An efficient and secure design of multi-server authenticated key agreement protocol

Azeem Irshad, Husnain Naqvi, Shehzad Ashraf Chaudhry, Shouket Raheem, Saru Kumari, Ambrina Kanwal & Muhammad Usman

Abstract

Multi-server authentication, being a crucial component of remote communication, provides the ease of one-time registration to users from a centralized registration authority. Therefore, the users could avail the offered services after getting authenticated of any service provider using the same registration credentials. In recent years, many multi-server authentication protocols have been demonstrated. Nonetheless, the existing schemes do not meet the security and efficiency requirements of the time. Recently, Chuang et al. presented a multi-server biometric authentication protocol which was later crypt-analysed and improved by Lin et al. with the identification of few attacks. Later, we discover that Lin et al.'s protocol is still prone to replay attack, privileged insider attack, trace attack, de-synchronization attack and key-compromise impersonation attacks. In this study, we present a multi-server authentication protocol which is not only comparable with Lin et al.'s scheme but also efficient than other state-of-the-art multi-server protocols. The security properties of our scheme are proved using formal analysis and evaluated with automated verification tool based on ProVerif

1. INTRODUCTION

Multi-server authentication enables cost efficient authentication that leads to the quick accessibility of remote online services for users in the public environment. The multi-server authentication is synonymous to overhead efficiency for it decreases the computational or communication delay when a subscriber seeks to avail multiple services from various service providers in a network. In multi-server authentication, the user needs a single password and registration phase to avail services from several online servers. The remote communications amid authentication of smart gadgets often encompass the multi-server authentication paradigm, which signifies towards the robustness and efficiency of these protocols. The multi-server setup comprises of three participating entities, i.e., user (subscriber), server (service dispenser) and Registration Centre (RC). The user gets the one-time registration performed through RC and thereafter, it could be dispensed with the offered services after having performed the mutual authentication phase with server.

In the previous decade many multi-server authentication techniques can be witnessed. Nevertheless, there is always need of more efficient and secure protocols in the wake of increasing mobile users and wireless gadgets. Earlier in 1981, Lamport [1] demonstrated a protocol for remote authentication using an open and insecure channel. Nonetheless, the requirement of password or verifiers' maintenance in a database on the server is taken as a limitation for malicious tendencies of an adversary to exploit it. Following the Lamport work, different authentication schemes were proposed [3-5], however, these were based on single server architecture, which do not cater to the requirements of multi-server architecture paradigm. Now in literature, we can see many multi-server authentication schemes [6-11] related to smart card bearing biometric and anonymity properties. In this regard, Liao and Wang [6] put forwarded a dynamic-ID-based authentication scheme. This scheme was investigated by Hsiang and Shih [8], who revealed about the scheme's vulnerability as privileged insider attack, impersonation attack, and not supporting mutual authentication. Then, some additional schemes were proposed by a few scholars [13-16]. To surmount the limitations of those protocols, biometric authentication schemes were demonstrated as a three-factor authentication [12, 17-18]. Nonetheless, these schemes are prone to weaknesses such as lack of efficiency and anonymity. Subsequently, Chuang and Chen [19] introduced an anonymous multi-server authentication scheme. Unfortunately, Lin et al, [22] initiated masquerading attacks, and the protocol could not protect its session key on the exposure of private secrets. Then, Lin et al, presented an enhanced scheme considering the above flaws. Unfortunately, Lin et al. protocol is prone to replay attack, privileged insider attack, trace attack, de-synchronization attack and key-compromise impersonation attack. The current study takes a review of Lin et al. protocol [22] along with demonstration of the scheme's cryptanalysis. Ultimately, we propose an enhanced biometric multi-server authentication scheme that eliminates the registration centre from mutual authentication between user and server, leading to communicational efficiency. Besides, our scheme is complemented with formal analysis and automated tool-based verification analysis that demonstrate the resilience of the contributed protocol in comparison with state-of-the-art protocol.

As per the organization of this paper, the Section 2 describes the preliminaries related to current work. The Section 3 illustrates the working of Lin et al.'s protocol. The Section 4 presents the contributed model of our scheme. The section 5 and 6 portray informal security discussion and performance evaluation, respectively. Finally, the last section summarizes the findings.

2. PRELIMINARIES

In this section, we have described the elliptic curve and hash function, which served as cryptographic building blocks for designing a protocol in this study.

2.1 Elliptic Curve Cryptography

The Elliptic Curve Cryptography affords effective cryptographic techniques in comparison with traditional ones like DSA, DH and RSA. Such crypto-primitives assure less key-size as compared to key sizes in conventional cryptography. A non-super singular elliptic curve E, can be defined over a finite field F_p, as $E_p(\rho, \partial)$: $y^2 = x^3 + \rho x + \partial \pmod{p}$ and $4 \rho^3 + 27 \partial^3 \neq 0 \pmod{p}$, while $\rho, \partial \epsilon F_p$ and p serves as a large prime number. In this context, we define some intractable problems, providing the basis of security for the existing work as shown below:

1. Referring to Elliptic Curve-based Computational Diffie–Hellman problem (EC_CDHP), it is computationally intractable to construct abP, given G's generator P, aP, bP. (1)
2. Referring to Elliptic Curve-based Discrete Logarithm Problem (EC_DLP), it is computationally intractable to derive a from a point $Q=aP$ on an elliptic curve, where P serves as a G's generator. (2)

2.2 Hash digest function

We assume a one-sided hash function as h:{0,1}* → Z^*_p that takes as random input a variable-sized string τ, and outputs y, a string of fixed length, i.e., $y=h(\tau)$, which is termed as the hash value. Any deliberate or accidental change in τ is instantly reflected in y. A secure hash-function underpins the following features:

1. It is intractable to alter the message ξ without, the $h(\xi)$ being modified.
2. It is improbable to construct the string ξ, which generates the hash i.e., $h(\xi)$, as pre-image resistance.
3. It is computationally difficult to find an input ξ_2, given ξ_1, where $\xi_1 \neq \xi_2$ and $h(\xi_1)=h(\xi_2)$ simultaneously.
4. Lastly, it is difficult in polynomial time to locate any two strings ξ_1 and ξ_2, provided the equality $h(\xi_1)=h(\xi_2)$ also holds, termed as a strong collision resistance.

3. WORKING AND CRYPTANALYSIS OF LIN ET Al. PROTOCOL

The design of Lin et al.'s protocol is illustrated as under:

3.1 Revisiting Lin et al.'s scheme

The Lin et al.'s protocol [22] encompasses the registration phase, login and mutual authentication phase as exhibited in Figure 1. We used some symbols in this study as given in Table I.

Table I. Symbolic Representations

Symbols	Meanings
Ui, Sj, RC	ith user, jth server, and Registration centre
ID_i, SID_j	Identities of Ui and Sj
PW_i, BIO_i:	Ui's password, Ui's biometric identity
PID_j:	The shared value between RC and S_j
h(.):	a secure hash digest function
H(.):	Bio-hashing function
Ti/Tj:	Timestamps

x:	RC’s master key
SK_{ij}:	Mutually agreed Session Key (Ui and S_j)
$||/\oplus$:	concatenation and XOR functions

3.1.1 Server Registration procedure

This scheme involves a trusted RC and ψ number of reliable servers S_j, such that j=1…..ψ. The S_j registers itself by sending SID_j to RC. The latter calculates $r_j=h(SID_j || x)$ and forwards to Sj to initialize the service providing server’s setup.

3.1.2 User Registration Stage

In this stage, Ui gets registered from the trusted RC. Afterwards, the former may receive the stipulated services of service providing servers (Sj). To register itself, the user Ui follows the under-mentioned steps:

1. The user submits $\{ID_i, h(PW_i|| BIO_i), h(ID_i|| BIO_i)\}$ by computing $h(PW_i|| BIO_i)$ and $h(ID_i|| BIO_i)\}$ to RC, using a confidential channel.
2. RC gets and calculates $Ai=h(IDi ||x)$, $Bij= h(Ai || r_j))$, $Dij= E_{h(IDi || BIO_i)} [Bij]$, $Ei= h(Ai ||h(PW_i || BIO_i)||IDi)$. It, then, stores the parameters in *SC {Dij, Ai, Ei}* and sends to Ui.

3.1.3 Mutual Authentication Stage

1. In login stage, Ui inserts its smart card (SC) to follow the login steps and get authenticated access of Sj. After inserting SC, Ui give the parameters IDi, PW_i and BIO_i as input in smart card and calculates $Ei^*= h(Ai ||h(PW_i || BIO_i)||IDi)$ and finds the validity of equation $Ei^* ?= Ei$. On finding it as true, it further generates nonce *m* and timestamp*Ti*. Then, it computes $M=m.P$, decrypts *Dij* into *Bij* as, $\{Bij \leftarrow D_{h(IDi || BIO_i)} [Dij]\}$. Further, it computes $Hi=Ai\oplus h(M||Ti || SID_j)$ and $Zi = E_{Bij} [h(PW_i || BIO_{i,}), M, Ti]$. Finally it submits the message *{Ai, Hi, Zi}* to *Sj* for authentication. It is worth mentioning here that the timestamp is used in Lin et al.’s protocol for time synchronization to avoid the replay attacks.
2. Upon receiving the parameters in authentication phase, the server computes $Bij= h(Ai || r_j)$*, and* $\{h(PW_i|| BIO_{i,}), M, Ti\}\leftarrow D_{Bij} [Zi]$ by decrypting *Zi*. Then, Sj Checks the timestamp freshness by generating *Tj* and verifying $Tj-Ti< \Delta T$. If true, then computes $Hi^*=Ai\oplus h(M||Ti || SID_j)$, and checks again $Hi^* ?= Hi$. Now on positive verification, it generates a random integer *n*, and calculates $Vi=h(SID_j\oplus h(PW_i || BIO_i))$, $N=n.P$, $Ki=E_{Bij} [Vi, N, SID_j]$, and $SKij=n.M$. Now it submits the message *{Ki}* to user for further verification.
3.

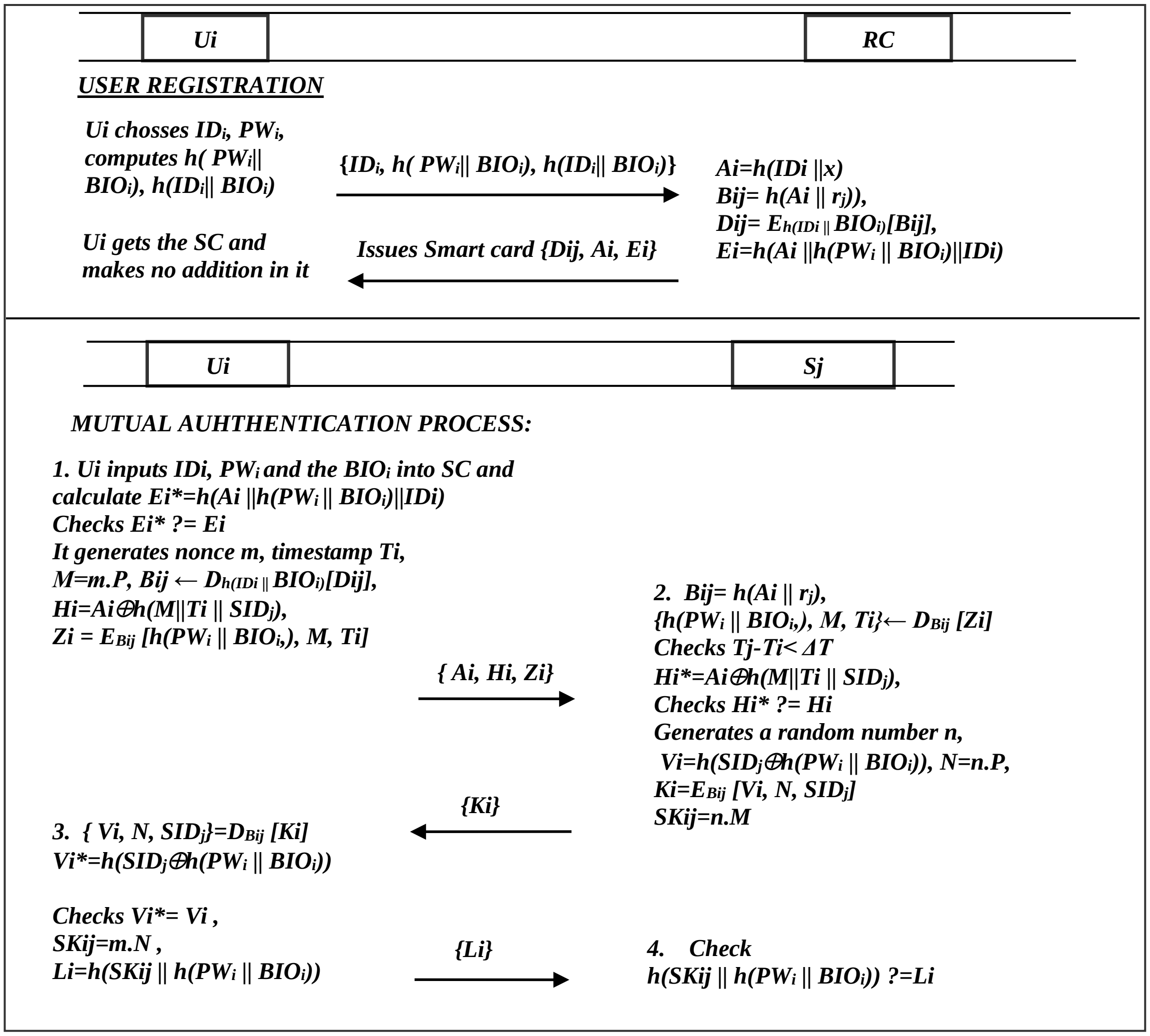

Ui
RC
USER REGISTRATION
Ui chosses ID_i, PW_i, computes $h(PW_i || BIO_i)$, $h(ID_i || BIO_i)$
$\{ID_i, h(PW_i || BIO_i), h(ID_i || BIO_i)\}$
$Ai=h(IDi || x)$
$Bij= h(Ai || r_j))$,
$Dij= E_{h(IDi || BIO_i)}[Bij]$,
$Ei=h(Ai || h(PW_i || BIO_i) || IDi)$
Ui gets the SC and makes no addition in it
Issues Smart card {Dij, Ai, Ei}
Ui
Sj
MUTUAL AUHTHENTICATION PROCESS:
1. Ui inputs IDi, PW_i and the BIO_i into SC and calculate $Ei^*=h(Ai || h(PW_i || BIO_i) || IDi)$
Checks $Ei^* ?= Ei$
It generates nonce m, timestamp Ti,
$M=m.P$, $Bij \leftarrow D_{h(IDi || BIO_i)}[Dij]$,
$Hi=Ai \oplus h(M || Ti || SID_j)$,
$Zi = E_{Bij} [h(PW_i || BIO_i,), M, Ti]$
{ Ai, Hi, Zi}
2. $Bij= h(Ai || r_j)$,
$\{h(PW_i || BIO_i,), M, Ti\} \leftarrow D_{Bij} [Zi]$
Checks $Tj-Ti< \Delta T$
$Hi^*=Ai \oplus h(M || Ti || SID_j)$,
Checks $Hi^* ?= Hi$
Generates a random number n,
$Vi=h(SID_j \oplus h(PW_i || BIO_i))$, $N=n.P$,
$Ki=E_{Bij} [Vi, N, SID_j]$
$SKij=n.M$
{Ki}
3. $\{ Vi, N, SID_j\}=D_{Bij} [Ki]$
$Vi^*=h(SID_j \oplus h(PW_i || BIO_i))$
Checks $Vi^*= Vi$,
$SKij=m.N$,
$Li=h(SKij || h(PW_i || BIO_i))$
{Li}
4. Check
$h(SKij || h(PW_i || BIO_i)) ?=Li$

Figure 1. Lin et al.'s protocol Registration and Mutual Authentication phases

4. Ui receives the message and decrypts *Ki* as $\{ Vi, N, SID_j\}=D_{Bij}[Ki]$.Now it computes $Vi^*=h(SID_j \oplus h(PW_i || BIO_i))$ and checks $Vi^*= Vi$. If true, then computes $SK_{ij}=m.N$, and $Li=h(SK_{ij} || h(PW_i || BIO_i))$. Ultimately, it submits the message *{Li}* to *Sj* for acknowledgement and verification.
5. *Sj* gets the parameter *{Li}* and computes $h(SK_{ij} || h(PW_i || BIO_i))$. Then, it compares $h(SK_{ij} || h(PW_i || BIO_i)) ?=Li$. If this inequality holds true, it treats the user as a valid user, on the other hand it terminates the session.

3.2 Drawbacks in Lin et al. scheme.

This section covers the Drawbacks of Lin et al. protocol, which is discovered as prone to replay attack, server spoofing attack and trace attacks. The limitations of Lin et al.'s protocol are narrated as under.

3.2.1 Replay Attack

The replay attack could be launched successfully by resending the message {*Ki*} by an adversary, and impersonating a legal server Sj to deceive a legitimate user Ui. Whenever, the same user Ui tries to establish a session with a specific Sj, the adversary may launch this attack by replaying *Ki* towards Ui. The parameter *Ki* does not contain any timestamp or any factor that Ui could authenticate Sj. Although an adversary cannot construct this Ki by itself, since it requires encryption of *Vi*, *N*, SID_j parameters using a *Bij* key, as $Ki=E_{Bij} [Vi, N, SID_j]$.This attack could be thwarted by bringing a timestamp mechanism from Sj entity or making include in this message any parameter value from the current session i.e $M=n.P$. This timestamp or the value *M*, if included, might debar an adversary to replay this message.

3.2.2 Privileged insider attack

This attack can be initiated by a privileged insider $\mathfrak{T}$, for instance, system administrator, or privileged insider of RC, who might get access to the parameters sent for registration purpose. In Lin et al. scheme, if the insider $\mathfrak{T}$ is able to approach any user's registration parameters $\{ID_i, h(PW_i || BIO_i), h(ID_i || BIO_i)\}$, then it may impersonate as a legal user Ui by putting a fake authentication request on behalf of legitimate user in the following manner.

1. Assuming, the malicious insider $\mathfrak{T}$ possesses the parameter *Ai* by intercepting the Ui's messages on a public channel, and *Dij* by extracting smart card contents using differential power analysis.
2. Next, $\mathfrak{T}$ computes $M_A = m.P$ by generating a random integer *m*.
3. Then, it further computes $Hi=Ai \oplus h(M||Ti || SID_j)$ by generating a fresh timestamp *Ti*.
4. Next, it decrypts *Dij*, i.e. $Bij \leftarrow D_{h(IDi || BIO_i)}[Dij]$, and computes $Zi = E_{Bij} [h(PW_i || BIO_i), M_A, Ti]$.
5. Then, it submits the authentication message $\{ Ai, Hi, Zi\}$ to server, that would be duly verified by *Sj*, however fake.
6. In this way, a successful insider attack may launched by a malicious insider.

3.2.3 Trace Attack

In this attack, an attacker may recognize or trace the location for any legal participant through finding the similar parameters in two different sessions. In Lin et al, the *Ai* parameter in a login request always remains the same for all sessions. Hence, an adversary, comfortably, analyze the traceability for a user Ui.

3.2.4 De-synchronization Attack

This is not an outsider attack. A user Ui might get trapped during login phase of the protocol, while interacting with smart card, and inputting its *IDi*, PW_i and BIO_i parameters. The smart card may refuse a valid user for non-matching of BIO_i parameter as input by the user. This might be due to minor difference in the capturing of BIO_i input through sensor, so the pre-stored BIO_i value may be a little different from the captured BIO_i input. In this case, the smart card may refuse to login a legitimate user. Consequently, a user will not be able to proceed with the mutual authentication phase with server.

3.2.5 Key-Compromise Impersonation Attack (KCI)

This is an attack that can be initiated by an adversary towards entity $\aleph_1$ through impersonating another legal entity $\aleph_2$, in case some key or secret of $\aleph_1$ is accessed by the attacker. In Lin et al. scheme, a malicious privileged insider, having access to $h(PW_i || BIO_i)$ and $h(ID_i || BIO_i)$ may easily initiate an attack towards Ui by impersonating as a server by adopting the following steps:

1. The adversary recovers *Dij* from smart card employing differential power analysis and extracts *Bij* by decrypting *Dij* through $h(IDi || BIO_i)$ i.e. $Bij \leftarrow D_{h(IDi || BIO_i)}[Dij]$.
2. Next, it generates a random number n_a and calculates $N_a = n_a.P$, $Vi=h(SID_j \oplus h(PW_i || BIO_i))$ and $Ki=E_{Bij} [Vi, N_a, SID_j]$.
3. Finally, it sends *Ki* to a user Ui for impersonating as a server.
4. The Ui confirms the adversary as a legal user, simply due to encrypted forged message out of *Bij*.

In this way, a successfull KCI attack could be launched against user, in Lin et al. scheme.

4. PROPOSED MODEL

To date, the previous multi-server authentication protocols seem to go through many pitfalls as far as security and efficiency is concerned. We present an improved Lin et al.'s scheme that bears the optimized and comparable security features including biometric attributes, in relation to existing protocols. Our proposed protocol encompasses four stages, i.e., initialization stage, user registration stage, mutual authentication stage, and password modification stage as shown below:

4.1 Initialization Phase

The contributed scheme involves a trusted RC and ψ trusted servers Sj, while j=(1….. ψ). The server Sj performs registration through RC before Ui's registration process, over a confidential channel. The Sj sends its identity SID_j towards RC. Subsequently, RC will compute $PID_j=h(SID_j || x)$ and submit towards Sj, which remains the shared secret between RC and S_j, while *x* is the master key of RC. Next, RC generates *y* as its private key, and computes

$k=h(y)$ and distribute k to each server. RC also computes its public key $Q=kP$ and publishes it publicly. The RC chooses Elliptic Curve $E_p(a,b)$, while P being the generator of further points with large primer number order, and it is a hard discrete logarithm problem in the cyclic subgroup G.

4.2 The Registration stage

In this stage, Ui gets registered from RC on a secure channel, and then, it can access all Sj servers for mutual authentication phase, thereafter. Ui performs the under-mentioned steps with RC:

1. Ui selects IDi, PW_i, r_1, r_2, and imprints BIO$_i$ on the sensor. It computes $R_1=h(PW_i|| H(BIO_i))$, $R_2 =h(ID_i|| H(BIO_i)) \oplus r_1$ and $R_3=h(h(ID_i)|| H(BIO_i)) \oplus r_2$. Then it sends {$ID_i, R_1 , R_2 , R_3$ } to RC for registration.
2. RC receives and computes $Ai=h(IDi ||x)$, $B_{ij}= h(Ai || PID_j))$, $Dij= R_2 \oplus B_{ij}$, $Ei= h(Ai || R_1 ||IDi)$ and $Fi = R_3 \oplus Ai$. It, then, stores the parameters in $SC \{D_{ij}, Ei, Fi\}$ and sends to Ui.
3. Ui receives the SC, computes $D_{ij}' = D_{ij}\oplus r_1$, $Fi'=Fi\oplus r_2$, and replaces D_{ij} with D_{ij}' and Fi with Fi' in smart card. The smart card now contains $\{D_{ij}', Ei, Fi'\}$.

5.3 Mutual Authentication Stage

1. In login stage, Ui employs its smart card for getting the verified access to services of Sj. For this objective, Ui inputs its IDi, PW_i and then imprints BIO_i in biometric scanner device. Next, the smart card calculates $R_1=h(PW_i|| H(BIO_i))$, $Ai = h(h(ID_i)|| H(BIO_i)) \oplus Fi'$, and $Ei*=h(Ai ||R_1||IDi)$. Then, it checks the validity $Ei* ?= Ei$. If true, then it will generate random number m and compute $M=m.P$, $W=mQ$, $X=Ai\oplus W$, $B_{ij} = h(ID_i|| H(BIO_i)) \oplus D_{ij}'$ and $Hi= h(M|| W ||Ai ||B_{ij} || SID_j)$. Ultimately, it submits $\{M, X, Hi\}$ to Sj using a public channel.
2. In authentication phase, the Sj receives parameters and computes $W'=kM$, $Ai = X\oplus W'$, $B_{ij}= h(Ai || PID_j)$, $Hi*= h(M|| W' ||Ai ||B_{ij} || SID_j)$ and checks the equality for $Hi* ?= Hi$. If this equality matches, it generates a random integer n and calculates $SKj=n.M$, $N=n.P$ and $Vi=h(SKj || Ai || Bij ||W ||M)$. Now it submits the contents $\{N, Vi\}$ to user for further verification on public channel.
3. Ui receives the message and computes $SKi=m.N$ and $Vi*=h(SKi || Ai || B_{ij} || W || M))$. Then, it verifies the equation $Vi* ?= Vi$. If the equality matches, it further calculates $Ji= h(SKi || Ai || Bij || M ||N || SID_j)$ and submits the message $\{Ji\}$ to Sj for acknowledgement and verification.
4. Sj receives the message Ji and computes $Ji*= h(SKj || Ai || B_{ij} || M ||N || SID_j)$. Then, it compares the equation $Ji* ?= Ji$. If this equation holds true, then treats the user as the valid user, otherwise, terminates the session.

4.4 Password Updating Procedure

Ui gets its password updated by adopting this phase, into a new password (PW_i^{new}) without consulting RC. The steps for the password modification are stated below:

1. First, the user puts its SC into the scanner and inputs the corresponding identity ($IDi*$), password (PW_i*), and onwards the biometric identity (BIO_i*) in scanner device. Then, SC calculates $R_1=h(PW_i|| H(BIO_i))$, $Ai = h(h(ID_i)|| H(BIO_i)) \oplus Fi'$, and $Ei*=h(Ai ||R_1||IDi)$. Next, it checks the validity for $Ei* ?= Ei$. If it does not find a match, it aborts the modification phase.
2. Otherwise, the smart card invokes the user to enter a new password PW_i^{new} and computes $Ai = h(h(ID_i)||H(BIO_i))\oplus Fi'$, $R_1^{new}=h(PW_i^{new}||H(BIO_i))$ and $Ei^{new}=h(Ai || R_1^{new} ||IDi))$.
3. Next, the SC stores Ei^{new} into the SC to replace Ei.

5. SECURITY ANALYSIS

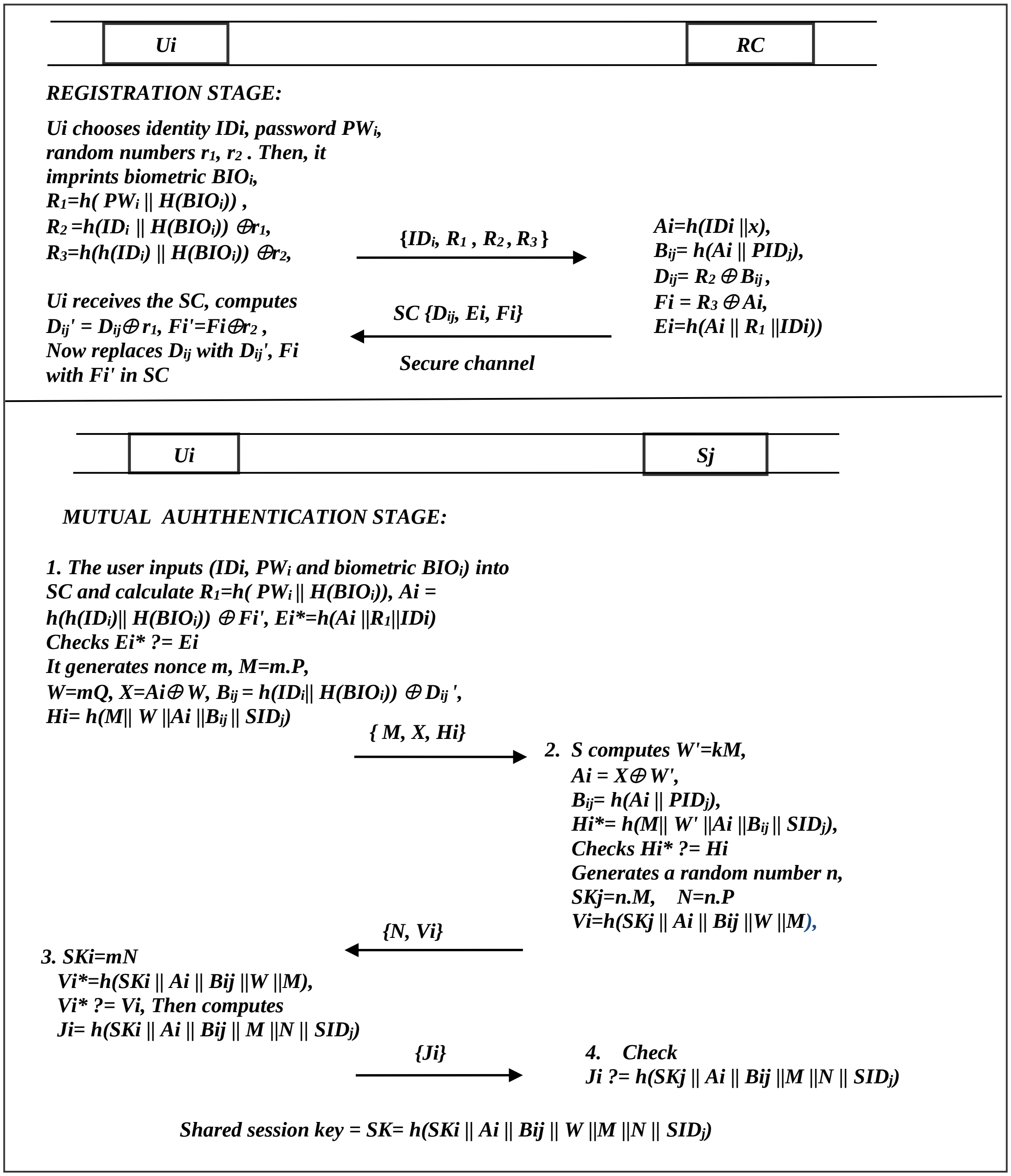
This section covers the informal security discussion, automated security verification, and formal logic analysis [24-31] of the proposed protocol and has been presented as under:

5.1 Security discussion

This sub-section entails the informal discussion about the security of the contributed scheme.

5.1.1 Replay Attacks

These attacks could be initiated while the adversary replays the intercepted contents to betray or masquerade any legitimate session member [32-36]. An attacker, having the open messages $\{M, X, Hi, N, Vi, Ji\}$ may attempt to replay those contents on both sides to forge the legitimate participants. Nonetheless, Ui validates Sj and nullifies the chances of any replay attack by computing and verifying the equation $Vi^* ?=h(SKi \,||\, Ai \,||\, Bij \,||W \,||M)$. The calculation of Vi comprises the parameter M, which must be concatenated with other parameters to foil the replay attack. Likewise, Sj may thwart the replay attack by computing and verifying the equation $Ji ?= h(SKj \,||\, Ai \,||\, Bij \,||M \,||N \,||\, SID_j)$ in the third run of the protocol. The presence of M and N parameters in the computation of Ji make certain that the replay attack is defeated. Therefore, the contributed protocol could thwart a replay attack.



REGISTRATION STAGE:

Ui chooses identity IDi, password PW_i, random numbers r_1, r_2. Then, it imprints biometric BIO_i,
$R_1=h(PW_i \,||\, H(BIO_i))$,
$R_2=h(ID_i \,||\, H(BIO_i)) \oplus r_1$,
$R_3=h(h(ID_i) \,||\, H(BIO_i)) \oplus r_2$,

$\{ID_i, R_1, R_2, R_3\}$

$Ai=h(IDi \,||x)$,
$B_{ij}= h(Ai \,||\, PID_j)$,
$D_{ij}= R_2 \oplus B_{ij}$,
$Fi = R_3 \oplus Ai$,
$Ei=h(Ai \,||\, R_1 \,||IDi))$

SC $\{D_{ij}, Ei, Fi\}$

Secure channel

Ui receives the SC, computes $D_{ij}' = D_{ij} \oplus r_1$, $Fi'=Fi \oplus r_2$, Now replaces D_{ij} with D_{ij}', Fi with Fi' in SC

MUTUAL AUHTHENTICATION STAGE:

1. The user inputs (IDi, PW_i and biometric BIO_i) into SC and calculate $R_1=h(PW_i \,||\, H(BIO_i))$, $Ai = h(h(ID_i) || H(BIO_i)) \oplus Fi'$, $Ei^*=h(Ai \,||R_1||IDi)$
   Checks $Ei^* ?= Ei$
   It generates nonce m, $M=m.P$,
   $W=mQ$, $X=Ai \oplus W$, $B_{ij} = h(ID_i|| H(BIO_i)) \oplus D_{ij}'$,
   $Hi= h(M|| W \,||Ai \,||B_{ij} \,||\, SID_j)$

$\{M, X, Hi\}$

2. S computes $W'=kM$,
   $Ai = X \oplus W'$,
   $B_{ij}= h(Ai \,||\, PID_j)$,
   $Hi^*= h(M|| W' \,||Ai \,||B_{ij} \,||\, SID_j)$,
   Checks $Hi^* ?= Hi$
   Generates a random number n,
   $SKj=n.M$, $N=n.P$
   $Vi=h(SKj \,||\, Ai \,||\, Bij \,||W \,||M)$,

$\{N, Vi\}$

3. $SKi=mN$
   $Vi^*=h(SKi \,||\, Ai \,||\, Bij \,||W \,||M)$,
   $Vi^* ?= Vi$, Then computes
   $Ji= h(SKi \,||\, Ai \,||\, Bij \,||\, M \,||N \,||\, SID_j)$

$\{Ji\}$

4. Check
   $Ji ?= h(SKj \,||\, Ai \,||\, Bij \,||M \,||N \,||\, SID_j)$

Shared session key = $SK= h(SKi \,||\, Ai \,||\, Bij \,||\, W \,||M \,||N \,||\, SID_j)$

Figure 2. Proposed Authentication Protocol

5.1.2 Modification Attacks

The modification attacks could be initiated in case; the attacker Ă changes and restructures the message parameters in an unlawful manner to betray any legal subscriber [37-40].

If any adversary tries to modify the messages {*M, X, Hi, N, Vi, Ji* }, the server will not be able to verify the equality $Hi\ ?{*}{=}\ h(M||\ W'\ ||Ai\ ||B_{ij}\ ||\ SID_{j})$,with an updated *M*. Since, *Ai* and B_{ij} cannot be produced by an adversary, so any modification in the sent parameters will be caught instantly on the other side. Similarly, an adversary may also try to forge another participant by modifying the exchanged messages {*N, Vi, Ji*}. If it is so, the other side will be in a sound position to detect any such modification on the basis of *Vi* ?= h(SKi || Ai || Bij ||W ||M),* and $Ji\ ?{=}\ h(SKi\ ||\ Ai\ ||\ Bij\ ||M\ ||N\ ||\ SID_{j})$ equation checks.

5.1.3 *Offline-password guessing attack*

This attack could be launched when an adversary tries to get either a Ui's password after intercepting the public messages {*M, X, Hi, N, Vi, Ji*} or stealing smart card parameters $\{D_{ij}',\ Ei,\ Fi'\}$. In all of these parameters, only *Ei* has been constructed with a combination of password PW_i, i.e. $Ei{=}h(Ai\ ||\ h(PW_i||\ H(BIO_i))\ ||IDi))$. An adversary may not be able to guess a password from *Ei* until it recovers the $H(BIO_i)$ parameter. Therefore, our protocol is immune to offline-password guessing attack.

5.1.4 *Stolen Verifier Attacks*

The attacker could steal valuable data which may be stored on the end of server; since the server might be maintaining the user-based verifier's database. Then, the adversary may exploit the contents to forge and impersonate the legitimate users, which is known as stolen verifier attack.

The contributed protocol does not manage any verifiers' database at the end of server or registration centre, which is a prerequisite for the adversary to initiate a stolen verifier attack.

5.1.5 *Stolen smart card attack*

In offline-dictionary attack [41-46], an adversary steals the user's smart card and attempts to utilize the extracted contents in initiating brute force attack.

Using a stolen smart card, an attacker may attempt to misuse its contents. Nonetheless, as remarked in sub-section 4.3, an adversary cannot guess password using stolen smart card parameters $\{D_{ij}',\ Ei,\ Fi'\}$. Hence, despite stealing the SC contents, that adversary may not initiate any kind of guessing or impersonation attack due to the lack of information about BIO_i parameter and dynamic identity *Ai*.

5.1.6 *Session Key Security*

This feature ensures that the agreed session key may be only in the knowledge of the legitimate participants in the session, such as user and server.

In contributed protocol, the session key is established by computing $SK{=}\ h(SKij\ ||\ Ai\ ||\ Bij\ ||\ W\ ||M\ ||N\ ||\ SID_{j})$. For establishing a legal session key the attacker requires to access *m* and *n*. These are high entropy integers, and cannot be guessed in polynomial time. An adversary cannot derive *n* from *N=n.P*, neither *m* from *M=m.P*, which might be intercepted by the adversary during the communication of messages on insecure channel. Hence, the computation of *m* or *n* from *M* and *N* is hard bounded by ECDLP problem.

5.1.7 *Known-Key Security*

This security feature assures the confidentiality of private keys of the communicating session members, if the current session key is exposed to the attacker.

In contributed protocol, if the session key $SK{=}\ h(SKij\ ||\ Ai\ ||\ Bij\ ||\ W\ ||M\ ||N\ ||\ SID_{j})$ is exposed by any means, the attacker may not be able to guess the user's password PW_i or server master key *x*. Thus, an attacker will not be able to derive the secrets from any revealed session key. Hence, the proposed protocol corresponds to the trait of known-key security.

5.1.8 *Perfect Forward Secrecy*

This feature focuses on the confidentiality of session keys, in case the high entropy private key of any participant (Ui) or (Sj) is stolen by the attacker [47-49].

The contributed scheme fulfills the requirement of perfect forward secrecy, notwithstanding the fact, that high entropy secrets of participating entities are exposed to the adversary. That is, if the RC's secret *x* is leaked, an adversary may not be able to compute previous session keys for the lack of knowledge of other parameters i.e. $SKij$, Ai , W and B_{ij} in a session key $SK= h(SKij \| Ai \| B_{ij} \| W \|M \|N \| SID_j)$.

5.1.9 *Mutual Authentication*

This property suggests that the communicating members must verify one another's identities during the same authentication protocol.

The Lin et al. scheme lack mutual authentication, as it may suffer replay attack that damages the feature of mutual authentication. However, the contributed model provides mutual authentication to both legitimate entities, Ui and Sj. An adversary, having intercepted the publicly available messages $\{M, X, Hi, N, Vi, Ji\}$ might try to modify or replay a message on both sides to deceive the legitimate participants. However, Ui and Sj mutually authenticate one another, and nullify the chances of any modification or replay attack by computing and verifying the equation $Vi^* ?=h(SKi \| Ai \| Bij \|W \|M)$ and $Ji ?= h(SKj \| Ai \| Bij \|M \|N \| SID_j)$. Hence, in proposed protocol both of the entities could authenticate one another.

5.1.10 *Anonymous Authentication and immunity from Trace attack*

The anonymous authentication ensures anonymity for the subscriber during its communication with Sj for mutual authentication phase. Further, the trace attack refers to the identification of a user's location by the adversary during the exchange of messages.

In contributed model, the attacker may not be able to use the real identities of the interacting members upon the utilization of intercepted message contents. This is because, Ui sends its dynamic identity Ai in the form of $X=Ai\oplus W$ after computing the parameter W. An adversary cannot recover the user's identity IDi from either X or other stolen smart card values, until Ui's biometric BIO_i or server's secret x are compromised. The biometric parameter provides additional protection to user-oriented credentials including identity and password. Likewise, an attacker may not be able to distinguish among various sessions, nor could identify location of a user. Therefore, this scheme not only affords sufficient anonymity to the user but also provides resistance from trace attack.

5.1.11 *Resists user impersonation attack*

As we see earlier, the proposed scheme offers mutual authentication to its members, and could resist replay attack and modification attacks. In the light of these proofs (see section 5.1.1, 5.1.2), we can rightly say that our scheme is immune to user impersonation attacks.

5.1.12 *Resists De-synchronization Attack*

The proposed scheme employed bio-hashing $H(.)$ to resist de-synchronization attack. This is a kind of self-attack that might occur without any sort external adversary. Since direct capturing of biometric without undergoing bio-hashing might suffer matching problems, so hash-function helps synchronize the captured and the stored biometric.

5.1.13 *Privileged insider attack*

A malicious insider may access the registration request information sent by the user during registration process. In proposed scheme, we have employed two random numbers r_2 and r_3 on the user's side to avoid the possible privileged insider attacks. The user encrypts the sent messages by taking XOR with r_2 and r_3, and decrypts the messages, received from RC, using the same r_2 and r_3. This procedure deceives the malicious insider by the use of encryption on the part of sent messages from user. In this manner, the proposed scheme remains protected of a privileged insider attack.

5.1.14 *Session-specific temporary information threat*

If session-specific temporary integers are leaked, an adversary might attempt to compute session keys [50-52]. However, unlike Lin et al., the contributed scheme is immune to such kind of attack. The reason being, proposed scheme's session key $SK= h(SKi \| Ai \| B_{ij} \| W \|M \|N \| SID_j)$ can only be computed, if the adversary is capable of accessing Ai, B_{ij} and W parameters along with the compromise of session-specific temporary values. Hence, our scheme is immune from temporary information threat.

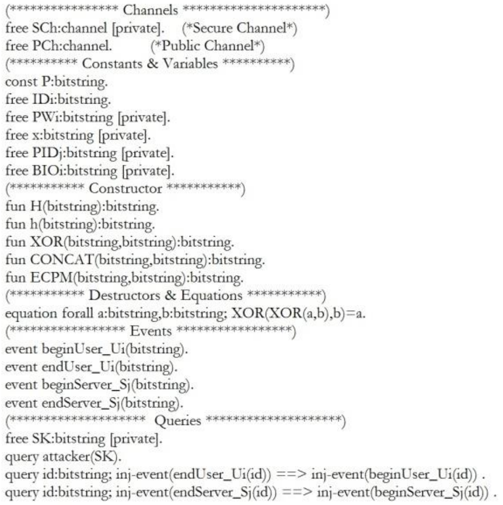
5.1.15 Resists Key-Compromise Impersonation attack

The contributed scheme is resistant to KCI attack, since a malicious privileged insider $\mathfrak{T}$ may not be able to derive a parameter *Bij* in case, the smart card contents are revealed to $\mathfrak{T}$. However, even if the Bij parameter is exposed to adversary accidentally, yet the adversary $\mathfrak{T}$ may not construct an up-to-date *Vi* message, due to the unknown *Ai* parameter. Thus, the adversary may not be able to impersonate as a server. Hence, the contributed protocol is immune to KCI attack.

5.2 Automated Security Proof

The purpose of this automated tool-based simulation is to measure the robustness of our scheme against an attacker. ProVerif [53] is one of the effective tools as used by the research community to gauge the scheme's robustness against attacks and infringement of the privacy. ProVerif relies on universally accepted rules of π calculus (applied) that could support various crypto-primitives like one-sided hash digest functions, encryption, digital signatures, Diffie-Helman etc. For measuring the security robustness of the contributed protocol, we have tested and analyzed the findings regarding the scheme's security in ProVerif simulation tool.

We initiate the simulation testing, initially by specifying two communication channels: the private and public channels i.e. *SCh* and *PCh* respectively, among various participants. We also define few constants and variables for this simulation. Besides, some constructors and equations are used for Bio-hashing, one-sided hash function, exclusive-or function, concatenation and the elliptic curve-based scalar point multiplication as shown in figure 3. We also design the queries to test the security and correctness of contributed scheme. .

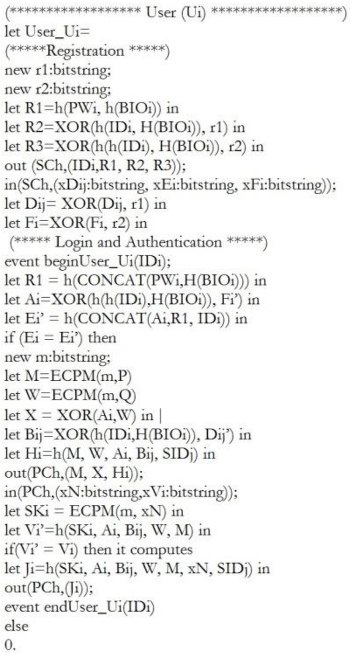


```
(**************** Channels ********************)
free SCh:channel [private].    (*Secure Channel*)
free PCh:channel.         (*Public Channel*)
(********** Constants & Variables **********)
const P:bitstring.
free IDi:bitstring.
free PWi:bitstring [private].
free x:bitstring [private].
free PIDj:bitstring [private].
free BIOi:bitstring [private].
(*********** Constructor ***********)
fun H(bitstring):bitstring.
fun h(bitstring):bitstring.
fun XOR(bitstring,bitstring):bitstring.
fun CONCAT(bitstring,bitstring):bitstring.
fun ECPM(bitstring,bitstring):bitstring.
(*********** Destructors & Equations ***********)
equation forall a:bitstring,b:bitstring; XOR(XOR(a,b),b)=a.
(***************** Events *****************)
event beginUser_Ui(bitstring).
event endUser_Ui(bitstring).
event beginServer_Sj(bitstring).
event endServer_Sj(bitstring).
(********************  Queries ********************)
free SK:bitstring [private].
query attacker(SK).
query id:bitstring; inj-event(endUser_Ui(id)) ==> inj-event(beginUser_Ui(id)) .
query id:bitstring; inj-event(endServer_Sj(id)) ==> inj-event(beginServer_Sj(id)) .
```

Figure 3. Constants, Events and Queries

For the simulation, we defined two events for each participating member in session, such as user and server. The beginning and finishing events for the user (Ui) are beginUser_Ui(bitstring) as well as endUser_Ui(bitstring). Likewise, the corresponding events for the server (Sj) are defined as beginServer_Sj(bitstring) as well as endServer_Sj(bitstring) as well. The accuracy of the contributed technique could be tested by verifying the corresponding relationship between either of entity's initial or final events. The events are also shown in figure 3.

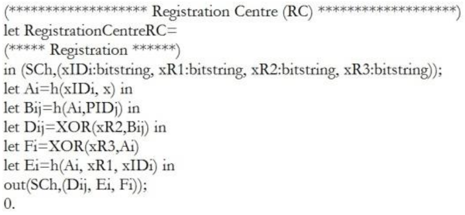
We build three separate processes, namely *User_Ui*, *Server_Sj*, *RegistrationCentreRC*, for modelling three participating entities as Ui, Sj and RC, respectively. First, the User_Ui process generates r1 and r2 numbers, and compute R1, R2 and R3. Next, it sends IDi, R1, R2, and R3 using the SCh channel to *RegistrationCentreRC* process. While, on receiving xDij, xEi and xFi from the same process, it computes and updates the values of Dij and Fi in smart card. During mutual authentication stage, the process *User_Ui* sends the parameters M, X and Hi using public channel PCh towards *Server_Sj*. Next, after receiving the xN and xVi parameters from *Server_Sj* process, the *User_Ui* process further computes Ski, Vi', and compares Vi' against Vi. If it matches, then computes Ji and sends Ji towards *Server_Sj* process for further proceedings as shown in Figure 4.



```
(***************** User (Ui) *****************)
let User_Ui=
(*****Registration *****)
new r1:bitstring;
new r2:bitstring;
let R1=h(PWi, h(BIOi)) in
let R2=XOR(h(IDi, H(BIOi)), r1) in
let R3=XOR(h(h(IDi), H(BIOi)), r2) in
out (SCh,(IDi,R1, R2, R3));
in(SCh,(xDij:bitstring, xEi:bitstring, xFi:bitstring));
let Dij= XOR(Dij, r1) in
let Fi=XOR(Fi, r2) in
 (***** Login and Authentication *****)
event beginUser_Ui(IDi);
let R1 = h(CONCAT(PWi,H(BIOi))) in
let Ai=XOR(h(h(IDi),H(BIOi)), Fi') in
let Ei' = h(CONCAT(Ai,R1, IDi)) in
if (Ei = Ei') then
new m:bitstring;
let M=ECPM(m,P)
let W=ECPM(m,Q)
let X = XOR(Ai,W) in |
let Bij=XOR(h(IDi,H(BIOi)), Dij') in
let Hi=h(M, W, Ai, Bij, SIDj) in
out(PCh,(M, X, Hi));
in(PCh,(xN:bitstring,xVi:bitstring));
let SKi = ECPM(m, xN) in
let Vi'=h(SKi, Ai, Bij, W, M) in
if(Vi' = Vi) then it computes
let Ji=h(SKi, Ai, Bij, W, M, xN, SIDj) in
out(PCh,(Ji));
event endUser_Ui(IDi)
else
0.
```

Figure 4. User process simulation

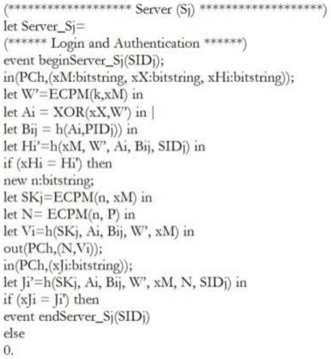
The *RegistrationCentreRC* process gets xIDi, xR1, xR2 and xR3 factors from *User_Ui* process using SCh, and calculates Ai, Bij, Dij, Fi and Ei. Next it sends Dij, Ei and Fi to *User_Ui* process using SCh as shown in Figure 5.



```
(******************* Registration Centre (RC) *******************)
let RegistrationCentreRC=
(***** Registration ******)
in (SCh,(xIDi:bitstring, xR1:bitstring, xR2:bitstring, xR3:bitstring));
let Ai=h(xIDi, x) in
let Bij=h(Ai,PIDj) in
let Dij=XOR(xR2,Bij) in
let Fi=XOR(xR3,Ai)
let Ei=h(Ai, xR1, xIDi) in
out(SCh,(Dij, Ei, Fi));
0.
```

Figure 5. RC Simulation

The *Server_Sj* process gets the parameters xM, xX and xHi from *User_Ui* process to check the user's authenticity as shown in Figure 6. Thereafter, it computes W', Ai, Bij and Hi'. Then it compares xHi and Hi'. If the equality holds true, then generates n and computes SKj, N and Vi. Then it sends N and Vi to *User_Ui* process through PCh channel. Similarly, *Server_Sj* receives xJi from *User_Ui* process, and computes Ji' to compare with xJi. If the equality holds, it confirms and verifies the *User_Ui* process, otherwise aborts the session.



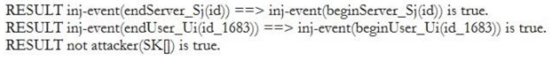
```
(******************* Server (Sj) *******************)
let Server_Sj=
(****** Login and Authentication ******)
event beginServer_Sj(SIDj);
in(PCh,(xM:bitstring, xX:bitstring, xHi:bitstring));
let W'=ECPM(k,xM) in
let Ai = XOR(xX,W') in |
let Bij = h(Ai,PIDj)) in
let Hi'=h(xM, W', Ai, Bij, SIDj) in
if (xHi = Hi') then
new n:bitstring;
let SKj=ECPM(n, xM) in
let N= ECPM(n, P) in
let Vi=h(SKj, Ai, Bij, W', xM) in
out(PCh,(N,Vi));
in(PCh,(xJi:bitstring));
let Ji'=h(SKj, Ai, Bij, W', xM, N, SIDj) in
if (xJi = Ji') then
event endServer_Sj(SIDj)
else
0.
```

Figure 6. Server process simulation

The three participating principals interact for an unrestrained number of parallel sessions, in this way, those processes act in replication as depicted in the following.

```
process
((!User_Ui) | (!RegistrationCentreRC) | (!Server_Sj) )
```

We get to the following results after employing the above queries in this simulation.



```
RESULT inj-event(endServer_Sj(id)) ==> inj-event(beginServer_Sj(id)) is true.
RESULT inj-event(endUser_Ui(id_1683)) ==> inj-event(beginUser_Ui(id_1683)) is true.
RESULT not attacker(SK[]) is true.
```

Figure 7. Simulation Result

The results (3) and (4) as shown in Figure 7 specify that the respective processes started as well as ended successfully. At the same time, the result (5) warrants that the attacker query is not able to derive or compute the session key as constructed by the processes during the authentication procedure.

5.3 BAN Logic-based Security Analysis

This sub-section exhibits security analysis employing Burrows Abadi Needham (BAN) logic [54-55], which is a logic model analyzing the security features in terms of mutual authentication and the inability of computing session key. Some of the terms are employed in the explanation of BAN logic as given below.

Principals denoted with (**Λ**), are the active participating agents in our protocol.
Keys are used in encryption for symmetric encryption.
Nonces are non-repeatable chunks of the message.
Some further notations that are employed in the BAN logic analysis are stated below:

$\boldsymbol{\Lambda} \mid\equiv \Xi$: **Λ** believes Ξ.
$\boldsymbol{\Lambda} \mid\sim \Xi$: **Λ** once said Ξ.
$\boldsymbol{\Lambda} \lhd \Xi$: **Λ** sees Ξ.
$\boldsymbol{\Lambda} \Rightarrow \Xi$: **Λ** has got jurisdiction on Ξ;
$\sharp(\Xi)$: Ξ is fresh.
$(\Xi)_{\Xi'}$ Formulae (Ξ) is combined with (Ξ').
(Ξ, Ξ'): Ξ or Ξ' are the components of message (Ξ, Ξ').
$(\Xi, \Xi')_{\underline{k}}$: Ξ or Ξ' is encrypted by the key $\underline{k}$.
$\boldsymbol{\Lambda} \xleftrightarrow{\underline{k}} \boldsymbol{\Lambda}'$: **Λ** and **Λ'** communicate using the shared key $\underline{k}$.
$\langle \Xi, \Xi' \rangle_{\underline{k}}$: Ξ or Ξ' is hashed by using key $\underline{k}$.

Some of the rules, we employed in the BAN Logic, such as (message_meaning_rule) implies R1, (nonce_verification_rule) implies R2, (jurisdiction_rule) implies R3, (freshness_conjuncatenation_rule) implies R4, (belief_rule) implies R5, and session_key_rule implies R6 as depicted under:

R1: $\dfrac{\boldsymbol{\Lambda} \mid\equiv \boldsymbol{\Lambda} \overset{K}{\leftrightarrow} \boldsymbol{\Lambda}',\ \ \boldsymbol{\Lambda} \lhd \langle \Xi \rangle_{\Xi'}}{\boldsymbol{\Lambda} \mid\equiv \boldsymbol{\Lambda}' \mid\sim \Xi}$

R2: $\dfrac{\boldsymbol{\Lambda} \mid\equiv \sharp(\Xi),\ \ \boldsymbol{\Lambda} \mid\equiv \boldsymbol{\Lambda}' \mid\sim \Xi}{\boldsymbol{\Lambda} \mid\equiv \boldsymbol{\Lambda}' \mid\equiv \Xi}$

R3: $\dfrac{\boldsymbol{\Lambda} \mid\equiv \boldsymbol{\Lambda}' \Rightarrow \Xi,\ \ \boldsymbol{\Lambda} \mid\equiv \boldsymbol{\Lambda}' \mid\equiv \Xi}{\boldsymbol{\Lambda} \mid\equiv \Xi}$

R4: $\dfrac{\boldsymbol{\Lambda} \mid\equiv \sharp(\Xi)}{\boldsymbol{\Lambda} \mid\equiv \sharp(\Xi,\ \Xi')}$

R5: $\dfrac{\boldsymbol{\Lambda} \mid\equiv (\Xi),\ \ \boldsymbol{\Lambda} \mid\equiv (\Xi')}{\boldsymbol{\Lambda} \mid\equiv (\Xi,\ \Xi')}$

R6: $\dfrac{\boldsymbol{\Lambda} \mid\equiv \sharp(\Xi),\ \ \boldsymbol{\Lambda} \mid\equiv \boldsymbol{\Lambda}' \mid\equiv \Xi}{\boldsymbol{\Lambda} \mid\equiv \boldsymbol{\Lambda} \overset{K}{\leftrightarrow} \boldsymbol{\Lambda}'}$

Our contributed scheme should meet the understated goals to support the security features between server S_j and user U_i, by employing BAN logic.

Ḡ1 : S_j |≡ S_j $\xleftrightarrow{S\_K}$ U_i

Ḡ2 : S_j |≡ U_i |≡ U_i $\xleftrightarrow{S\_K}$ S_j

Ḡ3: U_i |≡ S_j $\xleftrightarrow{S\_K}$ U_i

Ḡ4: U_i |≡ S_j |≡ S_j $\xleftrightarrow{S\_K}$ U_i

First, we change the communicated messages into idealized form as shown below:

M_1: U_i → S_j: $M, X, Hi: \{mP, (Ai)_{mQ}, \langle mP, W, Ai, SID_j \rangle_{Bij}\}$
M_2: S_j →U_i: $N, Vi: \{nP, \langle mnP, Ai, mQ, M\rangle_{Bij}\}$
M_3: U_i → S_j: $Ji: \{\langle mP, nP, Ai, SID_j \rangle_{mnP, Bij,}\}$

Secondly, the following assumptions are developed to prove the security of contributed work.

Ꝁ1 : U_i |≡ $\sharp m$
Ꝁ 2 : S_j |≡ $\sharp n$
Ꝁ 3 : U_i |≡ S_j $\xleftrightarrow{(B_{ij},\ Ai,\ SK_{ij})}$ U_i
Ꝁ4 : S_j |≡ S_j $\xleftrightarrow{(B_{ij},\ Ai,\ SK_{ij})}$ U_i
Ꝁ5 : U_i |≡ S_j |≡ U_i $\xleftrightarrow{(B_{ij},\ Ai,\ SK_{ij})}$ S_j
Ꝁ6 : Sj |≡ U_i |≡ U_i $\xleftrightarrow{(B_{ij},\ Ai,\ SK_{ij})}$ S_j
Ꝁ7 : U_i |≡ S_j ⇒ nP
Ꝁ8 : S_j |≡ U_i ⇒ mP

Thirdly, the constructed idealized forms such as M_1, M_2 and M_3 of our scheme can be evaluated using the above postulates and premises.

By applying the given rules, notations, premises along with idealizations, we arrive at the understated derivations:

Using M1 and M3 of those idealized forms:

M_1: U_i → S_j: $M, X, Hi: \{mP, (Ai)_{mQ}, \langle mP, W, Ai, SID_j \rangle_{Bij}\}$

M_3: U_i → S_j: $Ji: \{\langle mP, nP, Ai, SID_j \rangle_{mnP, Bij,}\}$

On applying the seeing_rule, we have

S1: S_j ⊲ $M, X, Hi: \{mP, (Ai)_{mQ}, \langle mP, W, Ai, SID_j \rangle_{Bij}\}$

S2: S_j ⊲ $Ji: \{\langle mP, nP, Ai, SID_j \rangle_{mnP, Bij,}\}$

Now using S1, S2, Ꝁ3 and *R1*, we say

S3: S_j |≡ U_i ~ $\{mP, (Ai)_{mQ}, \langle mP, W, Ai, SID_j \rangle_{Bij}\}$

S4: S_j |≡ U_i ~ $\{\langle mP, nP, Ai, SID_j \rangle_{mnP, Bij,}\}$

Referring S3, S4, Ꝁ1, *R2* and *R4*, we deduce

S5: S_j |≡ U_i |≡ $\{mP, (Ai)_{mQ}, \langle mP, W, Ai, SID_j \rangle_{Bij}\}$

S6: S $|\equiv$ U_i $|\equiv$ $\{\langle mP, nP, Ai, SID_j \rangle_{mnP, Bij,}\}$

Referring S5, S6, Ꝁ4, Ꝁ8 and *R3*, we get

S7: S_j $|\equiv$ $\{mP, (Ai)_{mQ}, \langle mP, W, Ai, SID_j \rangle_{Bij}\}$

S8: S_j $|\equiv$ $\{\langle mP, nP, Ai, SID_j \rangle_{mnP, Bij,}\}$

Using S7, S8, Ꝁ4, ($SK= h(mnP \,||\, Ai \,||\, Bij \,||\, W \,||mP \,||nP \,||\, SID_j)$) and *R6*, we get

S9: S_j $|\equiv$ S_j $\xleftrightarrow{SK}$ U_i **(Ḡ1)**

Referring to S9, Ꝁ6 we employe *R6* as

S10: S_j $|\equiv$ U_i $|\equiv$ U_i $\xleftrightarrow{S\_K}$ S_j **(Ḡ2)**

Next, again visualizing the idealized form M2:

M$_2$: S_j → U_i: $N, Vi: \{nP, \langle mnP, Ai, mQ, M\rangle_{Bij}\}$

On applying the seeing_rule, we get

S11: U_i $\lhd$ $N, Vi: \{nP, \langle mnP, Ai, mQ, M\rangle_{Bij}\}$

Referring to S11, Ꝁ4 and *R1*, we infer

S12: U_i $|\equiv$ S_j ~ $\{nP, \langle mnP, Ai, mQ, M\rangle_{Bij}\}$

Using S12, Ꝁ2, *R4* and *R2*, we infer

S13: U_i $|\equiv$ S_j $|\equiv$ $\{nP, \langle mnP, Ai, mQ, M\rangle_{Bij}\}$

Referring S13, Ꝁ3, Ꝁ7 and *R3*, we get

S14: U_i$|\equiv$ $\{nP, \langle mnP, Ai, mQ, M\rangle_{Bij}\}$

From S14, A3, $SK= h(mnP \,||\, Ai \,||\, Bij \,||\, W \,||mP \,||nP \,||\, SID_j)$, and *R6*, we have

S15: U_i $|\equiv$ S_j $\xleftrightarrow{S\_K}$ U_i **(Ḡ3)**

Referring to S15, Ꝁ5, we employ *R6* as

S16: U_i $|\equiv$ S_j $|\equiv$ S_j $\xleftrightarrow{S\_K}$ U_i **(Ḡ4)**

The presented analysis of BAN logic formally verifies that the contributed scheme ensures mutual authentication which implies that the established session key *SK* is mutually shared between the legitimate members (Ui and Sj).

6. COMPARISON AND PERFORMANCE EVALUATION

In performance analysis section, we analyze the security of contributed model with other smart card-based biometric multi-server authentication schemes in comparison. Table II demonstrates the comparison of different schemes for susceptibility to threats, which signifies the contributed model as a resistant and efficient authenticated key agreement protocol in comparison with other schemes. The comparison in Table II bears Li & Hwang [16], Chuang et al. [19], Lin et al. [22], Irshad et al. [44], Kumari et al. [56], Li et al. [57], Amin et al. [58] and proposed technique, which depicts that the contributed scheme is more resistant to attacks than those schemes as pointed, or efficient in other aspects.

We analyzed and tabulate the computational costs after installing MIRACL library [50] in a mobile gadget (Lenovo Zuk Z1 having Quad-core 2.5 Ghz processor with 3GB of Random Access Memory (RAM) and Android Operating System (OS) V5.1.1) , and a desktop computer (HP E8300 Core i5, 2.9 Ghz processor with 6GB of memory employing Ubuntu OS 16.10). The simulation was performed on mobile gadget for user, and personal computer on server's end. The computational cost and running time of various comparative studies is depicted in Table III.

Table II: Comparison for Multi-Server Authentication schemes

	[16]	[19]	[22]	[44]	[56]	[57]	[58]	Ours
Anonymity supported	×	×	√	√	√	√	√	√
Resists Offline password-guessing attack	√	×	√	√	√	√	√	√
Resists Replay attack	×	√	×	√	√	√	√	√
Resists privileged insider Attack	×	√	×	√	√	√	√	√
Mutual Authentication	×	×	×	√	√	√	√	√
Resists Stolen smart card attack	√	√	√	√	√	√	√	√
Resists user impersonation attack	×	×	×	√	√	√	√	√
Session key agreement	√	√	√	√	√	√	√	√
Resists session-specific temporary information attack	√	√	√	√	√	√	√	√
Resists De-synchronization attack	√	√	×	√	√	√	√	√
Resists KCI attack	√	√	×	√	√	√	√	√
Perfect forward secrecy	√	√	√	√	√	√	√	√
Mutual authentication with offline RC	√	√	√	×	×	×	×	√

√ : Implies, the feature is supported.
×: Implies, the feature is not supported.

Table III. Computational cost comparison

Schemes	Authentication phase (Computational cost)	
	Ui	*Sj/ RC*
[16]	$4T_H \approx 0.268$	$6T_H \approx 0.054$
[19]	$7T_H \approx 0.469$	$8T_H \approx 0.536$
[22]	$4T_H + 2T_{SE} + 2T_{ESM} \approx 22.9$	$4T_H + 2T_{SE} + 2T_{ESM} \approx 4.12$
[44]	$10T_H + 2T_{SE} + 4T_{ESM} \approx 45.84$	$11T_H + 3T_{SE} + 8T_{ESM} \approx 16.35$
[56]	$7T_H + 3T_{ESM} \approx 34.15$	$11T_H + 5T_{ESM} \approx 10.22$
[57]	$9T_H + 3T_{ESM} \approx 34.28$	$13T_H + 3T_{ESM} \approx 6.19$
[58]	$8T_H + 2T_{SE} + 4T_{ESM} \approx 45.71$	$13T_H + 3T_{SE} + 8T_{ESM} \approx 16.37$
Ours	$9T_H + 1T_{SE} + 3T_{ESM} \approx 34.41$	$4T_H + 1T_{SE} + 3T_{ESM} \approx 6.12$

To make the comparison of computational costs in Table III, we denote one-way hash function with T_H, elliptic scalar point multiplication T_{ESM}, symmetric key-based encryption T_{SE}, and ignoring the lightweight XOR function due to negligible cost. According to the simulation experiment, the computed delays for T_{ESM}, T_H and T_{SE} for user and server are shown in Table IV. According to Table III, although our scheme bears extra computational cost than schemes [16, 19, 22, 57], it is secure against many threats notably, privileged insider attack, trace attack, offline-password guessing threat, replay attack, stolen smart card threat, KCI attack, de-synchronization attack and user impersonation attack and also efficient in communication cost. Our scheme have not only less computational cost than schemes [44, 56-58] but also less communication cost, since these schemes employ RC during mutual authentication phase between user and server. Our scheme mutually authenticates user and server without involving RC, which minimizes the communication cost of scheme.

Table IV. Computational cost comparison

Time complexity for operations	*User (ms)*	*Server (ms)*
T_{ESM}	*11.227*	*2.025*
T_H	*0.067*	*0.009*
T_{SE}	*0.134*	*0.018*

The Table II and III manifests that the contributed scheme is immune to all attacks as discovered in [16, 19, 22], further it operates in less communication rounds as compared to [44, 56-68] which brings down the communication cost. Besides, the schemes [16, 19] do not provide anonymity. The schemes [16, 19, 22] do not provide mutual authentication and also prone to user impersonation attack. The schemes [16, 22] are not immune to replay attack and privileged insider attack.

Table V. Communication cost in bits

Schemes	*bits*
[16]	*800*
[19]	*1280*
[22]	*1120*
[44]	*2176*
[56]	*3520*
[57]	*3360*

[58]	*2336*
Ours	*1280*

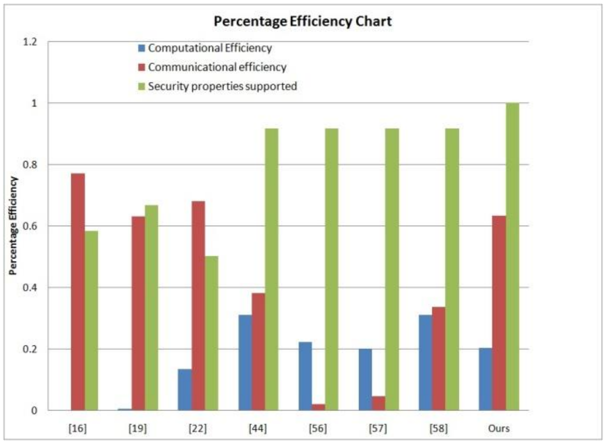


Figure 8. Efficiency comparison of schemes

To compute the round-trip based communicational cost, we assume that the operation hash digest (SHA-1) affords 160-bits, user or server identity affords 160-bits, random integer affords 160-bits, and the elliptic curve point uses 320-bits. Our scheme bears less communication cost than schemes [44, 56-58] which leads to better efficiency since the involvement of central authority for every session establishment may prove to be costly in peak hours in [44, 56-58]. The schemes [16, 19] has less communication cost since these forego without elliptic curve operations which takes 320-bits in transit as communication cost. The performance efficiency analysis of various schemes clearly indicates in Fig. 8 that our scheme yields enhanced security features along with computational and communicational efficiencies on average, comparatively. Hence, in the light of above performance evaluation analysis as depicted in Table II and III, we can safely deduce that the contributed scheme is a more secure and efficient computationally as well as in communicational terms.

CONCLUSION

The multi-server authentication has been proved to be as one of the crucial requirements of the state-of-the-art communication technology infrastructure. This paper studies and presents the review of Lin et al.'s multi-server authentication protocol. The cryptanalysis to Lin et al. reveals the five ways, in which it might be vulnerable to attacks, i.e. replay attack, trace attack, de-synchronization attack, key-compromise impersonation threat and privileged insider threat. The proposed scheme not only serves as an enhanced and improved version of Lin et al. scheme, but also proves to be efficient in terms of computation and communication with many contemporary multi-server authentication protocols. Moreover, the security features of this research work are supported with BAN logic-based formal security analysis and automated analysis using ProVerif tool. In future, we intend to address the privacy concerns for cloud-based multi-server authentication models.

REFERENCES

[1] Lamport, L. (1981). Password authentication with insecure communication. Communication of ACM, 24(11), 770–772.
[2] Li, X., Ma, J., Wang, W. D., Xiong, Y. P., & Zhang, J. S. (2013). A novel smart card and dynamic ID-based remote user authentication scheme for multi-server environments. Mathematical and Computer Modelling, 58(1–2), 85–95.
[3] Sun, H. M. (2000). An efficient remote user authentication scheme using smart cards. IEEE Transactions on Consumer Electronics, 46(4), 958–961.
[4] Awashti, A. K., & Sunder, L. (2004). An enhanced remote user authentication scheme using smartcards. IEEE Transactions on on Consumer Electronics, 50(2), 583–586.
[5] Khan, M. K. (2009). Fingerprint biometric-based self and deniable authentication schemes for the electronic world. IETE Technical Review, 26(3), 191–195.

[6] Liao, Y. P., & Wang, S. S. (2009). A secure dynamic ID based remote user authentication scheme for multi-server environment. Computer Standards and Interfaces, 31(1), 24–29.

[7] Wen, F. T., & Li, X. L. (2011). An improved dynamic ID-based remote user authentication with key agreement scheme. Computers and Electrical Engineering, 38(2), 381–387.

[8] Hsiang, H. C., & Shih, W. K. (2009). Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment. Computer Standards and Interfaces, 31(6), 1118–1123.

[9] Lee, C. C., Lin, T. H., & Chang, R. X. (2011). A secure dynamic ID based remote user authentication scheme for multi-server environment using smart cards. Expert Systems with Applications, 38(11), 13863–13870.

[10] Guo, D. L., & Wen, F. T. (2014). Analysis and improvement of a robust smart card based-authentication scheme for multi-server architecture. Wireless Personal Communications, 78(1), 475–490.

[11] Wen, F. T., Susilo, W., & Yang, G. M. (2013). A robust smart card based anonymous user authentication protocol for wireless communications. Security and Communication Networks, 7(6), 987–993.

[12] Khan, M. K., & Zhang, J. (2007). Improving the security of a flexible biometrics remote user authentication scheme. Computer Standards and Interfaces, 29(1), 82–85.

[13] Sood, S. K., Sarje, A. K., & Singh, K. (2011). A secure dynamic identity based authentication protocol for multi-server architecture. Journal of Network and Computer Applications, 34(2), 609–618.

[14] Li, X., Xiong, Y. P., Ma, J., & Wang, W. D. (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.

[15] Xue, K. P., Hong, P. L., & Ma, C. S. (2014). A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture. Journal of Computer and System Sciences, 80(1), 195–206.

[16] Li, C. T., & Hwang, M. S. (2010). An efficient biometrics-based remote user authentication scheme using smart cards. Journal of Network and Computer Applications, 33(1), 1–5.

[17] Kim, H. S., Lee, J. K., &Yoo, K. Y. (2003). ID-based password authentication scheme using smart cards and fingerprints. ACM SIGOPS Operating Systems Review, 37(4), 32–41.

[18] Lee, J. K., Ryu, S. R., & Yoo, K. Y. (2002). Finger print-based remote user authentication scheme using smart cards. Electronics Letters, 38(12), 554–555.

[19] Chuang, M. C., & Chen, M. C. (2014). An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics. Expert Systems with Applications,41(4), 1411–1418.

[20] Kocher, P., Jaffe, J., & Jun, B. (1999). Differential power analysis. Advances in Cryptology—CRYPTO'99, 1666(16), 388–397.

[21] Messerges, T. S., Dabbish, E. A., & Sloan, R. H. (2002). Examining smart-card security under the threat of power analysis attacks. IEEE Trans on Computers, 51(5), 541–552.

[22] Lin, H, Fengtong W., and Chunxia D. (2015). An Improved Anonymous Multi-Server Authenticated Key Agreement Scheme Using Smart Cards and Biometrics. Wireless Personal Communications 1-12.

[23] Chuang, Y.-H., and Tseng, Y.-M., An efficient dynamic group key agreement protocol for imbalanced wireless networks. Int. J. Netw. Manag. 20(4):167–180, 2010.

[24] Chatterjee, S., Roy, S., Das, A. K., Chattopadhyay, S., Kumar, N., & Vasilakos, A. V. (2016). Secure biometric-based authentication scheme using Chebyshev chaotic map for multi-server environment. IEEE Transactions on Dependable and Secure Computing.

[25] Reddy, A. G., Yoon, E. J., Das, A. K., Odelu, V., & Yoo, K. Y. (2017). Design of Mutually Authenticated Key Agreement Protocol Resistant to Impersonation Attacks for Multi-Server Environment. IEEE Access, 5, 3622-3639.

[26] Odelu, V., Das, A. K., & Goswami, A. (2015). A secure biometrics-based multi-server authentication protocol using smart cards. IEEE Transactions on Information Forensics and Security, 10(9), 1953-1966.

[27] Irshad, A., Chaudhry, S. A., Kumari, S., Usman, M., Mahmood, K., & Faisal, M. S. (2017). An improved lightweight multiserver authentication scheme. International Journal of Communication Systems, 30(17).

[28] Kumari, S., Das, A. K., Li, X., Wu, F., Khan, M. K., Jiang, Q., & Islam, S. H. (2017). A provably secure biometrics-based authenticated key agreement scheme for multi-server environments. Multimedia Tools and Applications, 1-31.

[29] Jangirala, S., Mukhopadhyay, S., & Das, A. K. (2017). A Multi-server Environment with Secure and Efficient Remote User Authentication Scheme Based on Dynamic ID Using Smart Cards. Wireless Personal Communications, 1-33.

[30] Reddy, A. G., Das, A. K., Yoon, E. J., & Yoo, K. Y. (2016). An Anonymous Authentication with Key-

Agreement Protocol for Multi-Server Architecture Based on Biometrics and Smartcards. KSII Transactions on Internet & Information Systems, 10(7).

[31] Reddy, A. G., Das, A. K., Odelu, V., & Yoo, K. Y. (2016). An enhanced biometric based authentication with key-agreement protocol for multi-server architecture based on elliptic curve cryptography. PloS one, 11(5), e0154308.
[32] Irshad, A., Kumari, S., Li, X., Wu, F., Chaudhry, S. A., & Arshad, H. (2017). An Improved SIP Authentication Scheme Based on Server-Oriented Biometric Verification. Wireless Personal Communications, 97(2), 2145-2166.
[33] He, D., & Wang, D. (2015). Robust biometrics-based authentication scheme for multiserver environment. IEEE Systems Journal, 9(3), 816-823.
[34] 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.
[35] 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
[36] 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.
[37] 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.
[38] 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.
[39] 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.
[40] 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.
[41] 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.
[42] 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.
[43] 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.
[44] 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
[45] 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 Comp. Sci., 9(1), 142-156.
[46] 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.
[47] Chaudhry, S. A. (2016). A secure biometric based multi-server authentication scheme for social multimedia networks. Multimedia Tools and Applications, 75(20), 12705-12725.
[48] 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.
[49] 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.
[50] 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.
[51] 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.

[52] 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.
[53] 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/).
[54] 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.
[55] 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.
[56] Kumari, S., Li, X., Wu, F., Das, A. K., Choo, K. K. R., & Shen, J. (2017). Design of a provably secure biometrics-based multi-cloud-server authentication scheme. Future Generation Computer Systems, 68, 320-330.
[57] Li, X., Wang, K., Shen, J., Kumari, S., Wu, F., & Hu, Y. (2016). An enhanced biometrics-based user authentication scheme for multi-server environments in critical systems. Journal of Ambient Intelligence and Humanized Computing, 7(3), 427-443.
[58] Amin, R., Islam, S. H., Biswas, G. P., Khan, M. K., & Kumar, N. (2015). An efficient and practical smart card based anonymity preserving user authentication scheme for TMIS using elliptic curve cryptography. Journal of medical systems, 39(11), 180.