Cryptanalysis and Improvement of Three-Factor-Based Confidentiality-Preserving Remote User Authentication Scheme in Multi-server Environment

Subhas Barman, Prantik Guha, Rituparna Saha and Soumil Ghosh

Abstract Lately, Ali–Pal addressed an improvement to Guo–Wen's scheme which proclaims to protect the anonymity of the user during remote authentication in a multi-server environment. But, the cryptanalysis of their scheme finds leakage of some sensitive information. Even, the scheme is not resilient to insider attack. In this paper, we address the problems and attempt to improve the security of the scheme. In addition, security of the proposed scheme is analyzed with the pi-calculus-based formal verification tool ProVerif. The proposed scheme is compared with other existing key exchange protocols reported in the literature with respect to computation and communication costs. We also prove that our proposed scheme provides mutual authentication and it is secured against various well-known attacks.

Keywords Multi-server environment · Remote authentication · ProVerif · Key exchange protocol

1 Introduction

In the era of modern technology, biometrics are used to either generate [1–3] or exchange a cryptographic key [4–6] for better network security. Now a user can access remote servers through a smart card in a public channel. Smart cards which contain the biometric data are vulnerable towards common attacks like stolen smart card, password update in the server without a secure channel, privileged insider attack, user impersonation attack, replay attack, and also offline password guessing attack. Multi-server environment provides a better solution as the user can communicate with any server by doing one-time registration. Mishra et al. [7] provided a secure and resilient scheme for a multi-server environment which was developed to deal with the user and server impersonation attack and stolen smart card attack of the previously available schemes. Later, Lu et al. [8] addressed the drawbacks of

S. Barman (✉) · P. Guha · R. Saha · S. Ghosh
Jalpaiguri Government Engineering College, Jalpaiguri 735102, West Bengal, India
e-mail: subhas.barman@gmail.com; bsubhas1980@gmail.com

M. Chakraborty et al. (eds.), *Proceedings of International Ethical Hacking Conference 2018*, Advances in Intelligent Systems and Computing 811,
https://doi.org/10.1007/978-981-13-1544-2_7

Mishra et al.'s scheme like forgery and server masquerading and lacks password forward secrecy. So they projected a scheme to overcome the issues. Lu et al. also pointed out the vulnerability towards replay attack and also incapability of password update phase. Then, they developed an improved authentication scheme. Thereafter, Chaudhry [9] found that Lu et al.'s scheme [8] prone to impersonation attack and is not facilitated to user anonymity. Similarly, Guo–Wen [10] proposed a more reliable and robust authentication protocol in a multi-server environment. However, this scheme is prone to fall prey to password and identity guessing attack, new smart card issue threat, user impersonation threat, known session key temporary information attack, and insider attack. To overcome these problems, Ali–Pal [11] came up with an enhanced and resilient three-factor-based confidentiality-preserving remote authentication scheme in multi-server environment. They addressed the pitfalls of previous schemes like new smart card issue attack, identity guessing attack, and known session key attack. Computation cost and estimated time are minimized in [11]. However, we find that there are still some threats to this scheme. For example, one of the random nonce can be computed from intercepted message. Moreover, Ali–Pal's scheme [11] is also vulnerable to insider attack.

We propose an improved scheme to surmount these drawbacks. Our scheme provides mutual authentication in multi-server environment. Moreover, in this paper, we use pi-calculus [12]-based formal verification tool ProVerif [13] to prove authentication and security of the proposed protocol.

2 Our Contribution

- We cryptanalysis of the Ali–Pal's scheme [11].
- We improved the scheme [11] to overcome the drawback and also add some new features. We simulate our scheme for the formal security analysis using ProVerif tool and show that proposed scheme is protected from different security attacks.
- We also compare communication cost and performance of the proposed scheme with other existing schemes.

3 Literature Review

Ali–Pal [11] addressed an improvement to Guo–Wen's [10] scheme. In this section, we reviewed the Ali–Pal's scheme [11]. The symbols and its meanings are given in Table 1.

Table 1 Meaning of notations

Notation	Description
RC	Registration center
ID_i, PW, BIO_i	Identity, password, biometrics of ith user U_i
$\mathcal{A}$	An attacker/adversary
SID_j, e_j, d_j	Identity, public key, private key of jth server S_j
h(.), H(.)	Hash function and Biohashing function
$\|\|$, $\oplus$	Concatenation and XOR operation
SK	Session key shared between U_i and S_j

3.1 Server Registration

S_j selects own identity SID_j and sends it to RC via a trustworthy channel. Then RC computes $X_j = h(d_j||SID_j)$ and transfers X_j, d_j to the server S_j via secure channel.

3.2 User Registration

In this phase, U_i selects own identity ID_i, password PW_i and imprints biometric BIO_i and then calculates $RPW_i = h(PW_i||H(BIO_i))$. Then U_i transfers ID_i, RPW_i to RC via trustworthy channel. Then RC computes $A_i = X_j \oplus h(RPW_i)$, $B_i = h(ID_i||RPW_i||X_j)$ and issues a smart card holding parameters $\langle A_i, B_i, h(.), H(.)\rangle$. RC finally sends smart card to U_i via secure channel.

3.3 Login and Authentication

U_i inserts smart card into a smart card reader and inputs own ID_i, password PW_i and imprints biometric BIO_i. Then smart card calculates $RPW_i = h(PW_i||H(BIO_i))$, $F_i = h(ID_i)$, $X_j' = A_i \oplus h(RPW_i)$ and $B_i' = h(ID_i||RPW_i||X_j')$. Now, if $B_i' = B_i$, U_i chooses a random nonce R_1 and computes $RPW_{ij} = h(RPW_i||SID_j)$, $M_1 = (F_i||R_1||SID_j)_j^e \mod n_j$, $M_2 = R_1 \oplus RPW_{ij} \oplus ID_i$ and $M_3 = h(RPW_{ij}||F_i||X_j||R_1)$ and then M_1, M_2, M_3 is sent to S_j via a public channel. After getting the login message M_1, M_2, M_3 from U_i, S_j decrypts $(F_i||R_1||SID_j) = M_1^{d_j} \mod n_j$ and computes $RPW_{ij}' = M_2 \oplus R_1 \oplus F_i$ and $M_3' = h(RPW_{ij}||F_i||X_j||SID_j)$ and compares with M_3. If M_3' is equals to M_3, then S_j believes that U_i is legal; otherwise, the session is expired. Now, S_j Selects a random nonce R_2 and computes $M_4 = h(RPW_{ij}||R_1) \oplus R_2$, $SK = h(R_1||R_2||X_j||ID_i)$ and $M_5 = h(RPW_{ij}||SK)$ and transmits M_4, M_5 via a public channel. After receiving M_4, M_5, U_i computes $R_2' = M_4 \oplus h(RPW_{ij}||R_1)$, $SK' = h(R_1||R_2'||X_j||F_i)$ and $M_5' = h(RPW_{ij}||SK)$. If M_5' is not equals to M_5, then session is

terminated. Otherwise, U_i believes on the legitimacy of S_j and mutual authentication holds.

4 Cryptanalysis of Ali–Pal's Scheme

In the Ali–Pal's scheme, ID_i, $H(BIO_i)$ are fixed for every communication initiated by U_i to any server S_k. X_j, R_1^j are varied for different servers (i.e., S_j and $j = 1, 2, ...$ and $S_k \neq S_j$). From received message, S_j can extract ID_i, R_1^j from message M_1^j and subsequently, S_j can computes $H(BIO_i)$ from message M_2^k, that is, $H(BIO_i = M_2^j \oplus R_1^j \oplus ID_i$. Now, S_j can act as an insider attacker and may try to know some information for communication of U_i with other server S_k. Attacker S_j intercepts login messages (say, M_2^k) from public channels. From the knowledge ID_i, $H(BIO_i)$ and publicly shared message M_2^k (shared by U_i with S_k), attacker can reveal R_1^k from M_2^k, that is, $R_1^k = M_2^k \oplus ID_i \oplus H(BIO_i)$. Moreover, Ali–Pal's does not consider the biometrics change phase.

5 Proposed Scheme

We present a three-factor-based authentication protocol. This can be used in multi-server environment. Our scheme consists of five phases (i) system setup, (ii) registration, (iii) login and authentication, (iv) password change, and (v) biometrics change.

5.1 System Setup

In this phase, the system setup is carried out following the similar process of Ali–Pal's scheme [11]. The detailed description is given below. Step 1: Registration center *RC* selects two large prime numbers, i.e., p_j and q_j for m servers where j=1 to m. After that *RC* computes $n_j = p_j \times q_j$, where $pj \neq q_j$.
Step 2: *RC* chooses $1 < e_j < \phi(n_j)$ where $\phi(n_j) = (p_j - 1)x(q_j - 1)$ and calculates d_j. Where $d_j = e_j - 1 \bmod \phi(n_j)$ and issues (e_1, n_1), (e_2, n_2), …, (e_m, n_m) as public key and $d_1, d_2, \ldots, d_m$ as private key.

5.2 Registration

This phase consists of two phases server and user registration.

1. Server Registration: Step 1. Server selects own identity SID_j and computes $C_j = h(SID_j)$. C_j is transferred to RC via secure channel.
   Step 2. After getting C_j, RC computes $X_j = h(d_j||C_j)$. Now RC transmits X_j, d_j to server via a secure channel.
2. User Registration: An user U_i can register by the following ways: Step 1. U_i selects an ID_i, password PW_i and imprints biometric BIO_i. Then U_i computes $RPW_i = h(PW_i||H(BIO_i)$ and $F_i = h(ID_i)$ and transfers F_i, RPW_i to RC via a secure channel.
   Step 2. Upon receiving F_i, RPW_i, RC computes $A_i = X_j \oplus h(RPW_i)$, $B_i = h(F_i||RPW_i||X_j)$ and stores the values A_i, B_i, $h(.)$, $H(.)$ into a smart card. Finally, RC transmits smart card to U_i via a secure channel. We elaborate the user registration phase in Table 2.

5.3 Login and Authentication

User U_i is authenticated by the smart card reader to access the remote server. Then the smart card reader sends a login message to the server S_j via a public channel. This phase is given Fig. 1. Detailed description is given below.

Step 1. U_i inserts smart card into a smart card reader and inputs own ID_i, password PW_i and imprints biometric BIO_i.Then smart card calculates $RPW_i = h(PW_i||H(BIO_i))$, $F_i = h(ID_i)$, $X_j' = A_i \oplus h(RPW_i)$ and $B_i' = h(ID_i||RPW_i||X_j')$. Now, smart card compares B_i' with B_i. If B_i' is not equals to B_i, then U_i is rejected.

Step 2. Otherwise, U_i chooses a random nonce R_1 and computes $RPW_{ij} = h(RPW_i||SID_j)$, $M_1 = (F_i||R_1||SID_j)_j^e \bmod n_j$, $M_2 = R_1 \oplus RPW_{ij} \oplus ID_i$ and $M_3 = h(RPW_{ij}||F_i||X_j||R_1)$ and then M_1, M_2, M_3 is sent to S_j via a public channel.

Step 3. After getting the login message M_1, M_2, M_3 from U_i, S_j decrypts $(F_i||R_1||SID_j) = M_1^{d_j} \bmod n_j$ and computes $RPW_{ij}' = M_2 \oplus R_1 \oplus F_i$ and $M_3' = h(RPW_{ij}||F_i||X_j||SID_j)$ and compares with M_3. If M_3' is equals to M_3, then S_j believes that U_i is legal otherwise the session is expired.

Step 4. Now, S_j Selects a random nonce R_2 and computes $M_4 = h(RPW_{ij}||R_1) \oplus R_2$, $SK = h(R_1||R_2||X_j||ID_i)$ and $M_5 = h(RPW_{ij}||SK)$ and transmits M_4, M_5 via a public channel.

Step 5. After receiving M_4, M_5, U_i computes $R_2' = M_4 \oplus h(RPW_{ij}||R_1)$, $SK' = h(R_1||R_2'||X_j||F_i)$ and $M_5' = h(RPW_{ij}||SK)$. If M_5' is not equals to M_5, then session is terminated. Otherwise, U_i believes on the legitimacy of S_j and mutual authentication holds.

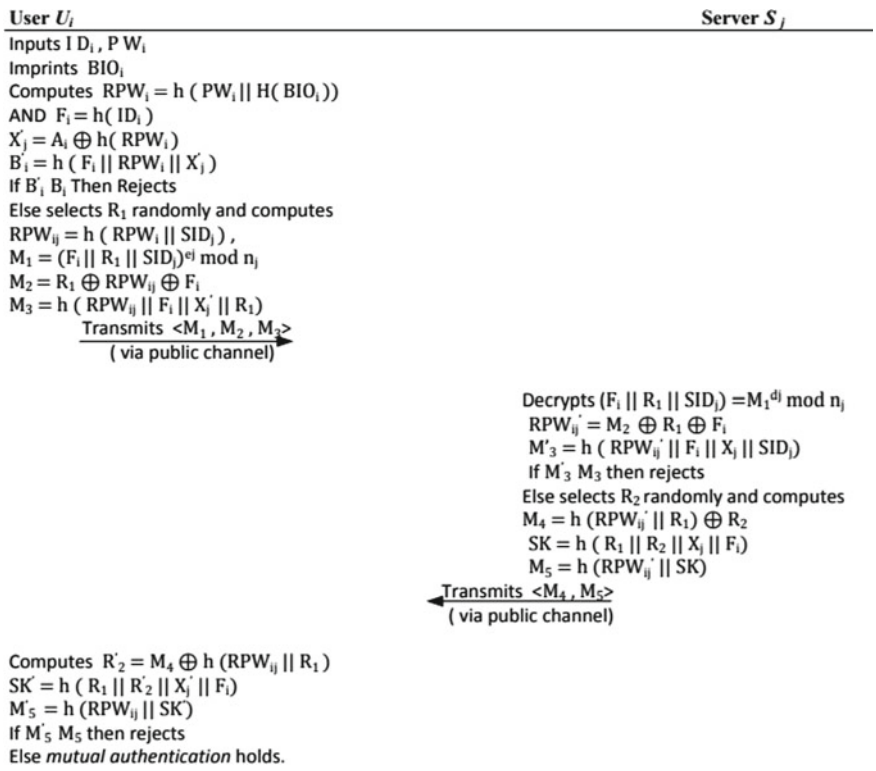

Fig. 1 Login and authentication protocol

5.4 Password Change

Changing of password in a varied time interval is a good habit which incurs the security. If the user wants to change his/her password, he/she can easily do that through simple steps.

Step 1. U_i touches own smart card into a terminal, input his/her ID_i and PW_i and also imprints BIO_i. After that the smart card computes RPW_i'=$h(PW_i||H(BIO_i))$, $X_j' = A_i \oplus h(RPW_i')$, $B_i' = h(h(ID_i)||RPW_i'||X_j')$. Then it checks B_i' is equals to B_i or not. If it is true, it means that the input for the user U_i 's identification is authorized that means U_i is a authorized user for that smart card and then smart card allows the user U_i to change his/her password and asks to input new password PW_i^{new}. Otherwise rejects.

Step 2. Now smart card calculates $RPW_i^{new} = h(PW_i^{new}||H(BIO_i))$, $A_i^{new} = X_j' \oplus h(RPW_i^{new})$, $B_i^{new} = h(h(ID_i)||RPW_i^{new}||X_j')$. Finally, smart card replaces A_i, B_i with the new A_i^{new}, B_i^{new} and stored it into the smart card.

5.5 Biometric Change

Suppose any user wants to update the biometric data. Then, authentication of the user is done by the same way as discussed in the password change phase. If the authentication holds, smart card allows the user U_i to change his/her biometric with the new biometric and asks to input new biometric BIO_i^{new}. Otherwise rejects. Now, smart card calculates $RPW_i^{new} = h(PW_i||H(BIO_i^{new}))$, $A_i^{new} = X_j' \oplus h(RPW_i^{new})$, $B_i^{new} = h(h(ID_i)||RPW_i^{new}||X_j')$. Finally, smart card replaces A_i, B_i with the new A_i^{new}, B_i^{new} and stored it into the smart card.

6 Security Analysis

The security of the proposed scheme is analyzed with formal as well as informal security analysis. We have verified our proposed protocol using ProVerif simulator.

6.1 Formal Security Analysis

In order to prove the security of cryptographic protocols, ProVerif is a widely used formal verification tool [9, 12, 13]. In this section, we prove secrecy and authentication using ProVerif, because it is performed automatically and efficiently and can detect errors easily. ProVerif makes use of Dolev–Yao model [14] and supports many cryptographic primitives, including digital signature, symmetric and asymmetric encryption, hash function.

The user and the server communicate among themselves through a public channel, which is defined as below:

```
free Ch_Pub:channel.
```

The variables used in the protocol are defined as follows:

```
free IDi:bitstring.
free PWi:bitstring.
free BIOi:bitstring [private].
free RPWi:bitstring [private].
const dj:bitstring[private].
const nj:bitstring.
const ej:bitstring.
free SIDj:bitstring [private].
free Ai:bitstring [private].
free Bi:bitstring [private].
free Xj:bitstring [private].
free SK:bitstring [private].
free SK':bitstring [private].
```

The functions (xor(), exp(), mod(), mult(), and concat() represent exclusive-OR, exponent function, modulo operation, scalar multiplication, and string concatenation, respectively) used in the protocol are defined as follows:

```
fun h(bitstring):bitstring.
fun H(bitstring):bitstring.
fun xor(bitstring,bitstring):bitstring.
fun mod(bitstring,bitstring):bitstring.
fun exp(bitstring,bitstring):bitstring.
fun mult(bitstring,bitstring):bitstring.
fun concat(bitstring,bitstring):bitstring.
```

The algebraic properties of the functions are defined as below:

```
equation for all a:bitstring,b:bitstring; xor(xor(a,b),b)=a.
```

According to the protocol, the user Ui computes and sends $M1$, $M2$, $M3$ to the Server Sj and then waits until he receives $M4$, $M5$ from the Server Sj. So, the user Ui is defined as follows:

```
let UserUi=
let RPWi=h(concat(PWi,H(BIOi))) in
let Fi=h(IDi) in
let Xj'=xor(Ai,h(RPWi)) in
let Bi'=h(concat(Fi,concat(RPWi,Xj))) in
if (Bi'=Bi) then
new R1:bitstring;
let RPWij=h(concat(RPWi,SIDj)) in
let M1=mod(exp(concat(Fi,concat(R1,SIDj)),ej),nj) in
let M2=xor(R1,xor(RPWij,Fi)) in
let M3=h(concat(RPWij,concat(Fi,concat(Xj',R1)))) in
out(Ch_Pub,(M1,M2,M3));
in(Ch_Pub,(xM4:bitstring,xM5:bitstring));
let R2'= xor(xM4,h(concat(RPWij,R1))) in
let SK'=h(concat(R1,concat(R2',concat(Xj',Fi)))) in
let M5'=h(concat(RPWij,SK')) in
if (M5'=xM5) then 0.
```

According to the protocol, the server Sj receives `{M1,M2,M3}` from the user Ui, then computes and sends `{M4,M5}` to the user Ui. We can define the server Sj as follows:

```
let ServerSj=
let Xj=h(concat(dj,SIDj)) in
in(Ch_Pub,(xM1:bitstring,xM2:bitstring,xM3:bitstring));
new R1':bitstring;
new Fi':bitstring;
let RPWij'=xor(xM2,xor(R1',Fi')) in
let M3'=h(concat(RPWij',concat(Fi',concat(Xj,SIDj)))) in
if (xM3=M3') then
new R2:bitstring;
let M4=xor(h(concat(RPWij',R1')),R2) in
```

```
let SK=h(concat(R1',concat(R2,concat(Xj,Fi')))) in
let M5=h(concat(RPWij',SK)) in
out(Ch_Pub,(M4,M5))
else 0.
```

In order to ensure mutual authentication, we define events as follows:

```
event begin_UserUi(bitstring).
event end_UserUi(bitstring).
event begin_ServerSj(bitstring).
event end_ServerSj(bitstring).
```

The process can be defined by:

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

To verify mutual authentication and session key's security, we define the following queries:

```
query attacker(SK).
query attacker(SK').
query id:bitstring; event(end_UserUi(id)) ==> event(begin_UserUi(id)).
query id:bitstring; event(end_ServerSj(id)) ==> event(begin_ServerSj(id)).
```

When the above code is performed in ProVerif, we find that both the correspondence queries are true and both the (not)attacker queries are true, thus indicating that both the mutual authentication property and session key security are satisfied for our proposed scheme.

6.2 Informal Security Analysis

In this section, we elaborate informal security analysis of our scheme and prove that our protocol is able to protect from different types of security vulnerabilities.

1. Password and identity guessing attack: We assume an adversary can eavesdrop all communication messages M_1, M_2, M_3, M_4, M_5 and extract all information A_i, B_i from smart card. But still $\mathcal{A}$ is not able to calculate PW_i and ID_i from A_i. $A_i = X_j \oplus h(RPW_i)$, where $X_j = h(SID_j||d_j)$ and $RPW_i = h(PW_i||H(BIO_i))$. To calculate PW_i, $\mathcal{A}$ needs to know BIO_i, SID_j, d_j at one time which is infeasible in polynomial time. $\mathcal{A}$ is not able to calculate PW_i from $B_i = (F_i||RPW_i||X_j)$, where $F_i = h(ID_i)$. For computing PW_i, $\mathcal{A}$ has to know the parameters $H(BIO_i)$, ID_i, SID_j, d_j at one time which is impossible. $\mathcal{A}$ also cannot evaluate ID_i from B_i. $\mathcal{A}$ cannot obtain ID_i from M_1, M_2, M_3, and M_5 because of hash function where $M_1 = (F_i||R_1||SID_j)^{e_j} \bmod n_j$, $M_2 = R_1 \oplus RPW_{ij} \oplus F_i$, $M_3 = h(RPW_{ij}||F_i||X_j||R_1)$ $M_5 = h(RPW_{ij}||SK)$, $RPW_{ij} = h(RPW_i||SID_j)$ and $SK = h(R_1||R_2||X_j||F_i)$.
2. Impersonation attack: We assume an attacker $\mathcal{A}$ intercepts all communication messages, and then he modifies all messages and tries to imitate as a legal server or user. But, in our protocol it is not possible for some reasons like $\mathcal{A}$ cannot calcu-

late $M_1 = (F_i||R_1||SID_j)_j^e \bmod n_j$ where R_1 is a random nonce because A is unable to obtain ID_i. $M_2 = R_1 \oplus RPW_{ij} \oplus F_i$ and $M_3 = h(RPW_{ij}||F_i||X_j||R_1)$ where $RPW_{ij} = h(RPW_i||SID_j)$ and $RPW_i = h(PW_i||H(BIO_i))$. To compute M_2, $\mathcal{A}$ has to know BIO_i, ID_i, PW_i, and SID_j at same time, which is not possible. For calculating M_3, $\mathcal{A}$ has to know PW_i, ID_i, BIO_i and X_j which is not feasible. $M_4 = h(RPW_{ij}||R_1) \oplus R_2$ and $M_5 = h(RPW_{ij}||SK)$ where $SK = h(R_1||R_2||X_j||F_i)$. So, $\mathcal{A}$ has to know PW_i, BIO_i, SID_j, X_j, ID_i at one time to calculate M_4 and M_5 which is not possible.

3. User untraceability attack: In this type of threat, an attacker intercepts two communication messages and tries to extract identity of user or server by matching values of each parameter. But, our protocol is able to protect this type of attack. In $M_1 = (F_i||R_1||SID_j)_j^e \bmod n_j$, user ID_i is secured using hash function and R_1 is a random nonce. So, value of M_1 is different in each session due to uniqueness property of R_1. $M_2 = R_1 \oplus RPW_{ij} \oplus F_i$ and $M_3 = h(RPW_{ij}||F_i||X_j||R_1)$ are also different in each session due to uniqueness property of R_1. Therefore, our protocol resists user untraceability attack.
4. Replay attack: Our protocol resists replay attack by using random nonce R_1 and R_2.
5. Insider attack: Our scheme is not vulnerable to insider attack because user U_i sends $RPW_i = h(PW_i||H(BIO_i))$ to RC. So, an insider of system cannot obtain

Table 2 Security features comparison

SF	Guo–Wen [10]	He–Wang [15]	Wen et al. [16]	Li et al. [17]	Irshad et al. [18]	Ali–Pal [11]	PS
A1	Yes	Yes	Yes	No	Yes	Yes	Yes
A2	Yes	No	No	No	No	No	Yes
A3	Yes	No	Yes	No	No	Yes	Yes
A4	Yes	No	Yes	Yes	No	Yes	Yes
A5	No	Yes	Yes	Yes	Yes	Yes	Yes
A6	No	No	No	Yes	Yes	Yes	Yes
A7	Yes	Yes	No	No	Yes	No	Yes
A8	No	Yes	Yes	Yes	Yes	No	Yes
A9	No	Yes	Yes	Yes	No	Yes	Yes
A10	Yes	Yes	Yes	Yes	Yes	Yes	Yes
A12	Yes	Yes	Yes	Yes	Yes	Yes	Yes
A13	Yes	Yes	Yes	Yes	Yes	Yes	Yes

SF security features, PS proposed scheme, A1 be proof against password guessing attack, A2 facilitating user anonymity, A3 be proof against user impersonation attack, A4 be proof against server impersonation attack, A5 be proof against replay attack, A6 be proof against session key temporary information attack, A7 be proof against user untraceability attack, A8 be proof against privileged insider attack, A9 be proof against identity guessing attack, A10 forward secrecy, A12 be proof against smart card theft attack, A13 session key verification

password because of hash function. Though attacker guesses PW_i but still he/she is unable to validate password without knowledge of biometrics BIO_i.

6. Known session key temporary information attack: In our scheme, attacker cannot compute session key $SK = h(R_1||R_2||X_j||F_i)$ with the knowledge of random nonce R_1 and R_2. Because, SK also depends on X_j and F_i.
7. Smart card stolen attack: Suppose attacker gets the smart card of an user and extracts parameters $A_i = X_j \oplus h(RPWi)$ and $B_i = (F_i||RPW_i||X_j)$, where $RPW_i = h(PW_i||H(BIO_i))$, $X_j = h(SID_j||d_j)$ and $F_i = h(ID_i)$. But, attacker is unable to calculate ID_i from F_i and PW_i from A_i, B_i.
8. Forward secrecy: Our scheme facilitates forward secrecy property. With the knowledge of a session key, attacker is not able to compute other session key.

We compare our scheme along with other schemes with respect to different security attacks and given in Table 2.

7 Performance

In this section, we compare our scheme with other existing schemes based on communication cost and estimated time.

7.1 Communication Cost

In Table 3, we represent comparison of communication cost of our scheme with respect to other existing schemes. Here, we assume lengths of ID_i, PW_i, random nonce and hash functions are 160 bits. e_j, d_j are 1024 bits and symmetric encryption, decryption is of 512 bits for each.

7.2 Estimated Time

To calculate estimated time, we have used the following notations, *Th*: time complexity of hash function, *Ts*: symmetric encryption or decryption, *Te*: modular exponentiation, and *Tm*: point multiplication of elliptic curve. We calculate estimated time in seconds. The time complexity of our scheme is $(28\text{Th} + 2\text{Te}) = 28 \times 0.0005 + 2 \times 0.522 = 1.058$. The comparison of computation time of our scheme with other scheme is given in Table 3.

Table 3 Estimated time comparison and communication cost

PC	Li et al. [17]	Ali–Pal [11]	Irshad et al. [18]	He–Wang [15]	Wen et al. [16]	PS
CCRP	8Th	4Th	5Th + 1Ts + 1Tm	3Th	4Th	4Th
CCLAP	20Th + 4Te	16Th + 2Te	17Th + 5Ts + 10Tm	23Th + 8Tm	12Th + 10Ts	16Th + 2Te
CCPCP	6Th	10Th	7Th + 1Ts + 1Tm	2Th	4Th	8Th
TCC	34Th + 4Te	30Th + 2Te	29Th + 7Ts + 12Tm	28Th + 8Tm	20Th + 10Ts	28Th + 2Te
ET	2.105	1.059	0.8232	0.5186	0.097	1.058
CC	2688	1664	2784	3360	4032	1664

PC performance comparison, *PS* proposed scheme, *CC* communication cost, *CCRP* computation cost of registration phase, *CCLAP* computation cost of login and authentication phase, *CCPCP* computation cost of password change phase, *TCC* total computation cost, *ET* estimated time

8 Conclusion

In this paper, we found some faults of Ali–Pal's scheme and overcome the drawbacks of the same scheme. We use ProVerif to verify the security of our scheme. Communication cost and estimated time of our scheme are comparatively better than other schemes. In our scheme, a legal user can change his/her password and biometrics without help of server's involvement.

References

1. Barman, S., Chattopadhyay, S., Samanta, D.: Fingerprint based symmetric cryptography. In: 2014 International Conference on High Performance Computing and Applications (ICHPCA), Bhubaneswar, pp. 1–6 (2014). https://doi.org/10.1109/ICHPCA.2014.7045306
2. Barman, S., Samanta, D., Chattopadhyay, S.: Approach to cryptographic key generation from fingerprint biometrics. Int. J. Biom. **7**(3), 226–248 (2015)
3. Barman, S., Samanta, D., Chattopadhyay, S.: Fingerprint-based crypto-biometric system for network security. EURASIP J. Info. Secur. **3** (2015). https://doi.org/10.1186/s13635-015-0020-1
4. Barman, S., Chattopadhyay, S., Samanta, D.: An approach to cryptographic key exchange using fingerprint. In: Mauri, J.L., Thampi, S.M., Rawat, D.B., Jin, D. (eds.) Security in Computing and Communications. Communications in Computer and Information Science SSCC 2014, vol. 467. Springer, Berlin (2014)
5. Barman, S., Chattopadhyay, S., Samanta, D.: An approach to cryptographic key distribution through fingerprint based key distribution center. In: 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), New Delhi, pp. 1629–1635 (2014). https://doi.org/10.1109/ICACCI.2014.6968299

6. Barman, S., Chattopadhyay, S., Samanta, D., Panchal, G.: A novel secure key-exchange protocol using biometrics of the sender and receiver. Comput. Electr. Eng. **64**, 65–82 (2017)
7. Mishra, D., Das, A.K., Mukhopadhyay, S.: A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards. Exp. Syst. Appl. **41**(18), 8129–8143 (2014)
8. Lu, Y., Li, L., Peng, H., Yang, Y.: A biometrics and smart cards based authentication scheme for multi-server environments. Secur. Commun. Netw. **8**(17), 3219–3228 (2015)
9. Chaudhry, S.A.: A secure biometric based multi-server authentication scheme for social multimedia networks. Multimed. Tools Appl. **75**, 12705 (2016). https://doi.org/10.1007/s11042-015-3194-0
10. Guo, D., Wen, F.: Analysis and improvement of a robust smart card based-authentication scheme for multi-server architecture. Wirel. Pers. Commun. **78**(1), 475–490 (2014)
11. Ali, R., Pal, A.K.: Three-factor-based confidentiality-preserving remote user authentication scheme in multi-server environment. Arab. J. Sci. Eng. **42**(8), 3655–3672 (2017). https://doi.org/10.1007/s13369-017-2665-1
12. Abadi, M., Fournet, C.: Mobile values, new names, and secure communication. In: Proceedings of the 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, pp. 104–115. ACM, New York (2001)
13. Abadi, M., Blanchet, B., Comon-Lundh, H.: Models and proofs of protocol security: a progress report. In: Computer Aided Verification, vol. 5643, pp. 35–49. Springer, Heidelberg (2009)
14. Dolev, D., Yao, A.C.: On the security of public key protocols. IEEE Trans. Inf. Theory **29**(2), 198–208 (1983)
15. He, D., Wang, D.: Robust biometrics-based authentication scheme for multi-server environment. IEEE Syst. J. **9**(3), 816–823 (2015)
16. Wen, F., Susilo, W., Yang, G.: Analysis and improvement on a biometric-based remote user authentication scheme using smart-cards. Wirel. Pers. Commun. **80**(4), 1747–1760 (2015)
17. Li, X., Niu, J., Kumari, S., Liao, J., Liang, W.: An enhancement of a smart card authentication scheme for multi-server architecture. Wirel. Pers. Commun. **80**(1), 175–192 (2015)
18. 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. Multimed. Tools Appl. (2016). https://doi.org/10.1007/s11042-016-3921-1