Wireless Personal Communications (2019) 107:351–372
https://doi.org/10.1007/s11277-019-06280-7

An ECC Based Secure Authentication and Key Exchange Scheme in Multi-server Environment

Ashish Tomar[1] · **Joydip Dhar**[1]

Published online: 25 March 2019

Abstract

For providing strong mutual authentication in a multi-server environment many algorithms have been proposed. Most of the algorithms provide mutual authentication between client and multiple servers by using single control server for registration. In this paper, we consider a scenario, in which client and server belong to the different control server. We have proposed a protocol for providing authentication in the multi-control server environment. In our scheme, for strong authentication, we use user's biometric and registered password value in the authentication process. We also use the concept of elliptic curve cryptography to provide security features in our scheme. Furthermore, Burrows–Abadi–Needham logic has been used for formal security analysis in our work. With informal security analysis, we prove that our scheme is secure against popular security attacks like—denial of service attack, man-in-the-middle attack, replay attack and stolen smart card attack.

Keywords Multi-server architecture · Authentication protocol · Multiple control servers · Smart card · Elliptic curve cryptography · Biometrics · BAN logic

1 Introduction

In the last two decades, internet has become very prominent among the people of the world. In current era most of the services are provided by internet for example—Online banking services, information access, medical related services, shopping through online websites, data transfer and many more. Despite these services internet is vulnerable to different types of security attacks due to this data and user security has become a major concern. The client-server architecture is the basis of these services, in which authentication is a major factor. The process of authentication can be done either in the single server or multi-server environment. But multi-server authentication is preferable over single server authentication. Because, in single server environment for accessing

✉ Ashish Tomar
ashishtomar244@gmail.com

Joydip Dhar
jdhar@iiitm.ac.in

[1] ABV-Indian Institute of Information Technology and Management, Gwalior, Madhya Pradesh 474015, India

different services, clients register separately at different servers [1, 2]. However, in a multi-server environment, there is a single server on which all the service providing servers are registered which is called control server, So clients have to register on a single control server.

In the single server environment, normal authentication based on password could be used as mutual authentication but this is infeasible in a multi-server environment, in this for accessing multiple services user will have to login into many servers separately and remember multiple identities and passwords. There are many authentication algorithms or protocols have been proposed for multi-server environment. These protocols are vulnerable to different kinds of security attacks like—stolen smart card, man in the middle, eavesdropping, replay attack etc.

In the current research era most popular algorithms are based on:

- Only smart card based [3–6].
- Smart card based and biometric based [7–19].

In which smart card and biometric based is bit complex but more secure.

For providing mutual authentication and key exchange, different types of biometric and smart card based algorithms have been evolved for multi-server environment.

Chaung et al. [7] presented a three factor based mutual authentication scheme for multi-server environment. Since it was hash based only, they claimed that their scheme is computationally efficient and free from different security attacks. But later in same year Mishra et al. [8] proved that Chaung et al.'s scheme does not provide security against impersonation attack and stolen verifier attack. They proposed a new biometric based authentication protocol. Baruah et al. [9] did the crypt-analysis of Mishra et al.'s scheme and proved that it does not provide security against impersonation attack, stolen smart card attack and man in the middle attack. Later Wang et al. [10] also crypt-analysed Mishra et al.'s scheme and proved that it is not secured against masquerade attack, replay attack and DoS attack. They also proved that Mishra et al.'s scheme has no provision of perfect forward secrecy and there was no revocation and re-registration facility. They also presented an updated protocol. Recently Reddy et al. [11] proved that Wang et al.'s protocol have no provision of perfect anonymity and it is vulnerable to clock synchronization problem, server impersonation attack and privileged insider attack. They also proposed a new updated biometric based algorithm. Further, few formal methods for crytographic protocal analysis discussed by Meadows [20].

He et al. [13] also proposed a protocol based on biometrics and smart cards, In which he used the concept of fuzzy extractor for taking the biometric values but later Odelu et al. [14] proved that He et al.'s scheme does not have resistance against some kinds of attacks like—impersonation attack, known session specific temporary information attack, wrong password login and also pointed out that there was no provision for revocation and re-registration and there was also drawback in password change phase.

Recently Amin [3] and Wei et al. [4] have crypt-analysed some old schemes which was based on only smart card and proposed new algorithm but later in Pan et al. [21] have proved that Amin's scheme does not resist offline identity guessing attack and offline password guessing with smart card stolen attack.

Some other authentication schemes for multi-server environment have also been proposed recently [15–18, 22, 23].

In multi-server environment, algorithms that are developed so far are based on single control server environment. Recently Gupta et al. [12] presented a hash based authentication scheme for multi-control server environment. In our work, we have done the security weakness analysis of Gupta et al.'s scheme and we have proved that it is susceptible to DoS attack, impersonation attack and stolen smart card attack.

The remaining part of the paper is categorised as follows: Sect. 2 gives the review of existing protocol then we have done security analysis of existing protocol in Sect. 3. In Sect. 4 we have presented our own scheme and then we have done security and performance analysis of our scheme in Sects. 5 and 6 respectively and finally we have concluded our paper in Sect. 7.

2 Review of Existing Protocol

In this section, we review existing Gupta et al.'s scheme. For understanding we have used the Table 1 that represents the notations in the scheme. Gupta et al. proposed a hash-based protocol for providing mutual authentication between client and server in a multi-server environment, in which client and server belong to different registration centers. Their scheme involved four entities i.e. server (S_j), user (U_i), registration center (RC_a) and registration center (RC_b). Registration center is the entity for the registration of users and servers. Their protocol has four phases i.e. authentication between registration centers, registration of user and server with RC, mutual authentication phase, and password update phase. Their scheme is as follows: Initially, registration centers RC_a and RC_b send their IDs to each other and establish a shared key k_{ab} for further communication. Users and servers are registered as follows:

For server registration, server chooses it's identity SID_j and a random number s then sends them to registration center. After receiving, RC calculates following:

Table 1 Notations used in Gupta et al.'s scheme [12]

Notation	Meaning
RC, RC_a, RC_b	Registration centers
UID_i	User identity
SID_j	Server identity
RCID	Registration center identity
u	secret value of user
s	Secret value of server
r, c	Master secret values of registration center
BOI_i	User biometrics
PW_i	User password
N_{i1}, N_{i2}, N_{i3}	User, server and RC's random numbers respectively
TS_i, TS'_i	User's timestamp value
TS_j, TS'_j	Server's timestamp value
TS_{RC}, TS'_{RC}	RC's timestamp value
h(.)	Hash function
$\oplus$	Ex-OR operation
$\|\|$	Concatenation operation
ΔT, $\Delta T'$	Maximum tolerable difference of time
RC_{U_i}, RC_{S_j}	RCs of U_i and S_j respectively
k_{ab}	Secret key between two RC

$$PSID_j = h(SID_j \parallel s),$$
$$BS_j = h(PSID_j \parallel r)$$

and send back to server. After receiving BS_j, server stores it for later verification.

For user verification, user chooses UID_i, PW_i, u and imprints his biometrics then following is calculated:

$$PP_i = h(BOI_i \parallel PW_i \parallel UID_i),$$
$$A_i = h(u \parallel PP_i)$$

after this, user sends UID_i and u to it's RC. After receiving, RC calculates following:

$$PID_i = h(UID_i \parallel u),$$
$$B_i = h(PID_i \parallel c \parallel RCID)$$

then RC sends B_i to user. After receiving B_i, U_i, user calculates following:

$$C_i = h(UID_i \parallel A_i),$$
$$D_i = B_i \oplus C_i,$$
$$E_i = u \oplus PP_i$$

and stores C_i, D_i, E_i and h() into smart card.

Finally user and server are registered.

After this login and mutual authentication phase begins which has been explained in Fig. 1 below.

3 Shortcomings of Gupta et al.'s Protocol

3.1 Denial of Service Attack

The imprint biometric values are not exactly same at each time and we know hash function outputs are very sensitive to a small change in its input. In Gupta et al.'s scheme, biometric value of user is associated with $PP_i = h(BOI_i \parallel PW_i \parallel UID_i)$ and stored in smart card with $E_i = u \oplus PP_i$. Since biometric value could differ at each time, the values of PP_i and E_i will differ due to the sensitivity of hash function to its input. Therefore, during login phase

$$PP_i^* = h(BOI_i \parallel PW_i \parallel UID_i),$$
$$u = E_i \oplus PP_i^*,$$
$$A_i^* = h(u \parallel PP_i^*),$$
$$Ci^* = h(UID_i \parallel A_i^*)$$

equality of C_i^* and C_i will not hold.

3.2 Stolen Smart Card Attack

In Gupta et al.'s scheme, stolen smart card attack can be launched in following steps:

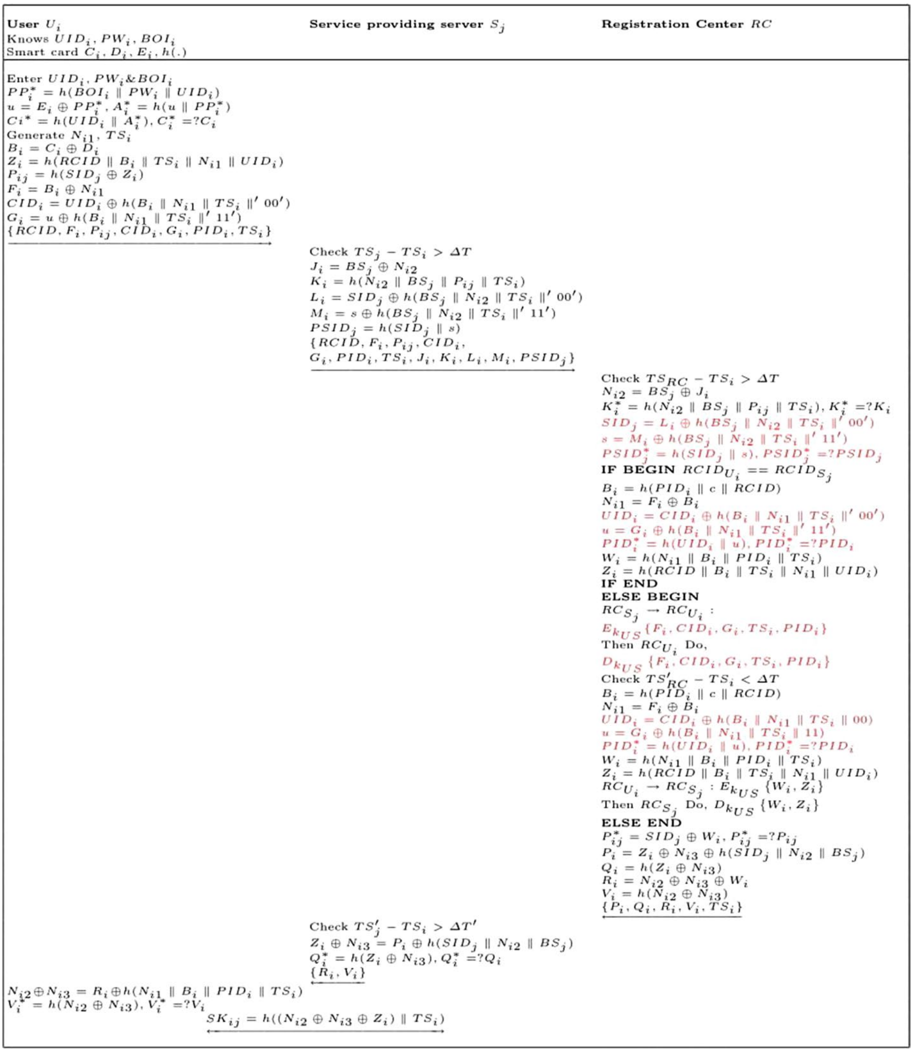

Fig. 1 Login and mutual authentication phase [12]

- In smart card, stored values can easily be extracted using the method of power analysis. So an attacker can get the values of C_i, D_i and F_i from the smart card.
- The attacker intercepts the login message $\{RCID, F_i, P_{ij}, CID_i, G_i, PID_i, TS_i\}$ and obtain the values. Then the attacker can calculate UID_i and u in the following steps:

$$
\begin{aligned}
B_i &= C_i \oplus D_i, \\
N_{i1} &= B_i \oplus F_i, \\
UID_i &= CID_i \oplus h(B_i \parallel N_{i1} \parallel TS_i \parallel 00), \\
u &= G_i \oplus h(B_i \parallel N_{i1} \parallel TS_i \parallel 11)
\end{aligned}
$$

now adversary can launch impersonation attack.

3.3 User Impersonation Attack

In Gupta et al.'s scheme, the attacker gets the UID_i and u as a result of the stolen smart card attack. Now the attacker can generate a valid login request message and impersonate a user without login at the client side. Steps are as follows:

- For impersonating a user, attacker generates a random number N_{i1} and computes system current time then following calculations are done:

$$\begin{aligned} Z_i &= h(RCID \parallel B_i \parallel TS_i \parallel N_{i1} \parallel UID_i), \\ P_{ij} &= h(SID \oplus Z_i), \\ F_i &= B_i \oplus N_{i1}, \\ CID_i &= UID \oplus h(B_i \parallel N_{i1} \parallel TS_i \parallel' 00'), \\ G_i &= u \oplus h(B_i \parallel N_{i1} \parallel TS_i \parallel' 11'). \end{aligned}$$

- After calculating these values, attacker generates login request message $\{RCID, F_i, P_{ij}, CID_i, G_i, PID_i, TS_i\}$ and sends it to server S_j.

3.4 No Provision of Perfect Forward Secrecy

Perfect forward secrecy means that if one of the long-term keys of a communication protocol is compromised then it will not lead to compromise of past session keys. Gupta et al.'s scheme have no provision of perfect forward secrecy. An attacker can generate the session keys as follows if he gets one of the long-term shared secret like-B_i.

- Initially, adversary A gets login request message $\{RCID, F_i, P_{ij}, CID_i, G_i, PID_i, TS_i\}$ and response message $\{R_i, V_i\}$.
- Now adversary have shared secret B_i, hence it calculates $N_{i1} = B_i \oplus F_i$, $UID_i = CID \oplus h(B_i \parallel N_{i1} \parallel TS_i \parallel' 00')$, $u = G_i \oplus h(B_i \parallel N_{i1} \parallel TS_i \parallel' 11')$, $W_i = h(N_{i1} \parallel B_i \parallel PID_i \parallel TS_i)$, $Z_i = h(RCID \parallel B_i \parallel TS_i \parallel N_{i1} \parallel UID_i)$.
- Then, further A calculates $N_{i2} \oplus N_{i3} = R_i \oplus W_i$.
- After this, A can calculate all the session keys as $SK = h((N_{i2} \oplus N_{i3} \oplus Z_i) \parallel TS_i)$.

Therefore, Gupta et al.'s scheme does not ensure perfect forward secrecy.

4 Proposed Scheme

In our scheme before all the phases to begin, initially control server will choose an elliptic curve E_p over a finite field F_p [24].

$$y^2 = x^3 + ax + b(modp).$$

where a, b$\in [1, p-1]$ and p is a prime number. Control server will choose a base point P on E_p with order n. It will release $\{E_p$, P$\}$ parameters and calculate public key as follows: It will choose a secret number y$\in [1, p-1]$ and then calculates

$$PK_{cs} = yP.$$

The notations used in the proposed scheme are described in Table 2.

4.1 Authentication Between Control Servers

CS_a will send it's $CSID_a$ to CS_b. After receiving $CSID_a$, CS_b will choose a master secret for control servers i.e. k_b then it will calculate

$$K_a = h(CSID_a \parallel k_b),$$

then CS_b will store this K_a and sends $CSID_b$ to CS_a. After receiving $CSID_b$, CS_a will use it's master secret k_a and calculates

$$K_b = h(CSID_b \parallel k_a),$$

then CS_a will store this K_b for later verification. The whole process has been shown in Fig. 2.

4.2 Registration Phase

4.2.1 Server Registration

Server registration will be done as follows:

1. Service providing server S_j will choose it's SID_j and sends it to control server through secure channel.
2. After receiving SID_j control server will choose master secret s, for service providing servers then it will calculate

$$AS_j = h(SID_j \parallel s),$$

   and sends it to server through a secure channel.
3. After receiving AS_j, server will store this. Now server is registered. The process has been depicted in Fig. 3.

4.2.2 User Registration

User registration will be done in the following way:

1. User chooses his ID as UID_i and password PW_i.
2. After this user will imprint his biometric B_i on sensor then U_i computes

$$Gen(B_i) = (\sigma_i, \theta_i).$$

3. User will choose a random number P_i and then calculates

$$PID_i = h(UID_i \parallel P_i),$$
$$A_i = h(PW_i \parallel \sigma_i)$$

   then user will send message (PID_i, A_i) to control server through secure channel.

Table 2 Notations

Notation	Meaning
E_p	A non-super singular elliptic curve over a finite field F_p
G	Additive group consisting points on E
P	Base point and generator of G with order n
n, p	Prime numbers
h(.)	Hash function
U_i	User/client
S_j	Server
CS, CS_a, CS_b	Control server
UID_i	User identity
SID_j	Server identity
CSID	Control server identity
PW_i	User's password
P_i	User's random value for registration
u	Control server's secret value for users
s	Control server's secret value for registration of servers.
k_a, k_b	Control server's master secret values for other control servers
RID_i	Authentication parameter for users
AS_j	Authentication parameter for servers
SK	Session key among user and server
PID_i	U_i pseudo identity
B_i	Biometric information of U_i
x, z	User and server random numbers for each session
y	CS's private key
PK_{cs}	CS's public key, where PK_{cs} = yP
PK_u, PK_s	Public keys of user and server, where $PK_u = xP, PK_s = zP$
$\|\|$	Concatenation operation
$\oplus$	Bitwise exclusive-or operation
$Gen()$	Generator function for biometrics
$Rep()$	Reproduction function for biometrics
TS_i, TS'_i	System time stamps of U_i
TS_j, TS'_j	System time stamps of S_j
TS_{CS}, TS'_{CS}	System time stamps of CS
$\Delta T, \Delta T'$	Maximum tolerant time difference between sender's and receiver's time stamp
CS_{U_i}, CS_{S_j}	CS belongs to U_i and S_j respectively

4. After receiving message from user, CS will choose a master secret u for users then following calculations will be done:

$$RID_i = h(PID_i \parallel u),$$
$$C_i = RID_i \oplus A_i,$$
$$R_i = h(PID_i \parallel A_i)$$

Control Server CS_a		**Control Server** CS_b
Choose $CSID_a$	$\xrightarrow{Sends\ CSID_a}$	After receiving $CSID_a$ Chooses master secret for control severs i.e k_b then it will calculate
Chooses master secret for control servers i.e k_a and calculates $K_b = h(CSID_b \parallel k_a)$ stores this for later verification	$\xleftarrow{Sends\ CSID_b}$	$K_a = h(CSID_a \parallel k_b)$ stores this for later verification

Fig. 2 Authentication between control servers

Server S_j		**Control Server**
Choose SID_j	$\xrightarrow{Sends\ SID_j\ (Secure\ channel)}$	After receiving SID_j Chooses master secret s for servers then it will calculate
Stores AS_j	$\xleftarrow{Sends\ AS_j\ (Secure\ channel)}$	$AS_j = h(SID_j \parallel s)$

Fig. 3 Server registration phase

User U_i		**Control Server** CS
Choose $UID_i, PW_i, B_i \& \ P_i$ $Gen(B_i) = (\sigma_i, \theta_i)$ $PID_i = h(UID_i \parallel P_i)$ $A_i = h(PW_i \parallel \sigma_i)$	$\xrightarrow{PID_i, A_i}$	Chooses master secret u $RID_i = h(PID_i \parallel u)$ $C_i = RID_i \oplus A_i$ $R_i = h(PID_i \parallel A_i)$ Store C_i, R_i in to smart card
Store θ_i into smart card	$\xleftarrow{smart\ card}$	

Fig. 4 User registration phase

now C_i and R_i will be saved in to a smart card and it will be given to user hand to hand then user will save θ_i in smart card. Now user is registered. The process has been shown in Fig. 4.

4.3 Login and Authentication Phase

User login will be done in the following way:

1. User scans his smart card in device and enters his UID and password and imprints his biometrics B_i^* in the sensor embedded in device and then device checks

$$Rep(B_i^*, \theta_i) = \sigma_i^*.$$

2. If σ_i^* matches with stored σ_i then it goes for further calculations otherwise it will not allow the access. Now after this device will calculate

$$PID_i^* = h(UID_i \parallel P_i),$$
$$A_i^* = h(PW_i \parallel \sigma_i),$$
$$R_i^* = h(PID_i^* \parallel A_i^*)$$

then it will check R_i against R_i^*, if it matches then it will go further otherwise login will be failed. Further it calculates

$$RID_i = C_i \oplus A_i.$$

3. Now device will choose a random number x$\in [1, \mathrm{p} - 1]$ and calculates following:

$$PK_u = xP,$$
$$PU_i = xPK_{cs},$$
$$G_i = A_i \oplus h(PU_i),$$
$$F_i = RID \oplus A_i,$$
$$CID_i = PID_i \oplus h(PK_u \parallel PU_i \parallel h(A_i) \parallel RID_i \parallel SID_j),$$
$$H_i = P_i \oplus h(PK_u \parallel PU_i \parallel h(A_i) \parallel SID_j)$$

now user U_i sends login request message M1 = {CSID, G_i, F_i, CID_i, H_i, PK_u, TS_i} to server S_j.

4. After receiving message from user, server S_j will proceed in following way. It will choose a random number z and computes it's system current time TS_j, then checks if $TS_j - TS_i > \Delta T$, if true then session expired otherwise calculates:

$$PK_s = zP,$$
$$PS_j = zPK_{cs},$$
$$J_i = AS_j \oplus h(PS_j),$$
$$K_i = SID_j \oplus h(PS_j),$$
$$L_i = h(AS_j \parallel PS_j \parallel PK_s \parallel K_i \parallel TS_i)$$

now server S_j will send M2 = {CSID, G_i, F_i, CID_i, H_i, J_i, K_i, L_i, PK_u, PK_s, TS_i} to it's control server.

5. After receiving message from server CS checks time stamp and checks the validity of message. Then CS calculates

$$PS_j = yPK_s,$$
$$AS_j = J_i \oplus h(PS_j),$$
$$SID_j^* = K_i \oplus h(PS_j).$$

Now CS will calculate

$$AS_j^* = h(SID_j^* \parallel s),$$

if AS_j^* is matching with AS_j then

$$L_i^* = h(AS_j^* \parallel PS_j \parallel PK_s \parallel K_i \parallel TS_i),$$

if L_i^* is matching with L_i then server is authenticated.

6. Now after the verification of S_j, user verification will be done in the following way. CS will calculate

$$\begin{aligned} PU_i &= yPK_u, \\ A_i &= G_i \oplus h(PU_i), \\ RID_i &= F_i \oplus h(A_i), \\ PID_i &= CID_i \oplus h(PK_u \parallel PU_i \parallel h(A_i) \parallel RID_i \parallel SID_j), \\ P_i &= H_i \oplus h(PK_u \parallel PU_i \parallel h(A_i) \parallel SID_j) \end{aligned}$$

after this CS will calculate RID_i by itself using secret number used for users

$$RID_i^* = h(PID_i \parallel u),$$

if RID_i^* is matching with RID_i then user is verified and calculates

$$\delta = h(h(PU_i) \parallel PK_u \parallel PK_s \parallel RID_i).$$

(a) Now if user is belonging to other CS then CS_{s_j} will calculate

$$K_{u_i} = h(CSID_{u_i} \parallel k_{s_j}).$$

Then CS will check if K_{u_i} is existing in database or not, if yes then CS_{u_i} is verified then further it will calculate

$$\begin{aligned} CSK &= y_{s_j} PK_{cs_{u_i}}, \\ \alpha_i &= SID_j \oplus h(CSK) \end{aligned}$$

then it will send following message to CS_{u_i} i.e. $\{CSID_{s_j}, G_i, F_i, CID_i, H_i, \alpha_i, PK_s, TS_i\}$.

(b) After receiving this message from CS_{s_j}, CS_{u_i} checks $TS_{cs} - TS_i < \Delta T$ if true then calculates $K_{s_j} = h(CSID_{s_j} \parallel k_{u_i})$, if it is existing in database then further calculates

$$\begin{aligned} CSK &= y_{u_i} PK_{cs_{s_j}}, \\ SID_j &= \alpha_i \oplus h(CSK), \\ PU_i &= y_{u_i} PK_u, \\ A_i &= G_i \oplus h(PU_i), \\ RID_i &= F_i \oplus h(A_i), \\ PID_i &= CID_i \oplus h(PK_u \parallel PU_i \parallel h(A_i) \parallel RID_i \parallel SID_j), \\ P_i &= H_i \oplus h(PK_u \parallel PU_i \parallel h(A_i) \parallel SID_j) \end{aligned}$$

after this CS_{u_i} will calculate RID_i by itself using secret number used for users

$$RID_i^* = h(PID_i \parallel u),$$

if RID_i^* is matching with RID_i then user is verified. Now following will be calculated.

$$\delta = h(h(PU_i) \parallel PK_u \parallel PK_s \parallel RID_i),$$
$$\beta_i = h(PU_i) \oplus h(CSK).$$

then CS_{u_i} will send $\{\beta_i, \delta\}$ to CS_{s_j}.

7. After receiving this, CS_{s_j} will calculate

$$h(PU_i) = \beta_i \oplus h(CSK),$$
$$W_i = h(PS_j \parallel h(PU_i) \parallel PK_u \parallel PK_s),$$
$$Q_i = W_i \oplus h(PS_j),$$
$$\gamma = h(PS_j \parallel PK_s \parallel PK_u \parallel AS_j \parallel W_i \parallel \delta).$$

Then CS_{s_j} will send $M_3 = \{Q_i, \gamma, \delta, TS_i\}$ to S_j.

8. After receiving this, S_j checks $TS_j' - TS_i < \Delta T$, if true then calculate

$$W_i = Q_i \oplus h(PS_j),$$
$$\gamma^* = h(PS_j \parallel PK_s \parallel PK_u \parallel AS_j \parallel W_i \parallel \delta),$$

if γ equal to γ^* holds then CS verified and further S_j calculates

$$PSU = zPK_u,$$

and calculates session key

$$SK = h(W_i \parallel PSU),$$

then S_j calculates

$$\eta_j = h(SID_j \parallel PSU \parallel \delta \parallel SK \parallel W_i),$$
$$V_j = W_i \oplus h(PSU)$$

and sends $M_4 = \{V_j, \eta_j, \delta, PK_s\}$ to user U_i.

9. After receiving this, U_i verifies if the equality of $\delta^* = h(h(PU_i) \parallel PK_u \parallel PK_s \parallel RID_i)$ and δ holds if yes then calculates session key as follows:

$$PSU = xPK_s,$$
$$W_i = V_j \oplus h(PSU),$$
$$SK = h(W_i \parallel PSU)$$

then it will verify

$$\eta_j^* = h(SID_j \parallel PSU \parallel \delta \parallel SK \parallel W_i),$$

if it holds then server and CS verified. Then it sends

$$\eta_i = h(PSU \parallel SID_j \parallel SK \parallel \delta),$$

$M_5 = \{\eta_i\}$ to S_j.

10. After receiving this, S_j verifies if the equality of $\eta_i^* = h(PSU \parallel SID_j \parallel SK \parallel \delta)$ and η_i holds then user U_i is authenticated and verified.

The login and authentication phase has been depicted in Fig. 5.

4.4 Password Update Phase

For updating password user scans smart card in device and enters his UID and password and imprints biometrics then device checks

$$Rep(B_i^*, \theta_i) = \sigma_i^*,$$

and if σ_i^* matched with stored σ_i then it calculates

$$\begin{aligned} PID_i^* &= h(UID_i^* \parallel P_i), \\ A_i^* &= h(PW_i^* \parallel \sigma_i), \\ R_i^* &= h(PID_i^* \parallel A_i^*) \end{aligned}$$

if R_i^* matches with R_i then $RID_i = C_i \oplus A_i$ will be calculated. After successful login user has to enter new password. After that following operations will be performed.

$$\begin{aligned} A_i^{new} &= h(PW_i^{new} \parallel \sigma_i), \\ R_i^{new} &= h(PID_i \parallel A_i^{new}), \\ C_i^{new} &= RID_i \oplus A_i^{new} \end{aligned}$$

now R_i^{new} and C_i^{new} will replace R_i and C_i. Thus, password is updated.

5 Security Analysis of Proposed Scheme

In this section, we are analyzing the security of our protocol. We present formal and informal security analysis. For formal analysis, we are using widely accepted Burrows–Abadi–Needham (BAN) logic [25] to show that our scheme is secure and valid. After that, we do formal security analysis by discussing different security attacks on our scheme such as stolen smart card attack, password guessing, man-in-the-middle attack and replay attack.

5.1 Formal Security Analysis Using BAN Logic

Different notations used in BAN logic are following:

- P| ≡X: Principal P believes statement X.
- #(X): Formula X is fresh.
- P| ⟹X: P has jurisdiction over statement X.
- P⊲X: P sees Statement X.
- P| ∼X: P once said statement X.

User U_i Knows UID_i, PW_i, B_i Smart card $C_i, R_i, \theta_i, h(.)$	**Service providing server** S_j	**Control Server** CS
Enter $UID_i, PW_i \& B_i$ $Rep(B_i^*, \theta_i) = \sigma_i^*)$ $If \sigma_i^* = \sigma_i$ then $PID_i^* = h(UID_i \parallel P_i)$ $A_i^* = h(PW_i \parallel \sigma_i)$ $R_i^* = h(PID_i^* \parallel A_i^*)$ $If R_i^* = R_i then$ $RID_i = C_i \oplus A_i$ chooses random number xϵ[1,p-1] $PK_u = xP$ $PU_i = xPK_{cs}$ $G_i = A_i \oplus h(PU_i)$ $F_i = RID \oplus A_i$ $CID_i = PID_i \oplus h(PK_u \parallel PU_i \parallel h(A_i) \parallel RID_i \parallel SID_j)$ $H_i = P_i \oplus h(PK_u \parallel PU_i \parallel h(A_i) \parallel SID_j)$ $\xrightarrow{\{CSID, G_i, F_i, CID_i, H_i, PK_u, TS_i\}}$		
	Check $TS_j - TS_i > \Delta T$ $PK_s = zP$ $PS_j = zPK_{cs}$ $J_i = AS_j \oplus h(PS_j)$ $K_i = SID_j \oplus h(PS_j)$ $L_i = h(AS_j \parallel PS_j \parallel PK_s \parallel K_i \parallel TS_i)$ $\xrightarrow{CSID, G_i, F_i, CID_i, H_i, PK_u, PK_s, J_i, K_i, L_i, TS_i\}}$	

		Check $TS_{CS} - TS_i > \Delta T$
		$PS_j = yPK_s$
		$AS_j = J_i \oplus h(PS_j)$
		$SID_j^* = K_i \oplus h(PS_j)$
		$AS_j^* = h(SID_j^* \parallel s)$
		Check $AS_j^* =?AS_j$
		$L_i^* = h(AS_j^* \parallel PS_j \parallel PK_s \parallel K_i \parallel TS_i)$
		Check $L_i^* =?L_i$
		IF BEGIN $CSID_{U_i} == CSID_{S_j}$
		$PU_i = yPK_u$
		$A_i = G_i \oplus h(PU_i)$
		$RID_i = F_i \oplus h(A_i)$
		$PID_i = CID_i \oplus h(PK_u \parallel PU_i \parallel h(A_i) \parallel RID_i \parallel SID_j)$
		$P_i = H_i \oplus h(PK_u \parallel PU_i \parallel h(A_i) \parallel SID_j)$
		$RID_i^* = h(PID_i \parallel u)$
		$RID_i^* =?RID_i$
		$\delta = h(h(PU_i) \parallel PK_u \parallel PK_s \parallel RID_i)$
		IF END
		ELSE BEGIN
		$K_{U_i} = h(CSID_{U_i} \parallel k_{s_j})$
		If K_{U_i} is existing in database then
		$CSK = y_{s_j} PK_{CS_{u_i}}$
		$\alpha_i = SID_j \oplus h(CSK)$
		$CS_{S_j} \rightarrow CS_{U_i}$: $\{CSID_{S_j}, G_i, F_i, CID_i, H_i, \alpha_i, PK_s, TS_i\}$
		$Check TS'_{CS} - TS_i < \Delta T$
		$K_{s_j} = h(CSID_{s_j} \parallel k_{u_i})$
		If K_{S_j} is existing in database then
		$CSK = y_{u_i} PK_{cs_{s_j}}$
		$SID_j = \alpha_i \oplus h(CSK)$
		$PU_i = y_{u_i} PK_u$
		$A_i = G_i \oplus h(PU_i)$
		$RID_i = F_i \oplus h(A_i)$
		$PID_i = CID_i \oplus h(PK_u \parallel PU_i \parallel h(A_i) \parallel RID_i \parallel SID_j)$
		$P_i = H_i \oplus h(PK_u \parallel PU_i \parallel h(A_i) \parallel SID_j)$
		$RID_i^* = h(PID_i \parallel u)$
		$RID_i^* =?RID_i$
		$\delta = h(h(PU_i) \parallel PK_u \parallel PK_s \parallel RID_i)$
		$\beta_i = h(PU_i) \oplus h(CSK)$
		$CS_{U_i} \rightarrow CS_{S_j} : \{\beta_i, \delta, TS_{cs}\}$
		$h(PU_i) = \beta_i \oplus h(CSK)$
		ELSE END
		$W_i = h(PS_j \parallel h(PU_i) \parallel PK_u \parallel PK_s)$
		$Q_i = W_i \oplus h(PS_j)$
		$\gamma = h(PS_j \parallel PK_s \parallel PK_u \parallel AS_j \parallel W_i \parallel \delta)$
		$\xleftarrow{\{Q_i, \gamma, \delta, TS_{cs}\}}$

	Check $TS'_j - TS_i < \Delta T'$ $W_i = Q_i \oplus h(PS_j)$ $\gamma^* = h(PS_j \parallel PK_s \parallel PK_u \parallel AS_j \parallel W_i \parallel \delta)$ $Check \gamma^* =?\gamma$ $PSU = zPK_u$ $SK = h(W_i \parallel PSU)$ $\eta_j = h(SID_j \parallel PSU \parallel \delta \parallel SK \parallel W_i)$ $V_j = W_i \oplus h(PSU)$ $\xleftarrow{V_j, \eta_j, \delta, PK_s, TS_{cs}}$	
$\delta^* = h(h(PU_i) \parallel PK_u \parallel PK_s \parallel RID_i)$ $Check \delta^* =?\delta$ $PSU = xPK_s$ $W_i = V_j \oplus h(PSU)$ $SK = h(W_i \parallel PSU)$ $\eta_j^* = h(SID_j \parallel PSU \parallel \delta \parallel SK \parallel W_i)$ $Check \eta_j^* = \eta_j$ $\eta_i = h(PSU \parallel SID_j \parallel SK \parallel \delta)$ $\xrightarrow{\eta_i}$		
	$\eta_i^* = h(PSU \parallel SID_j \parallel SK \parallel \delta)$ $Check \eta_i^* =?\eta_i$	

Fig. 5 Login authentication phase of our scheme

- (X, Y): Formula X or Y is one part of (X,Y).
- $\{X\}_K$: Formula X is encrypted with under the key K.
- $\langle X\rangle_Y$: Formula X is combined with formula Y.
- $P \overset{K}{\longleftrightarrow} Q$: P and Q uses the shared key K to communicate. It will never be known to anyone other than P and Q.
- $P \overset{X}{\rightleftharpoons} Q$: Formula X is known only to P and Q.

Four BAN logic rules are used to prove the mutual authentication of a particular authentication protocol:

- Message meaning rule: $\frac{R|\equiv R\overset{Y}{\longleftrightarrow}S, R\triangleleft\langle X\rangle_Y}{P|\equiv Q|\sim X}$
- Nonce verification rule: $\frac{P|\equiv\#(X), P|\equiv Q|\sim X}{P|\equiv Q|\equiv X}$
- Jurisdiction rule: $\frac{P|\equiv Q|\Longrightarrow X, P|\equiv Q|\equiv X}{P|\equiv X}$
- Freshness conjuncatenation rule: $\frac{P|\equiv\#(X)}{P|\equiv\#(X,Y)}$.

To prove that our scheme provides secure mutual authentication between service providing server and user we have to achieve following goals:

- G1: $U_i| \equiv \left(U_i \overset{SK}{\longleftrightarrow} S_j\right)$
- G2: $U_i| \equiv S_j| \equiv \left(U_i \overset{SK}{\longleftrightarrow} S_j\right)$
- G3: $S_j| \equiv \left(U_i \overset{SK}{\longleftrightarrow} S_j\right)$
- G4: $S_j| \equiv U_i| \equiv \left(U_i \overset{SK}{\longleftrightarrow} S_j\right)$.

After defining goals we transform messages in idealized form as follows:

- M1: $U_i \to S_j$: $\left\langle PID_i, SID_j, A_i, PK_u, U_i \overset{PU_i}{\longleftrightarrow} CS\right\rangle_{U_i \overset{RID_i}{\longleftrightarrow} CS}$
- M2: $S_j \to CS$: $\left\langle SID_j, PK_s, S_j \overset{PS_j}{\longleftrightarrow} CS\right\rangle_{S_j \overset{AS_j}{\longleftrightarrow} CS}$
- M3: $CS \to U_i$: $\left\langle PK_u, U_i \overset{PU_i}{\longleftrightarrow} CS, U_i \overset{PK_s}{\longleftrightarrow} S_j\right\rangle_{CS \overset{RID_i}{\longleftrightarrow} U_i}$
- M4: $CS \to S_j$: $\left\langle W_i, PK_s, \delta, U_i \overset{PK_u}{\longleftrightarrow} S_j, CS \overset{PS_j}{\longleftrightarrow} S_j\right\rangle_{CS \overset{AS_j}{\longleftrightarrow} S_j}$
- M5: $S_j \to U_i$: $\left\langle \delta, SID_j, W_i, U_i \overset{SK}{\longleftrightarrow} S_j\right\rangle_{S_j \overset{PSU}{\longleftrightarrow} U_i}$
- M6: $U_i \to S_j$: $\left\langle SID_j, \delta, U_i \overset{SK}{\longleftrightarrow} S_j\right\rangle_{S_j \overset{PSU}{\longleftrightarrow} U_i}$.

Now we make following initial assumptions about our protocol.

- A1: $U_i| \equiv \#(PK_u)$
- A2: $S_j| \equiv \#(PK_s)$
- A3: $U_i| \equiv U_i \overset{RID_i}{\longleftrightarrow} CS$
- A4: $CS| \equiv U_i \overset{RID_i}{\longleftrightarrow} CS$
- A5: $S_j| \equiv S_j \overset{AS_j}{\longleftrightarrow} CS$
- A6: $CS| \equiv S_j \overset{AS_j}{\longleftrightarrow} CS$
- A7: $U_i| \equiv CS| \Longrightarrow U_i \overset{PK_s}{\longleftrightarrow} S_j$
- A8: $S_j| \equiv CS| \Longrightarrow U_i \overset{PK_s}{\longleftrightarrow} S_j$
- A9: $S_j| \equiv U_i| \Longrightarrow U_i \overset{SK}{\longleftrightarrow} S_j$
- A10: $U_i| \equiv S_j| \Longrightarrow U_i \overset{SK}{\longleftrightarrow} S_j$.

Now we analyze idealized form of proposed scheme using BAN logic rules and given assumptions. The main proofs are following:

From message M1, we get
S1: $\text{CS} \lhd \left\langle PID_i, SID_j, A_i, PK_u, U_i \overset{PU_i}{\longleftrightarrow} CS \right\rangle_{U_i \overset{RID_i}{\longleftrightarrow} CS}$.
From A4, S1 and message meaning rule, we get
S2: $CS| \equiv U_i| \sim \left\langle PID_i, SID_j, A_i, PK_u, U_i \overset{PU_i}{\longleftrightarrow} CS \right\rangle_{U_i \overset{RID_i}{\longleftrightarrow} CS}$.
From message M2, we obtain
S3: $CS \lhd \left\langle SID_j, PK_s, S_j \overset{PS_j}{\longleftrightarrow} CS \right\rangle_{S_j \overset{AS_j}{\longleftrightarrow} CS}$.
From A6, message meaning rule and S3, we get
S4: $CS| \equiv S_j| \sim \left\langle SID_j, PK_s, S_j \overset{PS_j}{\longleftrightarrow} CS \right\rangle_{S_j \overset{AS_j}{\longleftrightarrow} CS}$.
From message M3, we get
S5: $U_i \lhd \left\langle PK_u, U_i \overset{PU_i}{\longleftrightarrow} S_j, U_i \overset{PK_s}{\longleftrightarrow} S_j \right\rangle_{U_i \overset{RID_i}{\longleftrightarrow} CS}$.
From A3, S5 and message meaning rule, we obtain
S6: $U_i| \equiv CS| \sim \left\langle PK_u, U_i \overset{PU_i}{\longleftrightarrow} S_j, U_i \overset{PK_s}{\longleftrightarrow} S_j \right\rangle_{U_i \overset{RID_i}{\longleftrightarrow} CS}$.
From A1, A2, S6 and nonce verification rule, we obtain
S7: $U_i| \equiv CS| \equiv U_i \overset{PK_s}{\longleftrightarrow} S_j$.
From A7, S7 and jurisdiction rule, we get
S8: $U_i| \equiv U_i \overset{PK_s}{\longleftrightarrow} S_j$.
According to PSU $= xPK_s$ we get
S9: $U_i| \equiv U_i \overset{PSU}{\longleftrightarrow} S_j$.
From message M4, we get
S10: $S_j \lhd \left\langle W_i, PK_s, \delta, U_i \overset{PK_u}{\longleftrightarrow} S_j, CS \overset{PS_j}{\longleftrightarrow} S_j \right\rangle_{CS \overset{AS_j}{\longleftrightarrow} S_j}$.

From A5, Message meaning rule and S10, we obtain

S11: $S_j| \equiv CS| \sim \left\langle W_i, PK_s, \delta, U_i \xleftrightarrow{PK_u} S_j, CS \xleftrightarrow{PS_j} S_j \right\rangle_{CS \xleftrightarrow{AS_j} S_j}$.

From A1, A2, S11 and nonce verification rule, we obtain

S12: $S_j| \equiv CS| \equiv U_i \xleftrightarrow{PK_u} S_j$.

From A8, S12 and jurisdiction rule, we obtain

S13: $S_j| \equiv U_i \xleftrightarrow{PK_u} S_j$.

According to PSU = zPK_u, we get

S14: $S_j| \equiv U_i \xleftrightarrow{PSU} S_j$.

From message M5, we get

S15: $U_i \triangleleft \left\langle \delta, SID_j, W_i, U_i \xleftrightarrow{SK} S_j \right\rangle_{S_j \xleftrightarrow{PSU} U_i}$.

From S9, S15 and message meaning rule

S16: $U_i| \equiv S_j| \sim \left\langle \delta, SID_j, W_i, U_i \xleftrightarrow{SK} S_j \right\rangle_{S_j \xleftrightarrow{PSU} U_i}$.

According to assumption A1 and A2, and freshness conjuncatenation rule we get

S17: $U_i| \equiv \#(W_i)$.

S18: $U_i| \equiv \#(\delta)$.

From statement S17, S18 and nonce verification rule, we obtain

S19: $U_i| \equiv S_j| \equiv \left\langle U_i \xleftrightarrow{SK} S_j \right\rangle$ (Goal 2).

From asuumption A10, S19 and jurisdiction rule, we obtain

S20: $U_i| \equiv \left\langle U_i \xleftrightarrow{SK} S_j \right\rangle$ (Goal 1).

From message M6, we get

S21: $S_j \triangleleft \left\langle SID_j, \delta, U_i \xleftrightarrow{SK} S_j \right\rangle_{S_j \xleftrightarrow{PSU} U_i}$.

From S14, S21 and message meaning rule, we get

S22: $S_j| \equiv U_i| \sim \left\langle SID_j, \delta, U_i \xleftrightarrow{SK} S_j \right\rangle_{S_j \xleftrightarrow{PSU} U_i}$.

From statement S17, S18 and nonce verification rule, we obtain

S23: $S_j| \equiv U_i| \equiv \left\langle U_i \xleftrightarrow{SK} S_j \right\rangle$ (Goal 4).

From A9, S23 and jurisdiction rule, we obtain

S24: $S_j| \equiv \left\langle U_i \xleftrightarrow{SK} S_j \right\rangle$ (Goal 3).

5.2 Informal Security Analysis

In this section, we will discuss different kinds of security attacks on our scheme to prove that our scheme is able to resist those attacks.

Mutual Authentication In our scheme, CS can authenticate both user and server separately. CS can verify user by checking whether RID_i and RID_i^* are matching or not, if they are equal then the user is verified. For server verification it checks AS_j with AS_j^* and L_i with L_i^*, if they are equal then server is verified. CS can also verify other CS in case of multi-control server environment by checking K_{u_i}, if it is existing in the database or not. After this CS generates authentication codes γ and δ and sends them to server S_j. Server S_j can verify user with the help of CS and can verify CS by checking the validity of γ. Now the server will generate authentication code η_j and sends it to user through which user can verify both CS and server by checking the validity of δ and η_j respectively. A server can also authenticate the user by checking the validity of η_i and η_i^*.

Anonymity and Untraceability In proposed scheme we are using pseudo identity for each user instead of real identity and also pseudo identity is included in $CID_i = PID \oplus h(PK_u \parallel PU_i \parallel h(A_i) \parallel RID_i \parallel SID_j)$ in which PK_u = xP, PU_i = xPK_{cs}, $A_i = h(PW_i \parallel \sigma_i)$. For obtaining PID_i attacker will have to compute all the given values otherwise he will not get user's real identity.

Also server ID, SID_j is included in $K_i = SID_j \oplus h(PS_j)$. For obtaining SID_j, the attacker will have to calculate PS_j, for which he will have to solve elliptic curve Diffie–Hellman problem. We can also see that values of CID_i and K_i will be different for each session as they are associated with x and z, which are random numbers chosen by user and server respectively for each session. Thus we can get the feature of untraceability because user and server are untraceable.

Insider Attack In our scheme, each user sends pseudo identity rather than actual UID_i and also sends pseudo password $A_i = h(PW_i \parallel \sigma_i)$ instead of actual password. It is very difficult to invert the one-way hash function and guessing the biometrics hence it is computationally difficult for an insider to derive the password PW_i. Therefore our scheme is secure against insider attack.

Password Guessing Attack In our scheme password PW_i is present in $A_i = h(PW_i \parallel \sigma_i)$. In smart card $R_i = h(PID_i \parallel A_i)$ and $C_i = RID_i \oplus A_i$ are stored. If smart card is stolen and if adversary A tries to guess the password then it is computationally infeasible without knowing identity and biometrics. Therefore A has no means of getting the password through stolen smart card. Hence our scheme is secure against password guessing attack.

Server Impersonation Attack For impersonating a server to user and CS, an attacker has to generate a valid $L_i = h(AS_j \parallel PS_j \parallel PK_s \parallel TS_j \parallel K_i)$, since attacker has no knowledge of AS_j and he cannot get PS_j. Therefore he cannot get generate valid L_i. Thus our scheme can withstand against server impersonation attack.

Replay Attack In our scheme, suppose attacker intercepts the message M1 = $\{CSID, G_i, F_i, CID_i, H_i, PK_u, TS_i\}$ and replay this message to establish new session with server. In step 9 even after getting message $\{V_j, \eta_j, \delta, TS_j, PK_s\}$, it will not be able to calculate PSU, W_i. Since server generates new PK_s for each session hence it will not be able to calculate session key.

In the second case if an attacker tries to replay the response message from the server then the equality of δ and δ^* and η_j and η_j^* will not hold for the same reason as PK_u and PK_s are new for each session. Also, we are using timestamp with each message so the attacker cannot send the same message again to launch replay attack.

Stolen Smart Card Attack Suppose user's smart card is stolen and an adversary is able to extract all the values from the smart card and also he has a previous login message still he will not be able to get the UID_i and P_i.

Denial of Service Attack In our scheme concept of fuzzy extractor has been used, when user imprints his biometrics B_i, then $Rep(B_i^*, \theta_i) = \sigma_i^*$ is calculated and σ_i^* is matched against stored σ_i. If there is a difference between σ_i and σ_i^* up to a threshold then the user will be granted access. Therefore the problem of the previous scheme is overcome.

User Impersonation Scheme Assume attacker A gets a valid SC and is trying to get access by impersonating a user. For that attacker has to generate a valid login message M1 =

$\{CSID, G_i, F_i, CID_i, H_i, PK_u, TS_i\}$ and for generating a login message attacker has to go through the login process. In login process attacker has to give valid credentials like—UID_i, PW_i and B_i. Although the attacker may try to guess UID and password but forging/copying biometrics is almost impossible. Therefore our scheme is secure against user impersonation attack.

Perfect Forward Secrecy Suppose all the secret keys used by control server are compromised even then an attacker cannot calculate session key because session key is including the parameters like—PS_j, PU_i which are including random numbers x and z, which are unique for each session. Calculating x and z from $PS_j = yzP$, $PU_i = xyP$ is computationally difficult due to the concept of elliptic curve Diffie–Hellman. Therefore our scheme ensures perfect forward secrecy.

Man-in-the-Middle Attack In our scheme control server authenticates user and server separately and user and server also authenticate each other. Therefore we can say our scheme is secure against the man-in-the-middle attack.

6 Performance Analysis

In this section we will compare our scheme with Gupta et al., He et al. and Yang et al's protocol. The comparison of security functionality between our scheme and existing schemes has shown in Table 3. We can conclude that our scheme provides security from different attacks like—stolen smart card attack and denial of service attack which were not resistible by Gupta et al.'s protocol. He et al.'s scheme was vulnerable to user impersonation attack and wrong password login and also it does not support multi-control server environment. Yang et al.'s scheme also has no support for multi-control server environment.

For analysing the performance we are assuming the length of identity, output of the hash function and the length elliptic curve point as 32, 160 and 320 bits respectively. In the server registration phase, server sends SID_j and the control server

Table 3 Security functionality analysis of propose and other multi-server authentication schemes

Security functionality	He et al. [13]	Gupta et al. [12]	Yang et al. [22]	Proposed protocol
Provides mutual authentication	Yes	Yes	Yes	Yes
Provides anonymity	Yes	Yes	Yes	Yes
Resist insider attack	Yes	Yes	Yes	Yes
Resist password guessing attack	Yes	Yes	Yes	Yes
Resist server impersonation attack	Yes	Yes	Yes	Yes
Resist replay attack	Yes	Yes	Yes	Yes
Resist stolen smart card attack	Yes	No	Yes	Yes
Resist denial of service attack	Yes	No	Yes	Yes
Resist user impersonation attack	No	No	Yes	Yes
Provides perfect forward secrecy	Yes	No	Yes	Yes
Resist man-in-the-middle attack	Yes	Yes	Yes	Yes
Provides three factor security	Yes	Yes	Yes	Yes
Provides multi-server environment	Yes	Yes	Yes	Yes
Supports multi-control server environment	No	Yes	No	Yes
Resist wrong password login	No	Yes	Yes	Yes
Drawback in password change phase	Yes	No	No	No
Single point registration of user/server	Yes	Yes	Yes	Yes

Table 4 Message length comparison between proposed scheme and other schemes

Protocols	Message length (byte)				
	$U_i \rightarrow S_j$	$S_j \rightarrow CS$	$CS \rightarrow S_j$	$S_j \rightarrow U_i$	$U_i \rightarrow S_j$
He et al. [13]	80	160	80	100	20
Gupta et al. [12]	104	204	80	40	–
Yang et al. [22]	100	–	–	40	20
Propose protocol	124	224	60	100	20

Table 5 Computational cost comparison between proposed scheme and other schemes

	He et al.'s scheme [13]	Gupta et al.'s scheme [12]	Yang et al.'s scheme [22]	Proposed scheme
User	$3T_m + 7T_h$	$6T_h$	$9T_h$	$11T_h + 3T_m$
Server	$2T_m + 5T_h$	$5T_h$	$8T_h$	$7T_h + 3T_m$
Control server	$2T_m + 9T_h$	$14T_h + 4T_{Enc/Dec}$	–	$12T_h + T_m$
Total	$7T_m + 21T_h$	$25T_h + 4T_{Enc/Dec}$	$17T_h$	$30T_h + 8T_m$

sends $AS_j = h(SID_j \parallel s)$ to the server. Hence the communication cost of server registration phase is (32 + 160) = 192 bits. In the user registration phase, user sends $PID_i = h(UID_i \parallel P_i)$ and $A_i = h(PW_i \parallel \sigma_i)$ to the control server then the communication cost of user registration phase is (160 + 160) = 320 bits. In login and authentication phase five messages are exchanged between user, server and control server. The length of messages{CSID, G_i, F_i, CID_i, H_i, PK_u, TS_i}, {CSID, G_i, F_i, CID_i, H_i, J_i, K_i, L_i, PK_u, PK_s TS_i}, {Q_i, γ, δ, TS_i }, {V_j, η_j, δ, PK_s}, {η_i} are 124, 224, 60, 100, 20 bytes respectively. Table 4 shows the comparison of communication cost of related schemes.

For computational cost analysis, we have taken the computation cost of elliptic curve point and hash function operation into consideration. Bitwise Ex-OR operation could be ignored as compared to the computational cost of other two operations. Table 5 presents the comparison of the computational cost of authentication phase of the proposed scheme and other existing schemes.

In Tables 4, 5, we see that our scheme has more communicational and computational cost than existing schemes. Nevertheless, Gupta et al.'s protocol cannot withstand the stolen smart card attack, user impersonation attack and denial of service attack. He et al.'s scheme does not support multi-control server environment and cannot withstand user impersonation attack and Yang et al.'s scheme also has no support for multi-control server environment. For a cryptographic protocol, security is the most important factor. So achieving higher security at the cost of increasing communication and computation cost is worthy. Our scheme could overcome the weaknesses of existing schemes. Therefore our scheme is more reliable for multi-control server and multi-server environment.

7 Conclusion

In this paper, we proposed an updated scheme to provide mutual authentication and key establishment among user and server for a multi-control server environment. We have discussed possible vulnerabilities in Gupta et al's scheme and mitigated those vulnerabilities.

We used the concept of elliptic curve cryptography for providing improved security and for biometrics we have used the fuzzy extractor to protect it from denial of service kind of scenario. In this scheme even if an attacker gets the smart card and login message both he will not be able to get UID and the random number, hence it is saved from user impersonation and stolen smart card attack. In the consequence of informal security analysis, we can say that proposed scheme is secure against the man-in-the-middle attack, impersonation attack, replay attack and offline password guessing attack. We have used the concept of BAN logic to prove the secure mutual authentication property of our scheme among client and server through formal security analysis. Finally, performance analysis has been done at the end of the paper and compared it with other schemes. After doing all the analysis work we come to the conclusion that our scheme is secure and provides better authentication between client and server among existing schemes.

References

1. Yang, H. W., Yang, C. C., & Lin, W. (2013). Enhanced digital rights management authentication scheme based on smart card. *IET Information Security*, *7*(3), 189–194.
2. Fan, C. I., Chan, Y. C., & Zhang, Z. K. (2005). Robust remote authentication scheme with smart cards. *Computers and Security*, *24*(8), 619–628.
3. Amin, R. (2016). Cryptanalysis and efficient dynamic id based remote user authentication scheme in multi-server environment using smart card. *International Journal of Network Security*, *18*(1), 172–181.
4. Wei, J., Liu, W., & Hu, X. (2016). Secure and efficient smart card based remote user password authentication scheme. *IJ Network Security*, *18*, 782–792.
5. Li, X., Niu, J., Kumari, S., Liao, J., & Liang, W. (2015). An enhancement of a smart card authentication scheme for multi-server architecture. *Wireless Personal Communications*, *80*(1), 175–192.
6. Pippal, R. S., Jaidhar, C. D., & Tapaswi, S. (2013). Robust smart card authentication scheme for multi-server architecture. *Wireless Personal Communications*, *72*(1), 729–745.
7. 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.
8. Mishra, D., Das, A. K., & Mukhopadhyay, S. (2014). A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards. *Expert Systems with Applications*, *41*(18), 8129–8143.
9. Baruah, K., Banerjee, S., Dutta, M., & Bhunia, C. T. (2015). An improved biometric-based multi-server authentication scheme using smart card. *International Journal of Security and Its Applications*, *9*, 397–408.
10. 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), 1–25.
11. 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.
12. Gupta, P. C., & Dhar, J. (2016). Hash based multi-server key exchange protocol using smart card. *Wireless Personal Communications*, *87*(1), 225–244.
13. He, D., & Wang, D. (2015). Robust biometrics-based authentication scheme for multiserver environment. *IEEE Systems Journal*, *9*(3), 816–823.
14. 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.
15. Feng, Q., He, D., Zeadally, S., & Wang, H. (2017). Anonymous biometrics-based authentication scheme with key distribution for mobile multi-server environment. *Future Generation Computer Systems*, *84*, 239.
16. Kumari, S., Das, A. K., Li, X., Wu, F., Khan, M. K., Jiang, Q., et al. (2018). A provably secure biometrics-based authenticated key agreement scheme for multi-server environments. *Multimedia Tools and Applications*, *77*(2), 2359–2389.

17. Xu, D., Chen, J., & Liu, Q. (2019). Provably secure anonymous three-factor authentication scheme for multi-server environments. *Journal of Ambient Intelligence and Humanized Computing*, *10*(2), 611–627.
18. Chandrakar, P., & Om, H. (2017). Cryptanalysis and extended three-factor remote user authentication scheme in multi-server environment. *Arabian Journal for Science and Engineering*, *42*(2), 765–786.
19. Kumar, A., & Om, H. (2018). An improved and secure multiserver authentication scheme based on biometrics and smartcard. *Digital Communications and Networks*, *4*(1), 27–38.
20. Meadows, C. (2006). Formal methods for cryptographic protocol analysis: Emerging issues and trends. *IEEE Journal on Selected Areas in Communications*, *21*(1), 44–54.
21. Pan, H. T., Pan, C. S., Tsaur, S. C., & Hwang, M. S. (2016). Cryptanalysis of efficient dynamic id based remote user authentication scheme in multi-server environment using smart card. In *2016 12th International conference on computational intelligence and security (CIS)* (pp. 590–593).
22. Yang, L., & Zheng, Z. (2018). Cryptanalysis and improvement of a biometrics-based authentication and key agreement scheme for multi-server environments. *PLOS ONE*, *13*(3), 1–27.
23. Xue, K., Hong, P., & Ma, C. (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.
24. Seroussi, G. (1999). Elliptic curve cryptography. In *1999 Information theory and networking workshop (cat. no. 99EX371)* (p. 41).
25. Burrows, M., Abadi, M., & Needham, R. (1990). A logic of authentication. *ACM Transactions on Computer Systems*, *8*(1), 18–36.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Ashish Tomar has received B.E. (Information Technology) from SGSITS, Indore, India in 2015. At present, he is pursuing Full Time M.Tech. in Information Security at ABV-Indian Institute of Information Technology and Management, Gwalior, India. His research areas of interest include Network Security, Cryptography etc.

Dr. Joydip Dhar received M.Sc. (1991) from Visva-Bharati University, Santenekaten, India; and Ph.D. (1997) from Indian Institute of Technology Kanpur, India. He is currently an Associate Professor at ABV-Indian Institute of Information Technology and Management, Gwalior, India since 2006. His research interests are computational mathematics, mathematical modelling and simulation of biological, environmental, managerial and engineering systems.