International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

# AN IMPROVED BIOMETRIC-BASED MULTI-SERVER AUTHENTICATION AND KEY AGREEMENT SCHEME

*Mridul K. Gupta[a], Rahul Kumar[b*]*

[a]*Chaudhary Charan Singh University, Meerut, 250004, India*
[b]*Chaudhary Charan Singh University, Meerut, 250004, India*
*E-mail address: rahulss.rahul1991@gmail.com*

**A B S T R A C T**

Biometrics certified protocols that are in line with the security requirements of the network. It is more important and widely deployed to be implemented in a multi-server environment. Due to advances in computing era and constraints within side the layout of the authentication protocols for single-server environment, the authentication protocols for multi-server settings had been a desired subject of research. Recently, Wang et al. [3] introduced a biometric based multi-server authentication and key agreement scheme and they said that their protocol is secure against various attacks. They also stated that their protocol is powerful. In this paper, we review Wang et al.'s protocol and find that their protocol is not secure against user impersonation attack, server spoofing attack. We also introduce an improvement of Wang et al.'s protocol.

***Keywords**: Multi-Server, Authentication, User Impersonation Attack, Bio-metrics, server spoofing attack.*

## 1. INTRODUCTION

With the fast improvement of the Internet, advances in records and communication technology has complemented the great online offerings for the allotted network, providing highly beneficial offers to the customers in diverse aspects including online therapy, online education, online shopping and internet banking. In the world of digital information, users can easily access a variety of services from distributed networks such as online shopping, online banks and pay-TV anywhere and anytime. Simple user authentication protocols are well suited for handling security issues for single user/server design scenarios. Nowadays, authentication protocols for multi-server formatting play a major role in the Internet world. There are three participants in a multi-server system, which includes the user, the server, and the registration center. A multi-server authentication scheme presents offerings to be accessed from other servers with a one-time registration.

Chuang and Chen [1] described an anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics and insisted that their protocol is safe from numerous attacks. Mishra et al. [2] reviewed Chuang and Chen's protocol and found that their protocol is suffering from denial-of-service attack, stolen smart card attack, user impersonation attack and server spoofing attack. To overcome these attacks from Chuang and Chen's protocol, Mishra et al. proposed an user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards. Wang et al. [3] analysed Mishra et al.'s protocol and found that their protocol is affected from masquerade attack, replay attack, denial-of-service attack, no perfect forward secrecy and no user revocation/re-registration phase.

In this paper, we review Wang et al.'s protocol [3] and show its weaknesses such as user impersonation attack and server spoofing attack. To conquer these weaknesses, we present an improved protocol.

## 2. PRELIMINARIES

Table 1 shows notations and their meaning.

**Table 1**

| Symbol | Meaning |
|---|---|
| $S_j$ | $j^{th}$ server |
| RC | Registration centre |
| $U_i$ | $i^{th}$ user |

| | |
|---|---|
| P | Generator of elliptic curve |
| $SID_j$ | Server's identity |
| $ID_i$ | User's identity |
| $AID_i$ | User's dynamic identity |
| $BIO_i$ | User's biometric |
| $H(\cdot)$ | Bio-hash function |
| $h(\cdot)$ | Hash function |
| $PW_i$ | User's password |
| SC | Smart card |
| SK | Session key |
| $\|\|$ | Concatenation |
| PSK | Pre shared key |
| $\mathcal{H}$ | Adversary |

## 3. REVIEW OF WANG ET AL.'S PROTOCOL

Wang at al.'s protocol includes six phases. Beginning from initialization phase, they discussed server enrollment phase, user enrollment phase, login phase, authentication and key agreement phase and password change phase.

### 3.1 Initialization phase

To boot up the system, *RC* selects a generator *P* of elliptic curve and chooses a secret key *y* as the system parameter.

### 3.2 Server enrollment phase

In this phase, server enrolls itself at the registration center *RC*. Server selects its own identity $SID_j$ and sends $\{SID_j\}$ to *RC* through open channel. When the request message is received by *RC* from the server. *RC* transmits the information $\{PSK\}$ to the server through secure channel.

### 3.3 User enrollment phase

First, user selects his/her identity $ID_i$, imprints $BIO_i$ and calculates $RPW_i = h(PW_i \| R_i)$ and forwards the message $\{ID_i, RPW_i\}$ to *RC* through open channel.

When a request message is received form the user then *RC* evaluates $A_i = h(ID_i \| x \| T_r)$, $B_i = RPW_i \oplus h(A_i)$, $C_i = B_i \oplus h(PSK)$, $D_i = PSK \oplus A_i \oplus h(PSK)$ and $V_i = h(ID_i \| RPW_i)$, where $T_r$ is registration time. Now *RC* inserts all information $\{B_i, C_i, D_i, V_i\}$ into *SC* and forwards $\{SC\}$ to the user. After receiving the message $\{SC\}$ from *RC*, user stores $P_i$ into *SC*.

### 3.4 Login phase

User embeds *SC* and enters $ID_j$, $PW_i$ and imprints $B_i$. Now *SC* evaluates $RPW_i = h(PW_i \| R_i)$ and checks whether $h(ID_i \| RPW_i) = V_i$ is valid. If it is valid, *SC* evaluates $h(PSK) = B_i \oplus C_i$. *SC* chooses a random number $N_1$ to evaluates $AID_i = ID_i \oplus h(N_1)$, $M_1 = RPW_i \oplus N_1 \oplus h(PSK)$ and $M_2 = h(AID_i \| N_1 \| RPW_i \| SID_j \| T_i)$, where $T_i$ is an additional timestamps.

Now, user transmits the login message $\{AID_i, M_1, M_2, B_i, D_i, T_i\}$ to $S_j$ through open channel.

### 3.5 Authentication and key agreement phase

When the login request message $\{AID_i, M_1, M_2, B_i, D_i, T_i\}$ is received from the user then $S_j$ checks whether $T_i - T_j \leq \Delta T$ holds. If the verification holds, $S_j$ continues to perform his/her next step. Otherwise, $S_j$ rejects $U_i$'s request. $S_j$ retrieves $RPW_i = B_i \oplus h(A_i)$, $A_i = PSK \oplus D_i \oplus h(PSK)$, $N_1 = RPW_i \oplus M_1 \oplus h(PSK)$ in order to verify whether $M_2 = h(AID_i \| N_1 \| RPW_i \| SID_j \| T_i)$ is consistent with $M_2$. If it holds, $S_j$ chooses a random number $N_2$ to evaluate their session key $SK = h(AID_i \| SID_j \| N_1 \| N_2)$. $S_j$ calculates $M_3 = N_2 \oplus h(AID_i \| N_1) \oplus h(PSK)$ and $M_4 = h(SID_j \| N_2 \| AID_i)$ in order to forward his/her authentication request message $\{SID_j, M_3, M_4\}$ to $U_i$ through an open channel. *SC* receives authentication request message from $S_j$ and retrieves $N_2 = M_3 \oplus h(AID_i \| N_1) \oplus h(PSK)$ and $SK = h(AID_i \| SID_j \| N_1 \| N_2)$ to check whether $M_4 = h(SID_j \| N_2 \| AID_i)$ holds. If it holds, *SC* evaluates $M_5 = h(SK \| N_1 \| N_2)$ in order to submit $U_i$'s authentication reply $\{M_5\}$ to $S_j$ over an insecure channel. $S_j$ verifies whether $M_5 = h(SK \| N_1 \| N_2)$ is valid.

If the verification is valid, $S_j$ further applies this $SK$ to communicate with $U_i$ in the following communication. Otherwise, authentication phase is rejected by $S_j$.

### 3.6 Password change phase

In this phase, user is allowed to modify his/her password easily without interfering with the server. First, user inserts his/her smartcard into a card reader and enters $ID_j$, $PW_i$ and also imprints $B_i$. Now, the smartcard reader evaluates $RPW_i = h(PW_i \| R_i)$ and verifies whether $V_i = h(ID_i \| RPW_i)$ is valid. If the equality does not hold then the connection is ended. Otherwise, the user selects new password $PW_i^{new}$ and evaluates $RPW_i^{new} = h(PW_i^{new} \| R_i)$, $B_i^{new} = B_i \oplus RPW_i^{new}$, $C_i^{new} = C_i \oplus RPW_i \oplus RPW_i^{new}$ and $V_i^{new} = h(ID_i \| RPW_i^{new})$. Finally, $SC$ replaces $B_i$ with $B_i^{new}$, $C_i$ with $C_i^{new}$, $V_i$ with $V_i^{new}$ in memory of the smartcard.

## 4. CRYPTANALYSIS OF WANG ET AL.'S PROTOCOL

In this phase, we describe the weaknesses of Wang et al.'s protocol [3].

### 4.1 User impersonation attack

Wang et al.'s protocol suffers from user impersonation attack as the explanation follows. Suppose, if $SC$ is stolen by any attacker $\mathcal{H}$, then $\mathcal{H}$ can harm the valid user. $\mathcal{H}$ eavesdrops all communication between $U_i$ and $S_j$. $\mathcal{H}$ has an ability to extract the stored data $\{B_i, C_i, D_i, V_i, P_i\}$ from $U_i$'s $SC$. Also, $\mathcal{H}$ is able to eavesdrop the login request message $\{AID_i, M_1, M_2, B_i, D_i, T_i\}$. Now, $\mathcal{H}$ evaluates $h(PSK) = B_i \oplus C_i$. Then, $\mathcal{H}$ chooses an arbitrary number $N_1^*$ and further computes $B_i^* = B_i \oplus h(PSK)$, $D_i^* = h(PSK)$, $M_1^* = B_i \oplus N_1^* \oplus h(PSK)$ and $M_2^* = h(AID_i \| N_1^* \| B_i \| SID_j \| T_i^*)$, wherein $T_i^*$ is a current timestamp. At last, $\mathcal{H}$ sends his\her login request message $\{AID_i, M_1^*, M_2^*, B_i^*, D_i^*, T_i^*\}$ to $S_j$ through the open channel. After getting login request message from $\mathcal{H}$, $S_j$ checks whether $T_i^* - T_j^* \leq \Delta T$ holds, where $T_j^*$ is the time when $S_j$ receives $\mathcal{H}$'s login request message. Therefore, $\mathcal{H}$ accepts $S_j$'s verification successfully and $S_j$ continues to execute the subsequent steps normally.

$S_j$ retrieves $A_i = D_i^* \oplus PSK \oplus h(PSK)$, $RPW_i = B_i^* \oplus h(A_i) = B_i$ and $N_1 = RPW_i \oplus M_1^* \oplus h(PSK) = N_1^*$ to verify whether $h(AID_i \| N_1 \| RPW_i \| SID_j \| T_i^*) = M_2^*$ holds. Further, $S_j$ chooses arbitrarily number $N_2^*$ and computes $SK_{ij}^* = h(AID_i \| SID_j \| N_1^* \| N_2^*)$, $M_3^* = N_2^* \oplus h(AID_I \| N_1^*) \oplus h(PSK)$ and $M_4^* = h(SID_j \| N_2^* \| AID_i)$. At last, $S_j$ forwards his\her authentication request message $\{SID_j, M_3^*, M_4^*\}$ to $\mathcal{H}$ through an insecure channel. After getting $S_j$'s authentication request message, $\mathcal{H}$ retrieves $N_2^* = M_3^* \oplus h(AID_i \| N_1^*) \oplus h(PSK)$ and $SK_{ij}^* = h(AID_i \| SID_j \| N_1^* \| N_2^*)$ in order to evaluate $M_5^* = h(SK_{ij} \| N_1^* \| N_2^*)$ and sent $\{M_5^*\}$ to $S_j$. $S_j$ verifies whether $h(SK_{ij}^* \| N_1^* \| N_2^*) = M_5^*$ is valid.

Therefore, $S_j$ authenticates $\mathcal{H}$ and they both apply the session key $SK_{ij}$ in the following communication. Unfortunately, $S_j$ mistakenly believes that he\she communicates with $U_i$. Therefore, Wang et al.'s protocol becomes week to the user impersonation.

### 4.2 Server spoofing attack

Assuming that $\mathcal{H}$ who is an insider but isn't another server $S_k$ has an ability to eavesdrop user's registration request message $\{ID_i, RPW_i\}$ and steal user's $SC$. Furthermore, $\mathcal{H}$ is able to collect some datas, for example, $\{B_i, C_i, D_i, V_i, P_i\}$. Thus $\mathcal{H}$ can masquerade as server spoofing attack. Now, we will explain below.

**Step 1:** Firstly, $\mathcal{H}$computes $h(PSK) = B_i \oplus C_i$ and $\mathcal{H}$ user's login request message $\{AID_i, M_1, M_2, B_i, D_i, T_i\}$.

**Step 2:** Secondly, $\mathcal{H}$calculates $N_1 = RPW_i \oplus M_1 \oplus h(PSK)$ and chooses an arbitrary number $N_2^E$.

**Step 3:** Next $\mathcal{H}$further calculates $M_3^E = N_2^E \oplus h(AID_i \| N_1) \oplus h(PSK)$ and $M_4^E = h(SID_j \| N_2^E \| AID_i)$.

**Step 4:** Finally $\mathcal{H}$issues his/her authentication request message $(SID_j, M_3^E, M_4^E)$ to $U_i$ over a public channel.

Furthermore, this fake authentication request message is successfully verified. Particularly, $\mathcal{H}$ is treated as server $S_j$ by $U_i$ without any doubt. Therefore, Wang et al.'s protocol can't resist the server spoofing attack.

## 5. OUR PROPOSED PROTOCOL

Wang et al.'s protocol includes six phases: initialization phase, server enrollment phase, user enrollment phase, login phase, authentication and key agreement phase and password change phase.

### Initialization phase

To boot up the system, $RC$ selects a generator $P$ of elliptic curve and chooses a secret key $y$ as the system parameter.

### 5.1 Server enrollment phase

In this phase, server enrolls itself at the registration center *RC*. Server selects its own identity $SID_j$ and sends $\{SID_j\}$ to *RC* through open channel. When the request message is received by *RC* from the server. *RC* transmits the information $\{PSK, s\}$ to the server through secure channel as shown in figure 1.

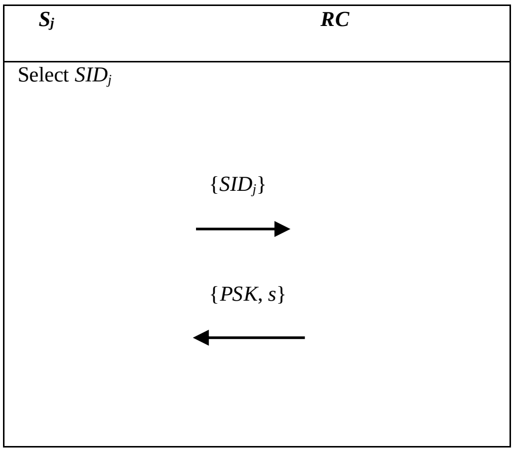


**Fig.1 Server Enrollment Phase of the Proposed Protocol**

**5.2 User enrollment phase**

First, user selects his/her identity $ID_i$, imprints $BIO_i$ and calculates $RPW_i = h(PW_i \| R_i)$ and forwards the message $\{ID_i, RPW_i\}$ to *RC* through open channel.

When a request message is received form the user then *RC* chooses an arbitrarily number $v_i$ and evaluates $A_i = h(ID_i \| s)$, $B_i = h(PSK) \oplus v_i$, $C_i = ID_i \oplus h(PSK \| v_i)$, and $V_i = h(ID_i \| RPW_i)$, where $T_r$ is registration time. Now *RC* inserts all information $\{A_i, B_i, C_i, V_i, h(\cdot)\}$ into *SC* and forwards $\{SC\}$ to the user. After receiving the message $\{SC\}$ from *RC*, user calculates $E_i = B_i \oplus h(R_i)$ and exchanges $B_i$ with $E_i$ and stores $P_i$ into *SC* as shown in figure 2.

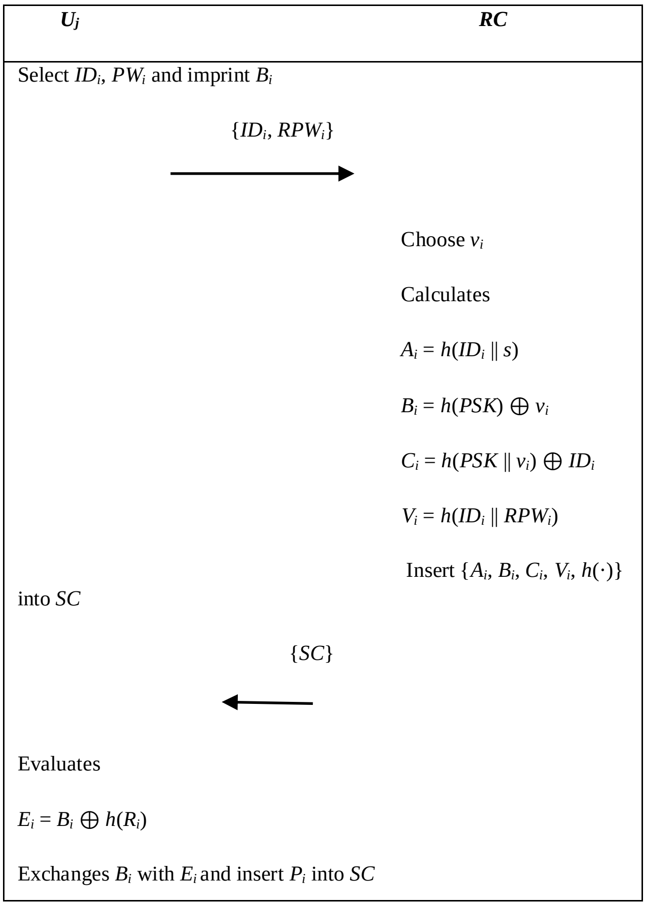

**Fig.2 User Enrollment Phase of the Proposed Protocol**

### 5.3 Login phase

User embeds *SC* and enters $ID_j$, $PW_i$ and imprints $B_i$. Now *SC* evaluates $RPW_i = h(PW_i \| R_i)$ and checks whether $h(ID_i \| RPW_i) = V_i$ is valid. If it is valid, *SC* evaluates $K_i = h(SID_j \| (ID_i \oplus C_i))$. *SC* chooses a random number $N_1$ to evaluates $M_1 = K_i \oplus N_1$, $M_2 = ID_i \oplus K_i$, $M_3 = RPW_i \oplus K_i$, $B_i = E_i \oplus h(R_i)$ and $D_i = h(N_1 \| RPW_i \| A_i \| T_i)$ where $T_i$ is an additional timestamps.

Now, user transmits the login message $\{M_1, M_2, M_3, B_i, D_i, T_i\}$ to $S_j$ through open channel as shown in figure 3.

| $U_j$ | *RC* |
|---|---|
| Insert *SC* | |
| Enter $ID_i$, $PW_j$ and imprint $B_i^*$ | |
| Calculate $RPW_i = h(R_i \| PW_i)$ | |
| Check $h(ID_i \| RPW_i) =? V_i$ | |
| Evaluates $K_i = h(SID_j \|(ID_i \oplus C_i))$ | |
| Choose $N_1$ | |
| Compute $M_1 = N_1 \oplus K_i$ | |
| $M_2 = ID_i \oplus K_i$ | |
| $M_3 = RPW_i \oplus K_i$ | |
| $B_i = E_i \oplus h(R_i)$ | |
| $D_i = h(N_1 \| A_i \| RPW_i \| T_i)$ | |
| $\{M_1, M_2, M_3, B_i, D_i, T_i\}$ ⟶ | |

**Fig.3 Login Phase of the Proposed Protocol**

### 5.4 Authentication and key agreement phase

After receiving the login request message $\{M_1, M_2, M_3, B_i, D_i, T_i\}$ from the user, $S_j$ checks $T_i - T_j \leq \Delta T$ and retrieves $v_i = B_i \oplus h(PSK)$, $K_i = h(SID_j \| h(PSK \| v_i))$, $N_1 = K_i \oplus M_1$, $ID_i = K_i \oplus M_2$, $RPW_i = K_i \oplus M_3$ and $A_i = h(ID_i \| s)$ to verify whether $h(N_1 \| RPW_i \| A_i \| T_i) = D_i$ is valid. If this verification is hold, $S_j$ chooses an arbitrary number $N_2$ and evaluates session key $SK_{ij} = h(ID_i \| SID_j \| N_1 \| N_2)$ between $U_i$ and $S_j$. $S_j$ evaluates $M_4 = N_2 \oplus h(A_i \| RPW_i \| N_1)$ and $M_5 = h(SID_j \| N_1 \| N_2 \| ID_i)$ and forwards his/her authentication request message $\{M_4, M_5\}$ to $U_i$ through an insecure channel.

When obtaining $S_j$,s authentication request message $\{M_4, M_5\}$, *SC* retrieves $N_2 = h(A_i \| RPW_i \| N_1) \oplus M_4$ and verifies whether $h(SID_j \| N_1 \| N_2 \| ID_i)$ is consistent with $M_5$. If they are consistent, *SC* evaluates $SK_{ij} = h(ID_i \| SID_j \| N_1\| N_2)$ and $M_6 = h(SK_{ij} \| N_1 \| N_2)$. And then *SC* delivers authentication reply $\{M_6\}$ is valid. If it is valid, $S_j$ adopts this session key $SK_{ij}$ to communicate with $U_i$ in the following communication. Otherwise, authentication will be rejected by $S_j$.

### 5.5 Password change phase

In this phase, user is allowed to modify his/her password easily without interfering with the server. First, user inserts his/her smartcard into a card reader and enters $ID_j$, $PW_i$ and also imprints $B_i$. Now, the smartcard reader evaluates $RPW_i = h(PW_i \| R_i)$ and verifies whether $V_i = h(ID_i \| RPW_i)$ is valid. If the equality does not hold then the connection is ended. Otherwise, the user selects new password $PW_i^{new}$ and evaluates $RPW_i^{new} = h(PW_i^{new} \| R_i)$ and $V_i^{new} = h(ID_i \| RPW_i^{new})$. Finally, *SC* replaces $V_i$ with $V_i^{new}$ in memory of the smartcard.

## 6. SECURITY ANALYSIS

### 6.1 Prevent to replay attack

Our proposed protocol provides security against replay attack because we use timestamps and random numbers. Suppose adversary eavesdrops the login request message $\{M_1, M_2, M_3, B_i, D_i, T_i\}$ of user and send it to the server after some times. After getting the login request message, server verifies the legality of this message by checking timelines of timestamp $T_i$ and correctness of random number $N_1$. Server rejects the request obviously because random number and timestamp are changed every time.

### 6.2 Prevent to password guessing attack

In our proposed protocol, there is no any transmitted message wherein user's password available openly and also we don't save it into *SC* individually. Suppose an adversary steals *SC* of user and extract the all information from *SC* but there is no any computation to check the password legality. So our proposed protocol is secured against password guessing attack as shown in figure 4.

| $U_j$ | | $S_j$ |
|---|---|---|
| | | Check $T_i - T_j \leq \Delta T$ |
| | | Retrieves $v_i = B_i \oplus h(PSK)$ |
| | | $K_i = h(SID_j \parallel h(PSK \parallel v_i))$ |
| | | $N_1 = K_i \oplus M_1$ |
| | | $ID_i = K_i \oplus M_2$ |
| | | $RPW_i = K_i \oplus M_3$ |
| | | $A_i = h(ID_i \parallel s)$ |
| | | Check $h(N_1 \parallel RPW_i \parallel A_i \parallel T_i) = D_i$ |
| | | Choose $N_2$ and Evaluate |
| | | $SK_{ij} = h(ID_i \parallel SID_j \parallel N_1 \parallel N_2)$ |
| | | $M_4 = N_2 \oplus h(A_i \parallel RPW_i \parallel N_1)$ |
| | | $M_5 = h(SID_j \parallel N_1 \parallel N_2 \parallel ID_i)$ |
| | $\{M_4, M_5\}$ ← | |
| Retrieves $N_2 = h(A_i \parallel RPW_i \parallel N_1) \oplus M_4$ | | |
| Checks $h(SID_j \parallel N_1 \parallel N_2 \parallel ID_i) = M_5$ | | |
| $SK_{ij} = h(ID_i \parallel SID_j \parallel N_1 \parallel N_2)$ | | |
| $M_6 = h(SK_{ij} \parallel N_1 \parallel N_2)$ | | |
| | $\{M_6\}$ → | |
| | | Check $h(SK_{ij} \parallel N_1 \parallel N_2) = M_6$ |

**Fig.4 Authentication and Key Agreement Phase of the Proposed Protocol**

### 6.3 Prevent to server spoofing attack

In our proposed protocol, under the assumption that adversary who is a malicious insider but isn't another server is able to steal user's smart card and eavesdrop his/her registration request message $\{ID_i, RPW_i\}$. Adversary tries to masquerade as server to spoof user by collecting the sensitive datas $A_i$, $B_i$, $C_i$, $V_i$, $P_i$. But it is hard to retrieve $h(PSK)$ so that adversary is unable to be authenticated by user successfully. He cannot acquire the random number $N_1$ and valid authentication request message $\{M_4, M_5\}$. Thus, adversary attempt fail. Therefore, our protocol prevents the server spoofing attack.

### 6.4 Prevent to user impersonation attack

Under the user impersonation attack, adversary who is an outside hacker tries to impersonate user without the password $PW_i$ or biometric information $BIO_i$. In the proposed scheme, adversary is unable to acquire $h(PSK)$ even if he eavesdrops user's previous login request message $\{M_1, M_2, M_3, B_i, D_i, T_i\}$ and extracts user's sensitive datas from smart card by SPA or DPA. Thus, adversary cannot retrieve the random numbers $N_1$, $N_2$ or session key $SK_{ij}$. Therefore, our protocol is secure against the user impersonation attack.

## 7. SECURITY AND PERFORMANCE COMPARISON

In this section, we describe the security and performance comparison along with Wang et al.'s protocol [3]. Some notations are described as: $\boldsymbol{T_H}$ indicates one way hash function, $\boldsymbol{T_{PM}}$ indicates scalar point multiplication and $\boldsymbol{T_S}$ indicates symmetric decryption/encryption functions as shown in Table 2.

Table 2 shows the comparison of the computation cost of the proposed protocol with Wang et al.'s protocol [3]. Wang et al.'s protocol needs to perform total 17 hash functions. On the other hand, our proposed protocol needs to perform 15 hash functions. According to Table 2, the computation overhead of our proposed protocol and Wang et al.'s protocol are almost same, the only change is the reduction of 2 hash function in our proposed protocol. Nevertheless, our protocol is secure against the attacks to which Wang et al.'s protocol is not resistant.

**Table 2 Comparison of Computation Cost**

| | **Wang et al. [3]** | **Our protocol** |
|---|---|---|
| Computation cost of registration phase | $6T_H$ | $4T_H$ |
| Computation cost of login and authentication phase | $11T_H$ | $11T_H$ |
| Total computation cost | $17T_H$ | $15T_H$ |

Table 3 shows the comparison of the security features of the proposed protocol with Wang et al.'s protocol [3]. As shown in Table 3, our protocol provides security against user impersonation attack and server spoofing attack but Wang et al.'s protocol doesn't provide security against above vulnerabilities. Therefore, our proposed protocol is more efficient and secure than Wang et al.'s protocol [3].

**Table 3: Comparison of Security Features**

| **Attacks** | **Wang et al. [3]** | **Our protocol** |
|---|---|---|
| Prevents user impersonation attack | No | Yes |
| Prevents server spoofing attack | No | Yes |

## 8. CONCLUSION

In this paper, we have analyzed Wang et al.'s protocol entitled "cryptanalysis and improvement of a biometric based multi-server authentication and key agreement scheme". We have found that their protocol is vulnerable to user impersonation attack and server spoofing attack. To reduce these weaknesses, we have proposed an improved biometric-based multi-server authentication and key agreement scheme. Our proposed protocol satisfies all securities perception given above. Our proposed protocol is powerful than Wang et al.'s scheme, and there is no extra computation needed in our scheme. In future work, we will propose a lightweight scheme for biometric multi-server environment with low computation cost and better security.

**Acknowledgements:**

The first author gratefully acknowledges the financial support received from CSIR (India) in the form of Junior Research Fellowship CSIR award no. 09/113(0020)/2018-EMR-I.

## REFERENCES

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

[2] Mishra D., Das A.K. and Mukhopadhyay S. (2014). A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards, Experts Systems with applications, 41(18), 8129-8143.

[3] Wang C., Zhang X., Zheng Z.(2016). Cryptanalysis and improvement of a biometric-based multi-server authentication and key agreement scheme, PLoS ONE, 11(2): e0149173. doi:10.1371/journal.pone.0149173