, from *Ri=h(IDi || PWi || Ni))* is not possible until the parameters *Ni* and *IDi* are recovered. *Ni* can neither be extracted until the parameters *βi* and n_2P are accessed. Likewise, *Ci* is also a function based on XOR i.e., *Ci = Di* ⊕ *TPWi*. The *PWi* cannot be guessed out of *Ci* or *TPWi* in any manner, by the adversary. Thus, this proof upholds the above proposition.

Proposition 7. *The contributed scheme achieves session-key security.*

This security feature affirms that the established session key is known merely to the authorized participants, i.e., Ui and server.

Proof. In contributed model, the constructed session key is based on *SK =h(n_1n_2P || h(PWi||Ni) || IDi)*. To construct a legal session key, the attacker has to access n_1, n_2, *IDi* and *h(PWi ||Ni)*. An adversary cannot derive n_1 from n_1P, neither n_1P from M_1, which is encrypted using the public of Sj. At the same time, it also needs *IDi* and *h(PWi ||Ni)* to create a valid session key. Hence, the above proposition is proved.

Proposition 8. *The contributed protocol maintains the attribute of known-key security.*

The known-key security feature points to the inability of guessing private keys of the associated participants, subject to the compromised session key.

Proof. If we assume, the session key $SK = h(n_1 n_2 P \,||\, h(PWi||Ni) \,||\, IDi)$ gets compromised by attacker, yet it may not be able to guess the secret keys of legal participants. The server private key *Prs* is quite safe, since the Ui makes a use of its public key for encryption, and Sj uses this *Prs* for decryption. Likewise, the Ui's password *PWi* cannot be guessed until *Ni*, *IDi* and $n_1 n_2 P$ are not known to the adversary. Hence, for the known-key security, the above proposition is proved.

Proposition 9. *The contributed scheme stands compiant to perfect forward secrecy.*

This security feature affirms the security of past session keys, in case the participants' long-term secrets are exposed.

Proof. The contributed work provides perfect forward secrecy, in case the private keys of one or more participants gets exposed, since, the contributed work employs ECC operations to ensure the ECDLP property that in turn leads to forward secrecy. In case, the adversary gets the participants' (Server/RC/User) secrets such as *Prs*, *PSK*, and *PWi* of server, yet A is not able to compute the previous session keys for hardness of computing either n_1 out of $n_1 P$, or n_2 from $n_2 P$ due to ECDLP and ECCDHP. Hence, the proposition is proved.

Proposition 10. *The proposed scheme mutually authenticates the intended participants.*

This feature stipulates that the parties involved should verify one another in the same protocol.

Proof. The contributed work affirms mutual authentication to involved participants as also proved in section 7 (BAN logic). The Sj starts authenticating Ui on the basis of received message $\{M_1, M_2, T_1\}$. Then it decrypts M_1 i.e., $\{IDi, n_1 P, h(PWi \,||Ni)\} = D_{Prs}\{M_1\}$, computes $Di^* = h(IDi \,||\, h(PSK))$, and $M_2^* = h(Di^* \,||\, n_1 P \,||h(PWi||Ni)||T_1)$. Finally, Sj compares $M2^* \,?= M2$, and authenticates on successful equality check. Otherwise, aborts the session. Likewise, Ui authenticates Sj on the basis of received message $\{M_3, M_4, T_3\}$. Ui extracts $n_2 P$ from M_3 by $n_2 P = M_3 \oplus h(PWi \,||\, Ni)$, and then computes $M_4^* = h(IDi \,||\, n_2 P \,||\, h(PWi||Ni)||T_3)$. Finally it compares the equality check $M_4^* \,?= M_4$, if true, authentication with Sj is validated. Otherwise, aborts the session.

Proposition 11. *The contributed work stipulates the user's anonymity and privacy.*

An anonymous authentication protocol should not mitigate the server's chances of verifying the user's authenticity. After the exchange of messages on an insecure channel during login and authentication phase, and successful session key establishment onwards, an attacker won't be able to discern about the identities of involved participants by examining the intercepted messages [43].

Proof. In proposed model, the user submits its identity *IDi* in message $\{M_3\}$ by encrypting through Sj's public key. This prevents any possibility of Ui's leakage of identity which substantiates the proposition.

Proposition 12. *The contributed work provides defense against previliged insider attack.*

The previliged insider attack could be initiated if a malicious adversary (priveleged insider on the registration authority's end) gets access to communicated parameters during registration process, and further launch impersonation attack and session key guessing attacks.

Proof. In proposed model, if a previliged malicious insider gets access to *IDi* and $\omega \oplus \underline{TPWi}$ parameters, it cannot guess either password from $\omega \oplus \underline{TPWi}$ or initiate any kind of impersonation attack. Moreover, in case, the adversary gets the smart card contents, it cannot compute *Di* from *Ci* on the basis of pre-stolen $\omega \oplus \underline{TPWi}$ during registration processs. Since, *Di* is the basis of mutual authentication between the participants, the inaccessibility of *Di* to adversary debars it to lauch any kind of impersonation attack. Therefore, our scheme provide resistance to previliged insider attack, proving the above proposition.

Now we present the security evaluation related to contributed work using Burrows-Abadi-Needham logic (BAN) logic [25]. This logic based model enables us to analyze authentication protocols in terms of session key generation in a secure manner, and mutual authentication between the intended participants.

In this logical analysis, principals ($\mathcal{A}$ and $\mathcal{B}$) refer to generic instances or agents, participating in a protocol. Some related notations w.r.t logical analysis are given as follows:

$\mathcal{A} \mid\equiv$ M: $\mathcal{A}$ believes the statement M.
$\mathcal{A} \lhd$ M: $\mathcal{A}$ sees M. $\mathcal{A}$ receives the message M and reads or may put it to any use.
$\mathcal{A} \mid\sim$ M: $\mathcal{A}$ once said M. Some time ago, $\mathcal{A}$ had sent some message M after generating it.
$\mathcal{A} \Rightarrow$ M: $\mathcal{A}$ enjoys jurisdiction over M; or $\mathcal{A}$ can influence M and be trusted.
$\sharp$ (M): The message M is fresh and not replayed.
$\langle M \rangle_{\mathcal{Z}}$: The formulae M is used in combination with formulae $\mathcal{Z}$.
(M, $\mathcal{Z}$): M or $\mathcal{Z}$ being the part of message (M, $\mathcal{Z}$).
$\{M, \mathcal{Z}\}_{\mathcal{S}}$: M or $\mathcal{Z}$ is encrypted using symmetry key $\mathcal{S}$.
$\langle M, \mathcal{Z} \rangle_{\mathcal{S}} \mapsto \mathcal{A}$: M or $\mathcal{Z}$ is encrypted using public key $\mathcal{S}$ of $\mathcal{A}$.
$(M, \mathcal{Z})_{\mathcal{S}}$: M or $\mathcal{Z}$ is hashed using the key $\mathcal{S}$.
$\mathcal{A} \overset{\mathcal{S}}{\longleftrightarrow} \mathcal{B}$: $\mathcal{A}$ and $\mathcal{B}$ can securely contact using shared key $\mathcal{S}$.

The assumptions or postulates related to current analysis are illustrated below:

R1. Message meaning rule: $\dfrac{\mathcal{A} \mid\equiv \mathcal{A} \overset{\mathcal{S}}{\leftrightarrow} \mathcal{B},\ \mathcal{A} \lhd \langle M \rangle_{\mathcal{Z}}}{\mathcal{A} \mid\equiv \mathcal{B} \mid\sim M}$

R2. Nonce verification rule: $\dfrac{\mathcal{A} \mid\equiv \sharp(M),\ \mathcal{A} \mid\equiv \mathcal{B} \mid\sim M}{\mathcal{A} \mid\equiv \mathcal{B} \mid\equiv M}$

R3. Jurisdiction rule: $\dfrac{\mathcal{A} \mid\equiv \mathcal{B} \Rightarrow M,\ \mathcal{A} \mid\equiv \mathcal{B} \mid\equiv M}{\mathcal{A} \mid\equiv M}$

R4. Freshness conjuncatenation rule: $\dfrac{\mathcal{A} \mid\equiv \sharp(M)}{\mathcal{A} \mid\equiv \sharp(M, \mathcal{Z})}$

R5. Belief rule: $\frac{A|\equiv(M),\ A|\equiv(\_)}{A|\equiv(M,\ \_)}$

R6. Session key rule: $\frac{A|\equiv \square(M),\ A|\equiv \forall |\equiv M}{A|\equiv A \overset{s}{\longleftrightarrow} \forall}$

R7. Public key encryption rule: $\frac{A|\equiv {}_s\mapsto\forall,\ A\triangleleft \{M\}_{s^{-1}}}{A|\equiv \forall |\sim M}$

Our proposed scheme needs to follow these goals for proving its session key-based security using BAN logic, keeping in view the above assumptions.

Goal1 : Sj |≡ Ui $\overset{SK}{\longleftrightarrow}$ Sj
Goal2 : Sj |≡ Ui |≡ Ui $\overset{SK}{\longleftrightarrow}$ Sj
Goal3 : Ui |≡ Ui $\overset{SK}{\longleftrightarrow}$ Sj
Goal4 : Ui |≡ Sj |≡ Ui $\overset{SK}{\longleftrightarrow}$ Sj

The generic protocol can be described as:

m_1: Ui → Sj: M_1, M_2, T_1
m_2: Sj → Ui: M_3, M_4, T_3

We adapt the generic protocol into idealized form as given below.
m_1: Ui → Sj: $\langle IDi, n_1P, h(PWi||Ni), Di\rangle_{Pub} \mapsto Sj$, $\langle n_1P, h(PWi||Ni), T_1\rangle_{Di}$, T_1
m_2: Sj → Ui: $\langle n_2P\rangle_{h(PWi||Ni)}$, $\langle IDi, n_2P, T_3\rangle_{h(PWi||Ni)}$, T_3

Secondly, the undermentioned premises are set up for proving the robustness of our scheme.

P1 : Ui |≡ ♯ n_1, T_1
P2 : Sj |≡ ♯ n_2, T_3
P3 : Ui |≡ Sj $\overset{Di}{\longleftrightarrow}$ Ui
P4 : Sj |≡ Sj $\overset{Di}{\longleftrightarrow}$ Ui
P5 : Ui |≡ Sj ⇒ n_2P
P6 : Sj |≡ Ui ⇒ n_1P

By using the above notations, rules, premises and idealizations, we get to the following proofs and derivations:

Mutual Authentication accuracy:

For verifying mutual authentication between Ui and Sj, we visualize the messages m1 and m2, stating the idealized form:
m_1: Ui → Sj: $\langle IDi, n_1P, h(PWi||Ni), Di\rangle_{Pub} \mapsto Sj$, $\langle n_1P, h(PWi||Ni), T_1\rangle_{Di}$, T_1
m_2: Sj → Ui: $\langle n_2P\rangle_{h(PWi||Ni)}$, $\langle IDi, n_2P, T_3\rangle_{h(PWi||Ni)}$, T_3

Lemma 1: *Sj can correctly prove the authenticity of login request message from Ui.*

Proof. User Ui generates the message (M_1, M_2, T_1) and sends towards server Sj in order to login it and avail its services. Sj gets timestamp along with some other session based parameters and verify the correctness for the source of the message as follows.

We apply the seeing rule, and get the derivation

D1: Sj $\lhd \langle IDi, n_1P, h(PWi \,||Ni), Di\rangle_{Pub} \mapsto Sj$, $\langle n_1P, h(PWi \,||Ni), T_1 \rangle_{Di}$, T_1

Applying D1, P4 and R1,

D2: Sj $|\equiv$ Ui $\sim n_1P, T_1$

According to P1, P6, and R4

D3: Sj $|\equiv \sharp(n_1P, T_1)_{Di}$

According to D2, D3 and R2, we have

D4: Sj $|\equiv$ Ui $|\equiv (n_1P, T_1)_{Di}$

According to P4, D4 and the application of R3, we can say

D5: Sj $|\equiv n_1P, T_1$

Hence, after verifying the timestamp freshness, Sj proves the accuracy of message source.

Lemma 2: *Ui can appropriately prove the authenticity of response message from Sj.*

Proof. In our protocol, the server Sj generates the message (M_3, M_4, T_3) and sends to Ui along with timestamp, in response to its Ui's login request. Ui proves the authenticity of Sj by checking the parameters freshness as follows.

By applying seeing rule, we get to this derivation,

D6: Ui $\lhd \langle n_2P\rangle_{h(PWi \,||Ni)}$, $\langle IDi , n_2P , T_3\rangle_{h(PWi \,||Ni)}$, T_3

According to D6, P3 and R1,

D7: Ui $|\equiv$ Sj $\sim n_2P, T_3$

According to P2, P5, and R4

D8: Ui $|\equiv \sharp(n_2P, T_3)_{Di}$

According to D7, D8 and R2, we have

D9: Ui $|\equiv$ Sj $|\equiv (n_2P, T_3)_{Di}$

According to P3, D9 and the application of R3, we can say

D10: Ui $|\equiv n_2P, T_3$

Hence, after verifying the timestamp freshness, Ui proves the accuracy of message source.

Theorem 1.

Proof. In relation to Lemma 1, the Sj may correctly prove the authenticity of a login request from user. In relation to Lemma 2, the user Ui may also correctly prove the authenticity of the response message from the server. Hence, we may deduce that Ui and Sj mutually authenticate one another.

Session Key Agreement:

A single session key, $Sk = h(n_1 n_2 P \ || \ h(PWi||Ni) \ || \ IDi)$, can be established and agreed upon between the communicating entities in proposed protocol. While, $(IDi, n_1 n_2 P, h(PWi||Ni))$ are necessary parameters for session key generation. This session key agreement between the participants can be achieved as follows.

According to P2, D4, and R2, we get

D11: Sj $|\equiv$ Ui $|\equiv$ Sj $\xleftrightarrow{SK}$ Ui **(Goal 2)**

According to P2, D11, and R6

D12: Sj $|\equiv$ Sj $\xleftrightarrow{SK}$ Ui **(Goal 1)**

According to P1, D9, and R2, we get

D13: Ui $|\equiv$ Sj $|\equiv$ Sj $\xleftrightarrow{SK}$ Ui **(Goal 4)**

According to P1, D13, and R6

D6: Ui $|\equiv$ Sj $\xleftrightarrow{SK}$ Ui **(Goal 3)**

Hence, the above analysis (BAN) sufficiently verifies that the contributed protocol can achieve mutual authenticity, while the established session key *SK* is mutually agreed between the legal participants (Ui and Sj).

Formal Security Analysis

We conduct a formal analysis with random oracle model, that validates the argument in the favor of a secure model [37-40]. For this purpose, we can define the hash function *h(.)* as follows.

Definition 3 *We define a one-sided function, hash as* h_f: $\{0, 1\}^* \rightarrow \{0, 1\}^\ell$, *that generates a binary string of length* ℓ *i.e.* $h_f(y) = \{0, 1\}^\ell$ *as output, on providing a randomly sized binary string* $y \in \{0, 1\}^\ell$ *as input.*

We define the following two oracles that could be used by an adversary $\mathcal{A}$, and outputs unconditionally as following:

reveal1: This oracle produces δ out of the corresponding hash value $\vartheta = h_f(\delta)$, unconditionally.

reveal2: The *reveal2* oracle outputs scalar *k* from public key $\mathfrak{Q} = kP$, unconditionally, given $P \in E_q(a, b)$.

Algorithm 1. $EXP1_{PSBASME}^{HASH}$

1. Eavesdrop the Login request message { M_1, M_2, T_1 } in the login phase, where $M_1=E_{pub}\{IDi, n_1P, h_f(PWi||Ni)\}$ and $M_2= h_f(Di || n_1P ||h_f(PWi||Ni)|| T_1)$
2. Call Reveal oracle on the input *M2* to produce Di', n_1P, $h(PWi||Ni)'$, T_1' as $(Di || n_1P || h_f(PWi||Ni)'|| T_1') \leftarrow reveal1\ (M_2)$
3. Call Reveal oracle on the input Di' to produce IDi', $h_f(PSK)$ as $(IDi' || h_f(PSK)) \leftarrow reveal1\ (Di')$
4. **If** $(T_1'=T_1)$ **Then**
5. Eavesdrop the Login request message $\{M_3, M_4, T_3\}$ in the authentication phase, where
   $M_3=n_2P \oplus h_f(PWi || Ni)$ and $M_4 = h_f(IDi || n_2P || h_f(PWi||Ni)|| T_3)$
6. Call Reveal oracle on input M_4 to retrieve IDi, n_2P', $h_f(PWi||Ni)''$, T_3' as $h_f(IDi || n_2P || h_f(PWi||Ni) || T_3') \leftarrow reveal1\ (M_4)$
7. **If** $(T_3'=T_3)$ and $(h_f(PWi||Ni)' = h_f(PWi||Ni)'')$**Then**
8. Compute $n_2P^*= h_f(PWi||Ni)' \oplus M_3$
9. **If** $(n_2P^*=n_2P')$ and $(IDi'=ID)$ **Then**
10. Accept *IDi* as the true identity of user, and *PSK* as the valid shared secret between RC and Sj.
11. Return 1 (True)
12. Else
13. Return 0 (False)
14. End if
15. **End if**

Theorem2

By undertaking ECDLP assumption, given that one-sided hash function performs narrower to a random oracle, the contributed technique stands protected against attacker Ⱥ, if Ⱥ tries to derive the identity (IDi) of some user (Ui) and shared secret PSK between RC and Sj.

Proof.

Here, we plan to set an attacker Ⱥ, capable of deriving the Ui's original IDi and shared secret PSK between RC and Sj, by employing random oracles Reveal1, Reveal2 and running the experiment as shown in algorithm $EXP1_{PSBASME}^{HASH}$. The success probability regarding $EXP1_{PSBASME}^{HASH}$ is Sucss1=Pro.2[$EXP1_{PSBASME}^{HASH}$ =1]-1, while Pro[E_v] characterize the probability of an event E_v. The gain function for the current experiment turns out to be $Adv_{PSBASME}^{HASH}$ (t_{m1},$q_{Ry1,}$ q_{Ry2})=$\max_A$ [$Sucss1_{PSBASME}^{HASH}$], having execution time t_{m1} while the Reveal-queries q_{Ry1} and q_{Ry2} maximized on attacker (Ⱥ). We term our contributed technique to be protected of Ⱥ for deriving *IDi*, *PSK*, if $Adv_{PSBASME}^{HASH}(t_{m1}, q_{Ry1,} q_{Ry2}) \leq \propto$ for adequately small $\propto > 0$. In accordance to the current experiment, if Ⱥ could invert a one-sided hash function h(.), then solving the hard problem ECDLP, onwards it could comfortably recover the valid *IDi* and shared secret *PSK* among the participants, and finally wins this game. Nonetheless, in keeping with definition (2), this would not be computationally viable to reverse the related hash-based function, the reason being $Adv_{PSBASME}^{HASH}(t_{m1}) \leq \propto$ for adequately small $\propto > 0$.

Algorithm 2 $EXP2_{PSBASME}^{HASH,SC}$

1. Eavesdrop the Login request message { M_1, M_2, T_1 }, where $M_1=E_{pub}\{IDi, n_1P, h_f(PWi||Ni)\}$ and $M_2=h_f(h_f(Di)||n_1P||h_f(PWi||Ni)||T_1)$
2. Call the oracle for input M_2 to produce Di', n_1P, $h_f(PWi||Ni)'$, T_1' as $(Di||n_1P||h_f(PWi||Ni)'||T_1')\leftarrow reveal1(M_2)$
3. Call the oracle for input n_1P to produce n_1' as $n_1'\leftarrow reveal2(n_1P)$
4. Call the oracle for input Di' to produce IDi', h_f (PSK) as $(IDi'||h_f(PSK))\leftarrow reveal1(Di')$
5. **If** $(T_1'=T_1)$ **Then**
6. Eavesdrop the Login request message $\{M_3, M_4, T_3\}$ in the authentication phase, where
   $M_3=n_2P\oplus h_f(PWi||Ni)$ and $M_4=h_f(IDi||n_2P||h_f(PWi||Ni)||T_3)$
7. Call Reveal oracle on input M_4 to produce IDi, n_2P, $h_f(PWi||Ni)''$, T_3' as $h_f(IDi||n_2P||h_f(PWi||Ni)||T_3')\leftarrow reveal1(M_4)$
8. **If** $(T_3'=T_3)$ and $(h_f(PWi||Ni)'=h_f(PWi||Ni)'')$**Then**
9. Call Reveal oracle on input n_2P to produce n_2' as $n_2'\leftarrow reveal2(n_2P)$
10. Compute $n_2P^*=h_f(PWi||Ni)'\oplus M_3$
    Compute $SK=h_f(n_1'n_2'P||h_f(PWi||Ni)'||IDi)$
11. If $(n_2P^*=n_2'P)$ and $(IDi'=IDi)$ **Then**
12. Accept SK as rightly agreed session key for Sj and Ui.
13. Return 1 (True)
14. Else
15. Return 0 (False)
16. End if
17. End if

Theorem 3

By taking ECDLP as assumption, as one-sided hash function stands too close to a random oracle, the contributed protocol stays protected, if some attacker goes malicously for deriving an agreed session key (SK) between participants.

Proof.

Here, we plan to set $\mathbb{A}$, capable of deriving the agreed session key (SK) between Sj and Ui, by engaging random oracles Reveal1, Reveal2, and running the experiment as shown in algorithm $EXP2_{PSBASME}^{HASH,SC}$. The probability of success against $EXP2_{PSBASME}^{HASH,SC}$ is Sucss1=Pro.2[$EXP2_{PSBASME}^{HASH,SC}$=1] - 1, while Pro[E] shows the probability of some event E. The corresponding advantage function of the related experiment turns out to be $Adv_{PSBASME}^{HASH,SC}(t_{m1}, q_{Ry1}, q_{Ry2})=\max_{\mathbb{A}}[Sucss1_{PSBASME}^{HASH,SC}]$, having total execution time t_{m1}, while the Reveal queries as q_{Ry1} and q_{Ry2} maximized on $\mathbb{A}$. We pronounce the contributed protocol as resillient to some attacker $\mathbb{A}$ against recovering the agreed session key (SK) between Sj and Ui, provided $Adv_{PSBASME}^{HASH,SC}(t_{m1}, q_{Ry1}, q_{Ry2}) \leq \propto$ for some adequately small $\propto > 0$. In relation to this experiment, if $\mathbb{A}$ is capable of reversing a one-sided function $h_f(.)$, then solving the hard problem ECDLP, onwards it could comfortably derive the correct session key (SK) computed among the participants, and finally the attacker shall win the game. However, in keeping with definition (2) and (3), this would not be computationally viable to reverse $h_f(.)$ and compute ECDLP, as $Adv_{PSBASME}^{HASH}$ and $Adv_{PSBASME}^{HASH,SC}(t_{m1}) \leq \propto$ for some adequately small $\propto > 0$. In this perspective, the contributed work stands protected in the wake of the strong secure

features of hash function and ECDLP that are evidently too intractable to solve.

7. PERFORMANCE EVALUATION

In performance analysis section, we compare the security properties of proposed scheme with other multi-server authentication schemes. **Table 2** depicts the functionality comparison and threat analysis for various techniques, while this analysis suggests the proposed model as robust against other contemporary MSA-based protocols. According to analysis, the Mishra scheme [21] is prone to forgery attack and besides, it does not conform to perfect forward secrecy. The Chuang scheme [31] lacks mutual authentication, and suffers stolen smart card and forgery attack. Likewise, He et al. scheme [24] does not offer anonymity to the user, and the scheme is prone to forgery and masquerading attacks as well, [35]. The Lu et al. scheme [32] is vulnerable to insider attack and lacks forward secrecy, as above remarked. For comparison of the computation costs, in **Table 3**, we symbolize hash operation with T_H, elliptic curve based point multiplication T_{PM}, asymmetric key encryption/decryption T_{AE} /T_{AD}, and ignoring XOR function for its insignificant cost. The computation time for different crypto-primitives by Kilinc [45] is defined in milliseconds: $T_H \approx 0.0023$ms, $T_{AE} = T_{AD} \approx 3.85$ms, and $T_{PM} \approx 2.226$ms. The comparison in **Table 2** entails Mishra et al., He et al., Chuang et al., Lu et al., and our proposed scheme.

Table 2. Functionality Comparison for different MSA-based schemes

	[21]	[24]	[31]	[32]	Ours
Provides Anonymity	√	×	√	√	√
Provides Mutual Authentication	√	√	×	√	√
Resists Insider Attack	√	√	√	×	√
Resists Forgery Attack	×	×	×	√	√
Resists Offline password guessing attack	√	√	√	√	√
Resists Stolen smart card attack	√	√	×	√	√
Resists Masquerading attack	√	×	√	√	√
Resists Replay attack	√	√	√	√	√
Provides Session key agreement	√	√	√	√	√
Compliant to Perfect forward secrecy	×	√	√	×	√

√ protected against attack or corresponding weakness.
× exposed to some attack or prone to corresponding weakness.

Table 3. Number of operations

		[21]	[24]	[31]	[32]	Ours
Registration		5 T_H	2 T_H	3 T_H	3 T_H	5 T_H
Authentication	User	8 T_H	7 T_H +3T_{PM}	8 T_H	4T_H + 1T_{AE}	5 T_H + 1T_{AE} + 1T_{PM}
	Server	6 T_H	5 T_H + 2 T_{PM}	7 T_H	4T_H + 1T_{AD}	3T_H + 1T_{AD} + 1T_{PM}
	RC	-	9T_H +2 T_{PM}	-	-	-
Total		14 T_H ≈0.0322ms	21 T_H+ 7 T_{PM} ≈15.6303ms	15 T_H ≈0.0345ms	8T_H + 1T_{AD}+ 1T_{AE} ≈7.7184ms	8 T_H +2T_{AE} + 2T_{PM} ≈12.1704ms
Password Modification		3 T_H	2 T_H	3 T_H	3 T_H	3 T_H

The computational cost for [21, 24, 31, 32] is compared against the proposed scheme. Although, the computational cost for [21, 31, 32] is quite low for the use of T_H and T_{AE} based operations, the schemes are susceptible to several attacks, as illustrated in **Table 2**. The He et al.'s scheme [24] is a costly scheme for having $8T_{PM}$ operations in a single iteration of the scheme, despite, the scheme is vulnerable to three attacks as shown above. Although, the proposed protocol comprises two additional T_{PM} operations as compared with [32], yet the proposed protocol is resistant to user impersonation attack and perfect forward secrecy violation. A bit extra cost could be afforded to enhance and improve the security of the protocol. We assume that the communication cost for various crypto-operations, i.e., elliptic curve scalar point is 320-bits, hash-based digest (SHA-1) is 160-bits, user or server's identity is 160-bits, randomly generated number is 160-bits, timestamp is 32-bits and AES encryption is 128-bits [44]. The communication cost for proposed scheme is remarkably lower than other counterparts as depicted in **Table 4**, which ensures the efficiency of the proposed work.

Table 4. Comparison of communication overhead

	No. of Messages	Cost (Bits)
Mishra [21]	(3)	1280
He [24]	(5)	3200
Chuang [31]	(3)	1280
Lu [32]	(3)	768
Ours	(2)	672

Thus, in view of given performance evaluation, we analyze that our proposed model is more resistant to threats than other protocols including Lu et al.'s scheme with a little bit more cost, though necessary. Since, without those extra cryptographic operations like T_{PM}, it might be improbable to bring the perfect forward secrecy in the scheme that also resist against insider attacks.

8. CONCLUSION

The multi-server authenticated key agreement is considered a critical requirement of the crucial requirement of the current internet-based authentication framework. In this work, we reviewed Lu et al.'s scheme which is a multi-server authentication protocol and is found to be vulnerable to many threats. The cryptanalysis revealed that the scheme could be exposed in two ways: malicious insider attack and perfect forward secrecy incompliance. Our proposal counters the identified threats and introduced an improved scheme. This scheme is duly analyzed against the threats using automated tools and formal security procedures, and also evaluated the results with other contemporary MSA-based schemes.

References

[1] Ch, S. A., Sher, M., Ghani, A., Naqvi, H., & Irshad, A., "An efficient signcryption scheme with forward secrecy and public verifiability based on hyper elliptic curve cryptography," *Multimedia Tools and Applications*, 74(5), 1711-1723, 2015. Article (CrossRef Link).

[2] Lamport L., "Password authentication with insecure communication," *ACM Communication*, 24 (11), 770–772, 1981. Article (CrossRef Link).

[3] Sun D, Huai J, Sun J, Li J, Zhang J, Feng Z., "Improvements of Juang's password authenticated key agreement scheme using smart cards," *IEEE Transactions on Industrial Electronics*, 56(6), 2284–2291, 2009. Article (CrossRef Link).

[4] Yu J, Wang G, Mu Y, Gao W., "An efficient generic framework for three-factor authentication with provably secure instantiation," I*EEE Transactions on Information Forensics and Security*, 9(12), 2302–2313, 2014. Article (CrossRef Link).

[5] Lu Y, Li L, Yang Y., "Robust and efficient biometrics based password authentication scheme for telecare medicine information systems using extended chaotic maps," *Journal of Medical Systems*, 2015. Article (CrossRef Link).

[6] Li C, Hwang M., "An efficient biometrics-based remote user authentication scheme using smart cards," *Journal of Network and Computer Applications*, 33(1), 1–5, 2010. Article (CrossRef Link).

[7] He D, Kumar N, Chen J, Lee C, Chilamkurti N, Yeo S., "Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks," *Multimedia Systems*, 21(1), 49–60, 2013. Article (CrossRef Link).

[8] Lu Y, Li L, Peng H, Yang Y., "An enhanced biometric based authentication scheme for telecare medicine information systems using elliptic curve cryptosystem," *Journal of Medical Systems*, 39(3), 1–8, 2015. Article (CrossRef Link).

[9] Wang D, Ma C, Gu D, Cui Z., "Cryptanalysis of two dynamic ID-based remote user authentication schemes for multi-server architecture," *Network and System Security*, 7645, 462–475, 2012. Article (CrossRef Link).

[10] Li X, Ma J, Wang W, Liu C., "A novel smart card and dynamic ID based remote user authentication scheme for multi-server environments," *Mathematical and Computer Modelling*, 58, 85–95, 2013. Article (CrossRef Link).

[11] He D, Zeadally S., "Authentication protocol for an ambient assisted living system," *IEEE Communications Magazine*, 53(1): 71–77, 2015. Article (CrossRef Link).

[12] Tsai J., "Efficient multi-server authentication scheme based on one-way hash function without verification table," *Computers & Security*, 27(3-4): 115–121, 2008. Article (CrossRef Link).

[13] Yang D, Yang B., "A biometric password-based multi-server authentication scheme with smart card," *IEEE International Conference on Computer Design and Applications (ICCDA)*, 5, 554–559, 2010. Article (CrossRef Link).

[14] Irshad, A., Sher, M., Chaudhary, S. A., Naqvi, H., & Farash, M. S., "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, 2016. Article (CrossRef Link).

[15] Chaudhry, S. A., Khan, I., Irshad, A., Ashraf, M. U., Khan, M. K., & Ahmad, H. F., "A provably secure anonymous authentication scheme for Session Initiation Protocol," *Security and Communication Networks*, 2016. Article (CrossRef Link).

[16] Irshad, A., Sher, M., Nawaz, O., Chaudhry, S. A., Khan, I., & Kumari, S., "A secure and provable multi-server authenticated key agreement for TMIS based on Amin et al. scheme," *Multimedia Tools and Applications*, 1-27, 2016. Article (CrossRef Link).

[17] Yoon E, Yoo K., "Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem," *Journal of Supercomputing*, 63(1), 235–255, 2013. Article (CrossRef Link).

[18] He D., "Security flaws in a biometrics-based multi-server authentication with key agreement scheme," *IACR Cryptology*, 1–9, 2011.

[19] Irshad, A., Sher, M., Chaudhry, S. A., Xie, Q., Kumari, S., & Wu, F., "An improved and secure chaotic map based authenticated key agreement in multi-server architecture," *Multimedia Tools and Applications*, 1-38, 2017. Article (CrossRef Link).

[20] Chuang M, Chen M., "An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics," *Expert Systems with Applications*, 41, 1411–1418, 2014. Article (CrossRef Link).

[21] Mishra D, Ashok K. D, Mukhopadhyay S., "A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards," *Expert Systems with Applications*, 41(18), 8129–8143, 2014. Article (CrossRef Link).

[22] Li, X., Niu, J., Kumari, S., Liao, J., & Liang, W., "An enhancement of a smart card authentication scheme for multi-server architecture," *Wireless Personal Communications*, 80(1), 175-192, 2015. Article (CrossRef Link).

[23] Wu, F., Xu, L., Kumari, S., & Li, X., "A novel and provably secure biometrics-based three-factor remote authentication scheme for mobile client–server networks," *Computers & Electrical Engineering*, 45, 274-285, 2015. Article (CrossRef Link).

[24] He D, Wang D., "Robust biometrics-based authentication scheme for multiserver environment," *IEEE Systems Journal*, 9(3), 816-823, 2015. Article (CrossRef Link).

[25] Burrow M, Abadi M, Needham R., "A logic of authentication," *ACM Transactions on Computer Systems*, 8(1), 18–36, 1990. Article (CrossRef Link).

[26] Li, X., Niu, J., Kumari, S., Khan, M. K., Liao, J., & Liang, W., "Design and analysis of a chaotic maps-based three-party authenticated key agreement protocol," *Nonlinear Dynamics*, 80(3), 1209-1220, 2015. Article (CrossRef Link).

[27] Jiang, Q., Ma, J., Lu, X., & Tian, Y., "An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks," *Peer-to-Peer Networking and Applications*, 8(6), 1070-1081, 2015. Article (CrossRef Link).

[28] Jiang, Q., Ma, J., Li, G., & Li, X., "Improvement of robust smart-card-based password authentication scheme," *International Journal of Communication Systems*, 28(2), 383-393, 2015. Article (CrossRef Link).

[29] Jiang, Q., Khan, M. K., Lu, X., Ma, J., & He, D., "A privacy preserving three-factor authentication protocol for e-Health clouds," *The Journal of Supercomputing*, 72(10), 3826-3849, 2016. Article (CrossRef Link).

[30] Li, X., Niu, J., Kumari, S., Islam, S. H., Wu, F., Khan, M. K., & Das, A. K., "A novel chaotic maps-based user authentication and key agreement protocol for multi-server environments with provable security. Wire. Pers. Comm., 89(2), 569-597, 2016. Article (CrossRef Link).

[32] Lu, Y., Li, L., Peng, H., & Yang, Y., "A biometrics and smart cards-based authentication scheme for multi-server environments," *Security and Communication Networks*, 8(17), 3219-3228, 2015. Article (CrossRef Link).

[33] Dodis Y, Reyzin L, Smith A., "Fuzzy extractors: how to generate strong keys from biometrics and other noisy data" *Advances in Cryptology—EUROCRYPT*, 3027: 523–540, 2004. Article (CrossRef Link).

[34] Dodis Y, Kanukurthi B, Katz J, Reyzin L, Smith A., "Robust Fuzzy Extractors and Authenticated Key Agreement From Close Secrets," *IEEE Transactions on Information Theory*, 58(9), 6207–6222, 2012. Article (CrossRef Link).

[35] Odelu, V., Ashok, K. D., and Adrijit G. "A secure biometrics-based multi-server authentication protocol using smart cards," *IEEE Transactions on Information Forensics and Security*, 10(9), 1953-1966, 2015. Article (CrossRef Link).

[36] Koblitz, N., Elliptic Curve Cryptosystems. Math. Of Comp., Vol. 48, 203-209, 1987.

[37] Ashok, K. D., Odelu, V., and Adrijit G., "A Secure and Robust User Authenticated Key Agreement Scheme for Hierarchical Multi-medical Server Environment in TMIS," *Journal of Medical Systems*, 39(9), 1-24, 2015. Article (CrossRef Link).

[38] Ashok, K. D., "A secure user anonymity-preserving three-factor remote user authentication scheme for the telecare medicine information systems," *Journal of medical systems*, 39(3), 1-20, 2015. Article (CrossRef Link).

[39] Chatterjee, S., and Das, A.K., "An effective ECC-based user access control scheme with attribute-based encryption for wireless sensor networks," *Secur. Commun. Netw.*, 8(9), 1752–1771, 2015. Article (CrossRef Link).

[40] Das, A.K., Paul, N.R., Tripathy, L., "Cryptanalysis and improvement of an access control in user hierarchy based on elliptic curve cryptosystem," *Information Sciences*, 209(C), 80–92, 2012. Article (CrossRef Link).

[41] Chaudhry SA, Farash MS, Naqvi H, Islam SH, Shon T, Sher M, "A robust and efficient privacy aware handover authentication scheme for wireless networks," *Wireless Personal Communication*, 2015. Article (CrossRef Link).

[42] Xie, Q., Hu, B., Dong, N., & Wong, D. S., "Anonymous three-party password-authenticated key exchange scheme for telecare medical information systems," *PLoS One*, 9(7), e102,747, 2014. Article (CrossRef Link).

[43] Odelu, V., Das, A. K., & Goswami, A., "SEAP: secure and efficient authentication protocol for NFC applications using pseudonyms," *IEEE Transactions on Consumer Electronics*, 62(1), 30-38, 2016. Article (CrossRef Link).

[44] Amin, R., Islam, S. H., Biswas, G. P., Khan, M. K., & Kumar, N., "An efficient and practical smart card based anonymity preserving user authentication scheme for TMIS using elliptic curve cryptography," *Journal of medical systems*, 39(11), 1-18, 2015. Article (CrossRef Link).

[45] Kilinc, H. H., & Yanik, T., "A survey of SIP authentication and key agreement schemes," *Communications Surveys & Tutorials*, IEEE, 16(2), 1005-1023, 2014. Article (CrossRef Link).

Azeem Irshad received Master's degree from Arid Agriculture University, Rawalpindi, Pakistan. Currently, he is persuing his PhD in security for multi-server architectures, from International Islamic University, Islamabad, Pakistan. His research interests include strengthening of authenticated key agreements in SIP multimedia, IoT, WBAN, TMIS, WSN, Ad hoc Networks, e-health clouds and multi-server architectures.

Muhammad Sher is a Professor having more than 120 scientific publications. He is chairman of the Department of Computer Science & Software Engineering, International Islamic University. He is also Dean of the Faculty of Basic & Applied Sciences. He did his Ph.D. Computer Science from TU Berlin, Germany and M. Sc. From Quaid-e-Azam University, Islamabad. His research interests include Next Generation Networks and Network Security.

Bander A Alzahrani is an assistance professor at King Abdulaziz University, Saudi Arabia. He completed his M.Sc. in Computer Security (2010), and his Ph.D. in Computer Science (2015), both from University of Essex, United Kingdom. His research interests include Network security, Information centric networks, Bloom filter data structure and its applications, secure content routing, IoT.

Aiid Albeshri received M.S. and Ph.D. degrees in Information Technology from Queensland University of Technology, Brisbane, Australia in 2007 and 2013 respectively. He has been an assistant professor at the Computer Science Department of the King Abdulaziz University, Jeddah, Saudi Arabia since 2013. His current research focuses on Security and Trust in Cloud computing and big data.

Shehzad Ashraf Chaudhry received distinction in his Masters and PhD from International Islamic University Islamabad, Pakistan in 2009 and 2016 respectively. He was awarded Gold Medal for achieving 4.0/4.0 CGPA in his Masters. Currently, he is working as an Assistant Professor at the Department of Computer Science & Software Engineering, International Islamic University, Islamabad. He authored more than 60 scientific publications appeared in different international journals and proceedings including 42 in SCI/E journal. His research interests include Lightweight Cryptography, Elliptic/Hyper Elliptic Curve Cryptography, Multimedia Security, E- Payment systems, MANETs, SIP authentication, Smart Grid Security, IP Multimedia sub-system and Next Generation Networks.

Dr. Saru Kumari is currently an Assistant Professor with the Department of Mathematics, C.C.S. University, Meerut, U.P, India. She received Ph.D. degree in Mathematics in 2012 from C.C.S. University, Meerut, Uttar Pradesh, India. She has published 85 papers in international journals and conferences including 69 research publications in SCI indexed journals. Her current research interests are on cryptology and information security.