Remote Login Authentication Scheme based on Bilinear Pairing and Fingerprint

[*]Shipra Kumari[1] and Hari Om[2]
Department of Computer Science and Engineering
Indian School of Mines, Dhanbad – 826004 India
Email: [1]shiprakumari18jan@gmail.com, [2]hariom4india@gmail.com
*Correspondent author: Shipra Kumari

Received June 26, 2015; revised September 23, 2015; accepted October 7, 2015; published December 31, 2015

Abstract

The bilinear pairing, also known as Weil pairing or Tate pairing, is widely used in cryptography and its properties help to construct cryptographic schemes for different applications in which the security of the transmitted data is a major concern. In remote login authentication schemes, there are two major requirements: i) proving the identity of a user and the server for legitimacy without exposing their private keys and ii) freedom for a user to choose and change his password (private key) efficiently. Most of the existing methods based on the bilinear property have some security breaches due to the lack of features and the design issues. In this paper, we develop a new scheme using the bilinear property of an elliptic point and the biometric characteristics. Our method provides many features along with three major goals. a) Checking the correctness of the password before sending the authentication message, which prevents the wastage of communication cost; b) Efficient password change phase in which the user is asked to give a new password after checking the correctness of the current password without involving the server; c) User anonymity - enforcing the suitability of our scheme for applications in which a user does not want to disclose his identity. We use BAN logic to ensure the mutual authentication and session key agreement properties. The paper provides informal security analysis to illustrate that our scheme resists all the security attacks. Furthermore, we use the AVISPA tool for formal security verification of our scheme.

Keywords: Authentication, User anonymity, Bilinear Pairing, Biometric, Elliptic curve cryptography (ECC).

http://dx.doi.org/10.3837/tiis.2015.12.014 **ISSN : 1976-7277**

1. Introduction

In the realm of computer networks, we save time and money by accessing the resources and services online. For innumerable day-to-day activities, we depend on the internet that makes our life much easier, for example, using ATM instead of waiting in long bank queues, booking e-tickets for train and flights, shopping through e-Commerce websites. Though these facilities are easily available and widely used all around the world, yet for accessing these services or resources, we depend on the transmission of data through the insecure channels. It involves a high risk of eavesdropping and intercepting of messages/resources by an adversary or enemy for his benefit. Thus, there is a need for a remote login authentication mechanism which can verify the legitimacy of a user and the service provider before exchanging the actual services.

In this paper, we propose a remote user authentication scheme that uses the bilinear property of an elliptic point and the biometric characteristics of the user. The bilinear property of an elliptic point and the user biometric can provide stronger security. The biometric enhances the security of the scheme as these characteristics of a person cannot easily be copied, guessed, stolen, forged or forgotten and are unique for every user. Thus, use of biometric with smart card and password provides three way authentication. Our proposed scheme checks the correctness of a password before sending the authentication message and as a result it avoids the communication cost in case a legal user or an adversary enters wrong password. In password change phase, it does not require the server's involvement; thus, avoiding the communication cost. It hides the user identity, making it suitable for the applications where the user does not want to disclose his identity. Furthermore, it can resist many security attacks, such as replay attack, impersonation attack, password guessing attack, known key secrecy, denial of service attack, etc.

In order to design an efficient user authentication and key agreement protocol for accessing either a single server or multiserver system, the following security aspects should be achieved:

a. No verification table should be involved at the server end.
b. An efficient login phase and password change phase is necessary so that the protocol can detect wrong input information(s) in the early stage.
c. Mutual authentication property should be provided.
d. Session key agreement and verification are essential.
e. Resistance to the denial-of-service (DOS) attack.
f. User anonymity should be preserved.
g. Resistance to the off-line password/identity guessing attack.
h. Resistance to the user-server impersonation attack.
i. Resistance to the insider attack.
j. Resistance to the replay attack.
k. The protocol should resist the session key disclosure attack.
l. The protocol should provide perfect forward/backward secrecy.
m. The protocol should resist the stolen smart card attack.

2. Related Works

Remote login authentication has wide applications, especially in today's scenario, where most of the transactions are done using computer networks. This has led several researchers to develop secure schemes that can achieve all the security goals and requirements. Since the development of the remote scheme by Lamport [1], several schemes [2-22,28,32] have been discussed that use various approaches. These schemes have some pros and cons as far as the security features and security breaches are concerned. Now-a-days, bio-cryptography is emerging as a powerful solution for user authentication, which can combine the advantages of both conventional cryptography and biometric security [32]. Li and Hwang have discussed a biometric based remote user authentication scheme that uses the user biometric information to prove the legality of user [2]. In [3], Das points out the flaw of the scheme [2] and proposes a new security protocol. In [4], Lee et al. discuss a scheme by removing the security flaws in the Li et al.'s scheme [2]. The paper [5] shows that the Das's scheme [3] does not resist the insider attack, password guessing, user and server impersonation attacks, and fails to achieve mutual authentication. Li et al. show that the Das's scheme is vulnerable to the forgery and stolen smart card attacks and have enhanced the scheme in [7]. In [8], Chaturvedi et al. report that the schemes [5,7] are inefficient in the login phase as the user password has no role in these schemes; moreover, they cannot resist the replay attack and known session specific temporary information attack. The Chaturvedi et al.'s scheme [8] is based on the exponential computation, which makes it costly as it needs more bits for transmitting the authentication messages. There have been developed some schemes by using the elliptic curve property, which reduces the computational and communication costs as less number of bits is required to compute the elliptic curve points. The paper [9] discusses a scheme for smart card authentication using bilinear pairings that provides the users a facility to choose and change their passwords by their own choices. However, the papers [10-13] report its vulnerability. Juang et al. [10] report that the scheme [9] suffers from different attacks like replay attack, password guessing attack, forgery attack, etc. Furthermore, it lacks mutual authentication and verification of the old password in the password change phase. Fang et al. [11] improve the scheme [9] by removing its weaknesses. Giri and Srivastava [12] discuss an improvement over the Fang et al.' scheme. Awashthi [13] shows that the Giri and Srivastava's scheme is still insecure against the theft and on-line attack and discusses a better scheme. The Awasthi's scheme [13], however, lacks the mutual authentication feature and cannot resist some important attacks. Yoon et al. [14] discuss an important authentication scheme. This scheme, as reported by Xie [15] scheme cannot resist the stolen-verifier attack, off-line password guessing attack. Xie [15] discusses an authentication scheme using the elliptic curve cryptography (ECC). Farash et al. [16] find that the Xie's scheme [15] is also susceptible to the impersonation attack and off-line password guessing attack.

The above mentioned schemes do not provide the user anonymity. Based on the Farash et al.'s work [16], Zhang et al. [17] have recently discussed an authentication scheme with anonymity. Islam and Biswas discuss an ECC-based password authentication and key agreement scheme using a smart card [18]. Li [19] points out that the Islam and Biswas's scheme [18] cannot resist the off-line password guessing attack, stolen-verifier attack, and insider attack and overcomes its drawbacks in his scheme. Lee et al. [20] discover that both the original and modified schemes [18, 19] are vulnerable to the insider attack and they have overcome this problem in their scheme.

Tang et al. [21] discuss a scheme based on ECC; however, it does not check the password correctness before sending the authentication message, resulting in wastage of the communication cost. Karuppiah et al. [22] present a scheme, which is claimed to be more secure. It however uses exponentiation to compute the authentication messages; thus, increasing the cost of communication.

In this paper, we propose a new authentication scheme based on ECC and biometric, which fulfills all the security requirements, and also prevents the waste of communication cost. The rest of the paper is organized as follows. Section 3 gives the attacker model, which defines the capabilities of an adversary on an insecure channel. Section 4 provides preliminaries that are required for further discussion in the paper. In Section 5, we discuss our proposed scheme. Section 6 presents its security analysis and section 7 presents the security proof using BAN Logic. In section 8, we present the simulation of our scheme using the AVISPA tool and in section 9 the comparative performance of our scheme along with the related schemes is given. Finally, section 10 concludes the paper.

3. Attacker Model

In this section, we describe the risk of the authentication schemes. As an authentication protocol is executed over an insecure channel, the attacker has several advantages or capabilities. In the following, we present some valid assumptions:

- An attacker first taps the communication channel to obtain the messages and then tries to get the secret values.
- An attacker may be a legitimate user.
- An attacker may eavesdrop all the communications between the entities involved in the protocol over a public channel. It is also assumed that an attacker cannot intercept the message over a secure channel.
- An attacker can modify, delete, resend, and reroute the eavesdropped messages.
- An attacker can extract the smart card information by monitoring its power consumption. For example, if an attacker gets the smart card of a valid user, he may get all the stored information in the smart card.
- The attacker knows the protocol description, i.e., the protocol is public.
- An attacker can guess a low entropy password and identity individual password (parameters) easily, but guessing two secret parameters (e.g. Password, identity) is computationally infeasible in polynomial time. If we assume that the length of the user's identity and password are of n characters, then the probability of guessing a string composed of approximately n characters is $1/2^{6n}$ [27].

4. Preliminaries

In this section, we briefly review the basic concepts of fuzzy extraction, ECC, bilinear pairings, and the related mathematical problems.

4.1 Fuzzy Extractor

A fuzzy extractor deals with non-uniformity and error tolerance [24-25, 29-31]. It reliably alters biometric input information in a uniformly random string R in an error tolerant approach.

Therefore, it may be appropriate for the cryptographic schemes which use biometric. If the input changes, but remains closed, the extracted R remains the same. To assist in recovering R from the entered biometric, a fuzzy extractor outputs a public string P. P, known as Helper data, is derived only from the biometric template and the cryptographic key R is generated from the helper data and the biometric query B. If the biometric template and query are from the same user, then the generated keys will be the same with overwhelming probability [31]. A fuzzy extractor consists of a pair of efficient randomized procedures, Gen and Rep, which mean 'generate' and 'reproduce', respectively, as given below:

$Gen(B) = (R, P)$,

where B is biometric information, R and P are random strings generated by *Gen*.

$R^* = Rep(B^*, P)$,

where B* is biometric information and P is a public string used by *Rep* to reproduce R*.

To reproduce the same R, i.e., R=R*, the metric space distance between B and B* has to satisfy the verification threshold.

4.2 Elliptic Curve

The equation of a non-singular elliptic curve $E_q(a, b)$ over a finite field Z_q (q is a large prime number greater than 3) can be written as follows:

$$y^2 \equiv x^3 + ax + b(\text{mod } q)$$

where a and b are constants such that $4a^3+27b^3 \neq 0$ *mod* q, which must be satisfied for its non-singularity.

Any point $Q(x, y) \in E_q(a, b)$, $x, y \in Z_q$ together with O, called 'point at infinity' forms an additive cyclic group $E = \{(x, y) \in E_q(a, b)\} \cup \{O\}$,where O serves as the additive identity element of the group. The point addition and scalar multiplication with a point are defined as follows:

a. Point Addition

If $Q(x_1,y_1)$ and $R(x_2,y_2)$ are two points on an elliptic curve, the resultant point $S(x_3, y_3) = Q + R$ is computed as follows:

$x_3 = m^2 - x_1 - x_2$;

$y_3 = -(y_1 + m(x_3 - x_1))$

where,

$m = (y_2 - y_1) / (x_2 - x_1)$, If $R \neq Q$

$m = (3x_1^2 + a)/2y_1$, If $R = Q$.

b. Point Multiplication with a scalar value

The point multiplication with a scalar *k* is computed by repeated addition of k times as follows:

$k \cdot Q = Q + Q + \ldots$ k times.

4.3 Bilinear Pairings

Let G_1 denote an additive cyclic group of prime order q, and G_2 a multiplicative cyclic group of the same order. A pairing is a map $\hat{e}: G_1 \times G_1 \rightarrow G_2$, which satisfies the following properties:

• *Bilinearity:* For all $Q, R, S \in G_1$, $\hat{e}(Q + R, S) = \hat{e}(Q, S) \cdot \hat{e}(R, S)$ and $\hat{e}(Q, R + S) = \hat{e}(Q, R) \cdot \hat{e}(Q, S)$. As a result $\hat{e}(a \cdot Q, b \cdot R) = \hat{e}(Q, R)^{ab}$ for all $Q, R \in G_1$ and for all $a, b \in Z_q^*$, where $Z_q^* = Z_q - \{0\}$.

- *Non-degenerate:* There exist S, Q$\in$ G_1 such that $\hat{e}$ (S, Q) $\neq 1_{G2}$, where 1_{G2} is the identity element of group G_2.
- *Computability:* There is an efficient algorithm to compute $\hat{e}$ (Q, R) for any Q, R $\in$ G_1.

4.4 Computational Problems

Definition 1: Elliptic Curve Discrete Logarithm Problem (ECDLP)
Given Q, R $\in$ G_1,
Find an integer k $\in$ Z_q^*, such that R = k · Q.

Definition 2: Computational Diffie–Hellman Problem (CDHP)
Given (Q, a·Q, b·Q) for any a, b $\in$ Z_q^*,
Computation of ab·Q is hard in group G_1.

Definition 3: Decisional Diffie–Hellman Problem (DDHP)
Given (Q, a·Q, b·Q, c·Q) for any a, b, c $\in$ Z_q^*,
Decide whether c·Q = ab·Q, i.e., decide if c = ab *mod* q or not.

5. Proposed Scheme

In this section, we propose an efficient remote login authentication scheme using fingerprint. There are two kinds of participants in our scheme: the login users and server. Each legitimate user can get services from the server only when he has registered with the server. So, a new user must register himself with the server to access the services. The scheme has five phases: initialization phase, registration phase, login phase, authentication phase, and password change phase. In the initialization phase, the server computes its public and private parameters. In the registration phase, the new user requests the server for registration and after some initial verification, the server registers the new user and provides a smart card to him. The smart card contains some user's parameters. In login phase, a user must enter his secret values like identity, password, and biometric in the device attached to the system along with the smart card. In this phase, the correctness of the values entered by the user are first checked and then a login message is sent to the server. In authentication phase, the server first verifies the user's legitimacy and then sends an authentication message to him. After receiving the authentication message, the user also verifies the server's authenticity. Additionally, a session key is computed by both the participants, i.e. the user and server, for further communication in current login session. A password change phase is a feature provided in the scheme for giving a facility for a user to change his password whenever he wishes. Figs. 1 and 2 illustrate the proposed scheme and Table 1 consists of the notations used in our scheme. The detailed description of all the steps involved in the scheme is given below.

Table 1. Notations used in the Paper

Symbol	Meaning
U_i	i^{th} User
S_j	j^{th} Server
SID_j	j^{th} Server's identity
ID_i	i^{th} User's Identity
PW_i	i^{th} User's Password
B_i	i^{th} User's Biometric
R_i and P_i	Random strings generated by Gen(B_i) function
q	Large prime number
H(.)	One way hash function
$E_q(a,b)$	Elliptic Curve over F_q with parameters a & b
G	Base point on $E_q(a,b)$
d	Secret key of server
Q_d	Public key of server
ê	Pairing operation
T_1,T_2,T_3 and T_4	Timestamps
r_u and r_s	Random numbers
G_1	Additive cyclic group of prime order q
G_2	Multiplicative cyclic group of prime order q

5.1 Initialization Phase

This is the setup phase of the system in which the server computes the public and secret parameters.

The server chooses G_1 as an additive cyclic group of a prime order q, and G_2 as a multiplicative cyclic group of the same order. It defines a bilinear mapping ê: $G_1 \times G_1 \rightarrow G_2$. It also defines a cryptographic one-way hash function H and a Elliptic curve $E_q(a,b)$.

The server selects a Base Point G on the elliptic curve and a secret key *d* and then computes the corresponding public key $Q_d = d \cdot G$. Finally, it publishes the system parameters $\{G_1, G_2, ê, q, G, Q_d, H\}$ and keeps *d* secret.

5.2 Registration Phase

This phase is used to register a new user with the server as only the registered users can access the server. To register himself as a new user, the user U_i first chooses his identity ID_i and password PW_i and then he registers his fingerprints B_i using a fuzzy extractor such that $Gen(B_i) = (R_i, P_i)$, where R_i and P_i are random strings generated by *Gen* function. The $PB_i = H(PW_i \,||\, R_i)$ is computed and the message $\{ID_i, PB_i, P_i\}$ is sent to the server through a secure channel.

The server computes $CID_i = (d||ID_i) \cdot G$, $HPW_i = CID_i + PB_i \cdot Q_d$ and $A_{1i} = (PB_i \,||ID_i) \cdot G$.

The values $\{HPW_i, A_{1i}, G, Q_d, q, P_i, H, E_q(a,b)\}$ are stored in the smart card and it is sent to the user securely.

The user registration phase is summarized in **Fig. 1.**

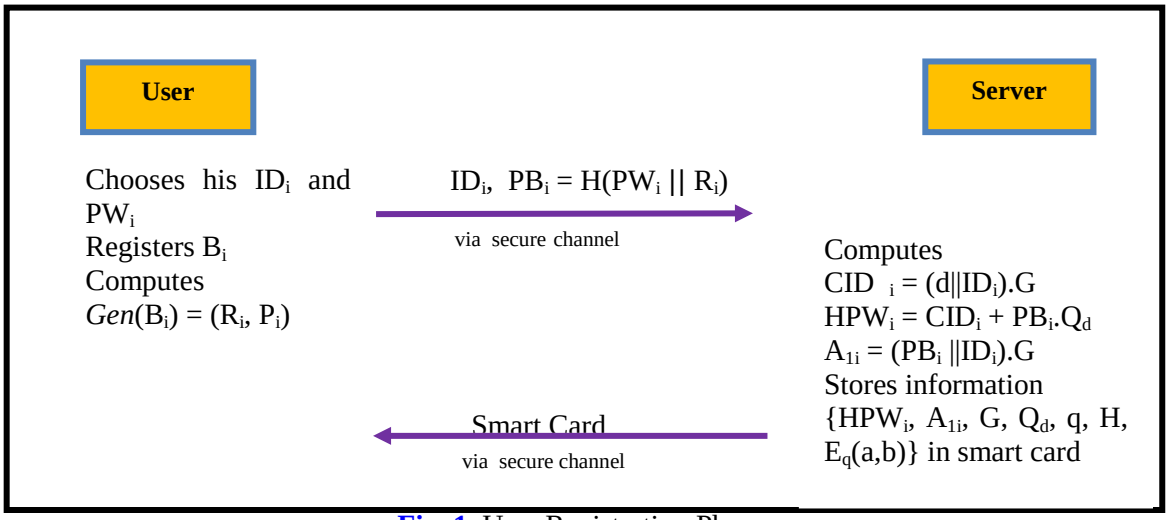

Fig. 1. User Registration Phase

5.3 Login Phase

When a user wants to log into the system, he inserts his smart card into the terminal attached with the system and keys in his ID_i* and password PW_i* into the terminal and also provides his fingerprint into the device.

The smart card computes R_i* = Rep(B_i*,P_i), PB_i* = H(PW_i* || R_i*) and A_{1i}* = (PB_i*||ID_i*)·G.

If A_{1i}* ≠ A_{1i}, terminate request; otherwise, the smart card computes CID_i* = HPW_i – (PB_i*·Q_d).

Proof: CID_i* = HPW_i – PB_i*·Q_d = CID_i + PB_i · Q_d – PB_i*·Q_d = CID_i

The smart card generates a random number r_u and computes $A_{2i} = r_u \cdot G$, $NID_i = ID_i^* + r_u \cdot Q_d$ and $A_{3i} = \hat{e}((T_1 \cdot r_u \cdot CID_i^* + A_{2i}), Q_d)$, where T_1 is the current time of login and it is assumed that the system is time synchronized.

The smart card sends the login message $\{NID_i, A_{2i}, T_1, H(A_{3i})\}$ to server.

5.4 Authentication phase

The server receives a login message $\{NID_i, A_{2i}, T_1, H(A_{3i})\}$ at time T_2. It checks if $(T_2\text{-}T_{1)} < \Delta T$, where ΔT is legal tolerant time. If $(T_2 – T_1) > \Delta T$, terminate a login session; otherwise, continue.

The server computes $ID_i^{**} = NID_i – (d \cdot A_{2i})$ and checks the format and existence of ID_i^{**}

It also computes $CID_i^{**} = (d\|ID_i^{**}) \cdot G$ and $A_{3i}^* = \hat{e}(CID_i^{**}, A_{2i})^{T1.d} \cdot \hat{e}(d \cdot A_{2i}, G)$.

Then the server compares $H(A_{3i}^*)$?= $H(A_{3i})$. If they are equal, the server authenticates the user; otherwise terminate login session.

Proof:

$$
\begin{aligned}
A_{3i}^* &= \hat{e}(CID_i^{**}, A_{2i})^{T1.d} \cdot \hat{e}(d \cdot A_{2i}, G) \\
&= \hat{e}(T_1 \cdot CID_i^{**}, d \cdot A_2) \cdot \hat{e}(A_{2i}, G)^d \\
&= \hat{e}(T_1 \cdot CID_i^{**}, d \cdot r_u \cdot G) \cdot \hat{e}(A_{2i}, d \cdot G) \\
&= \hat{e}(T_1 \cdot CID_i^{**}, r_u \cdot d \cdot G) \cdot \hat{e}(A_{2i}, Q_d) \\
&= \hat{e}(T_1 \cdot CID_i^{**}, r_u \cdot Q_d) \cdot \hat{e}(A_{2i}, Q_d)
\end{aligned}
$$

$= \hat{e}(T_1 \cdot CID_i^{**}, Q_d)^{ru} \cdot \hat{e}(A_{2i}, Q_d)$
$= \hat{e}(T_1 \cdot r_u \cdot CID_i^{**}, Q_d) \cdot \hat{e}(A_{2i}, Q_d)$
$= \hat{e}((T_1 \cdot r_u \cdot CID_i^{**} + A_{2i}), Q_d)$
$= A_{3i}$

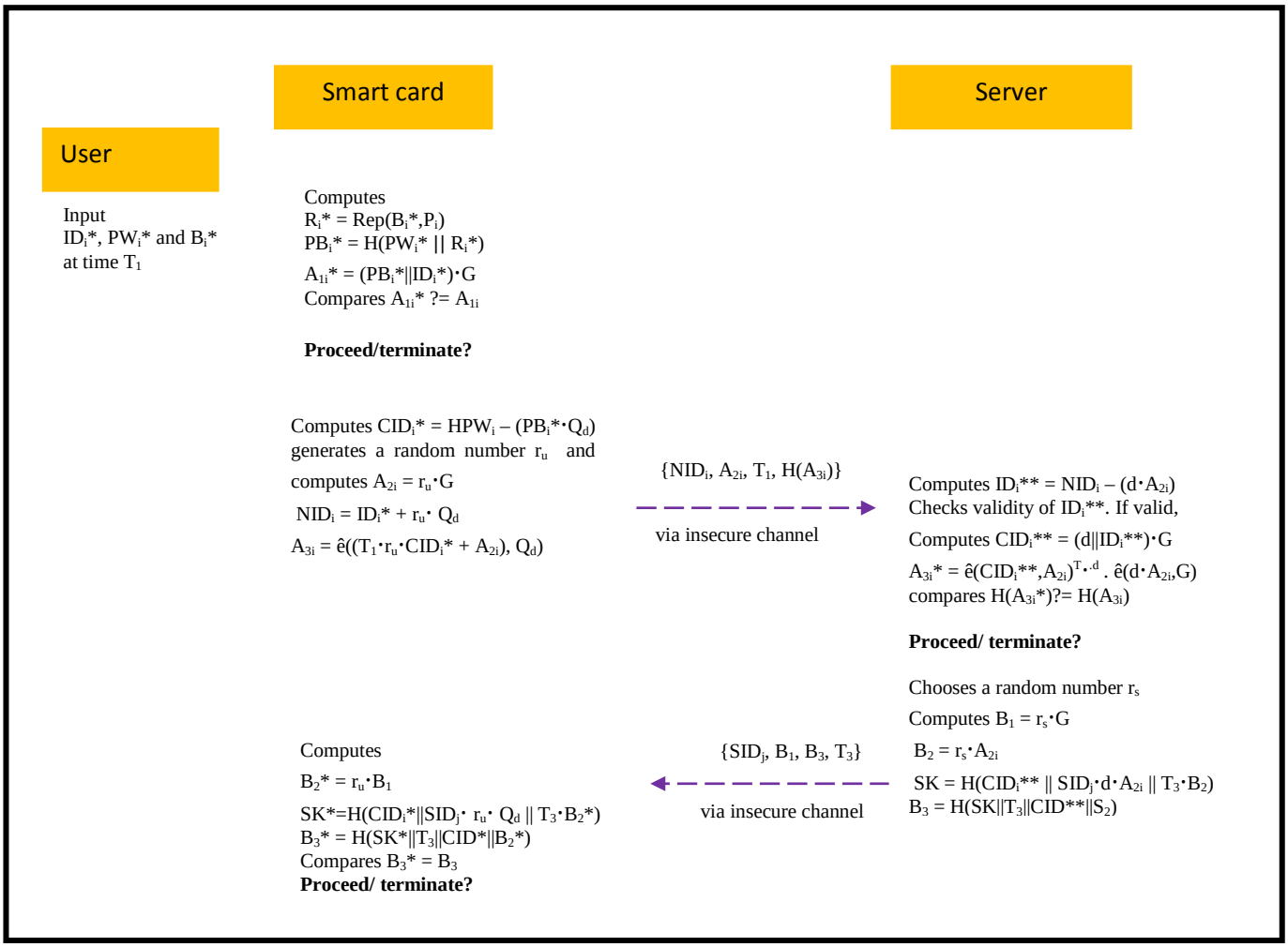

Fig. 2. Mutual Authentication and Key agreement

Further, the server chooses a random number r_s and computes $B_1 = r_s \cdot G$, $B_2 = r_s \cdot A_{2i}$, $SK = H(CID_i^{**} \| SID_j \cdot d \cdot A_{2i} \| T_3 \cdot B_2)$, and $B_3 = H(SK \| T_3 \| CID^{**} \| S_2)$, where SID_j is server's identity and T_3 is a time when the server sent the authentication message.

The server sends an authentication message $\{SID_j, B_1, B_3, T_3\}$ to smart card, which is received at time T_4.

The smart card checks if $(T_4 - T_3) < \Delta T$. If true, the smart card computes $B_2^* = r_u \cdot B_1$, $SK^* = H(CID_i^* \| SID_j \cdot r_u \cdot Q_d \| T_3 \cdot B_2^*)$, and $B_3^* = H(SK^* \| T_3 \| CID^* \| B_2^*)$.

Finally, the smart card compares $B_3^* = B_3$. If both are equal, the smart card authenticates the server; otherwise, login session is terminated.

Mutual authentication and key agreement feature of the scheme are summarized in **Fig. 2**.
Note: SK* = SK is a session key computed by both the user and server for this session.

5.5 Password Change Phase

In this section, we provide the password change procedure for a registered user of the system. If a user wants to change his password for any reason, he inserts his smart card into the terminal and keys in his ID_i^* and password PW_i^* into the terminal, and also gives his fingerprints into the device.
The smart card computes $R_i^* = Rep(B_i^*, P_i)$, $PB_i^* = H(PW_i^* || R_i^*)$, and $A_{1i}^* = (PB_i^* || ID_i^*) \cdot G$.
If $A_{1i}^* \neq A_{1i}$, then the process is terminated; otherwise, the smart card computes $CID_i^* = HPW_i - PB_i^* \cdot Q_d$, and the user is asked to enter new Password PW_{new}.
The smart card computes $PB_{new} = H(PW_{new} || R_i^*)$, $A_{1new} = (PB_{new} || ID_i^*) \cdot G$ and $HPW_{new} = CID_i^* + PB_{new} \cdot Q_d$.
Replace HPW_i with HPW_{new} and A_{1i} with A_{1new} in the smart card. The password is successfully changed.

6. Informal Security Analysis of Proposed Scheme

Security analysis of a scheme determines its efficacy and robustness. In order to achieve all security requirements, this section presents the security features that our scheme provides, followed by all security attacks that our scheme can resist. Based on the capabilities of an attacker as mentioned in the attacker model in section 3, we assume that an adversary has the smart card information $\{HPW_i, A_{1i}, G, Q_d, q, H, E_q(a,b)\}$ and he also traps the communication messages $\{NID_i, A_{2i}, T_1, H(A_{3i})\}$ and $\{SID_j, B_1, B_3, T_3\}$ between the user and server. Here we present the security analysis of our scheme and claim that it is highly secure against the attacks.

6.1 No Verification Table is Needed

In our scheme, the server does not store any secret value in its database. So, in case an adversary somehow accesses the database, there is no chance for him to get/alter the secret values of the user. Thus, due to absence of verification table, our scheme resists the stolen verifier attack.

6.2 Efficient Login Phase

In our scheme, before sending any authentication request to the server, the smart card checks the correctness of ID_i and password PW_i entered by the user. If a legal user by mistake enters the wrong password PW_i^*, the smart card itself terminates the login session. Thus, there is no wastage of computation as well as communication cost.
If $PW_i^* \neq PW_i$, then $PB_i^* = H(PW_i^* || R_i) \neq PB_i$
Therefore, $A_{1i}^* = (PB_i^* || ID_i^*) \cdot G \neq A_{1i}$.
It means that when a smart card compares the computed A_{1i}^* with the stored A_{1i} and finds the inequality, it stops further computation and terminates the login session. In this way, it reduces

the extra overload on the communication channel. Thus, our scheme provides efficient login phase.

6.3 Efficient Password Change Phase

To change the password, the correctness of the password PW_i, user identity ID_i, and R_i are first checked by comparing A_{1i}* with A_{1i} by the smart card itself in a similar way as discussed above. If they match, the user is asked to give new password. The smart card then computes new values of HPW_i and A_{1i} and replaces the old values with new ones. In our scheme, the server is not involved in password change phase. Thus, there is no communication cost for changing the password and the user is free to change his password whenever he wishes.

6.4 Mutual Authentication

In mutual authentication, the user and the server both authenticate each other. They use their own secret keys to compute the authentication messages, which are used to verify their authenticity. In authentication phase, the server computes the following message to authenticate the user by using his private key *d* as $CID_i^{**} = NID_i - d \cdot A_{2i}$, $A_{3i}^* = \hat{e}(CID_i^{**}, A_{2i})^{T1,d} \cdot \hat{e}(d \cdot A_{2i}, G)$. If $H(A_{3i}^*) = H(A_{3i})$, the user is authenticated.

To authenticate the server, the smart card computes the following message using the user's private value $PW_i^* = PW_i$ and $R_i^* = R_i$ as $R_i^* = Rep(B_i^*, P_i)$, $PB_i^* = H(PW_i^* \| R_i^*)$, $CID_i^* = HPW_i - PB_i^* \cdot Q_d$, $B_2^* = r_u \cdot B_1$, $SK^* = H(CID_i^* \| SID_j \cdot r_u \cdot Q_d \| T_3 \cdot B_2^*)$ and $B_3^* = H(SK^* \| T_3 \| CID^* \| B_2^*)$. If $B_3^* = B_3$, the server is authenticated.

6.5 Session Key Agreement

In our scheme, we compute a session key for the current session when a user wishes to communicate with the server. It is to be noted here that the session key is a temporary value, which is accepted in a particular session and it is of no use in any other login session for the user. The session key depends on the temporary value selected by the user and the server. The $SK = H(CID_i^{**} \| SID_j \cdot d \cdot A_{2i} \| T_3 \cdot B_2)$ is the session key computed by the server and the $SK^* = H(CID_i^* \| SID_j \cdot r_u \cdot Q_d \| T_3 \cdot B_2^*)$ is computed by the user. It may be noted that $SK^* = SK$ (proved). The unique key construction for each session ensures the key freshness property.

6.6 Resistance to Denial of Service Attack

In denial of service (DOS) attack, an adversary attempts to prevent a legal user from accessing the services. The adversary usually sends huge forged messages to make the network or server busy all the time. In our scheme, the correctness of the user's secret values are checked before sending the login message for authentication. Furthermore, there is no role of the server in password change phase; thus, reducing the server load and network congestion as well. Thus, there is no chance of the denial of service attack.

6.7 User Anonymity Preservation

In our scheme, the user identity is stored in the encrypted form in the smart card as $HPW_i = CID_i + PB_i \cdot Q_d$, where $CID_i = (ID_i \| d) \cdot G$. Finding ID_i from $(ID_i \| d) \cdot G$ is a problem of the ECDLP,

which is intractable. If somehow the data of the smart card are extracted by the adversary, even then he cannot recover the value of the ID_i. The user identity is also encrypted in authentication message, which is sent on an insecure channel to the server as $NID_i = ID_i^* + r_u \cdot Q_d$. If the adversary intercepts the message and finds NID_i, then, due to the problem of ECDLP, he cannot extract the random number r_u and without knowing r_u he cannot compute ID_i^*. Moreover, knowing $A_{2i} = r_u \cdot G$ and $Q_d = d \cdot G$, getting $d \cdot A_{2i}$ is not feasible due to the CDHP problem. Thus, our scheme preserves user anonymity.

6.8 Resistance to Offline Password/Identity Guessing Attack

In our scheme, the ID_i and PW_i of a user are stored in the smart card in encrypted form and it is not easy to extract them. Therefore, to find the values of ID_i and PW_i, the adversary performs guessing both the values. We have already mentioned that an adversary can get lots of parameters (HPW_i, A_{1i}, NID_i, A_{2i}, B_1, B_3,) from the smart card and the communicating messages during execution of the protocol. Our claim is that the attacker cannot guess and derive both ID_i and PW_i in polynomial time as discussed below.

- **From A_{1i}:** The parameter A_{1i} is defined as $A_{1i} = (PB_i||ID_i) \cdot G$, where $PB_i = H(PW_i||R_i) \cdot$ In this case, the adversary has no knowledge about PW_i, ID_i and $R_i \cdot$ Due to the ECDLP computational problem, he cannot get PB_i or ID_i from $(PB_i||ID_i) \cdot G$. Moreover, due to non-invertible one way hash function, the adversary cannot find PW_i and R_i. If the adversary wants to guess password, he needs to guess other secret values: ID_i and R_i. We however have already discussed in the attacker model (in section 3) that the probability of guessing all these three parameters ID_i, PW_i, and R_i at the same time is $1/2^{18n}$, which is infeasible.
- **From HPW_i:** The parameter HPW_i is defined as $HPW_i = CID_i + PB_i \cdot Q_d$, where $CID_i = (d||ID_i) \cdot G$. Here, we can see that an extra unknown parameter *d* is involved in computation of HPW_i, which cannot be extracted due to the ECDLP computational problem. If the adversary wants to guess all the four values of ID_i, PW_i, R_i, and d, then the probability of guessing these values is $1/2^{18n+m}$, which is very less. Here, we have assumed that *d* has m characters. The probability of guessing the password is enormously negligible.
- **From NID_i:** The attacker can use NID_i to guess ID_i, where $NID_i = ID_i + r_u \cdot Q_d$. However, there are two unknown parameters ID_i and r_u. To verify the guessed ID_i, the adversary also needs to guess r_u. The probability of guessing the identity ID_i is $½^{(6n+64)}$, where the length of the random number is 64 bits, which is infeasible in polynomial time.

6.9 Resistance to User-server Impersonation Attacks

As mentioned in the attacker model (section 3), we assume that an attacker can catch the transmitting messages as and when it is conveyed through the public channel and after making some alteration in a message, he can re-transmit the message for verification. If the re-transmitted message is somehow verified, the attacker can break the security system and access the server, which isnot possible in our scheme as discussed below.

- If the adversary wants to impersonate as a legal user, he eavesdrops the transmitted message $\{NID_i, A_{2i}, H(A_{3i}), T_1\}$ from the insecure channel and tries to construct a new message by changing the random nonce. Suppose, at current time T_1', he selects a random number r_u'.

However, to compute $NID_i = ID_i + r_u' \cdot Q_d$ and $A_{3i} = \hat{e}((T_1' \cdot r_u' \cdot CID_i^* + A_{2i}), Q_d)$, he needs CID_i^* that cannot be computed without ID_i and *d*, and extracting or guessing the ID_i and *d* at the same time is not possible in polynomial time as we have already proved above. Thus, the adversary cannot impersonate a legal user in our scheme.

- If the adversary wants to impersonate the server, he eavesdrops the transmitted message $\{SID_j, B_1, B_3, T_3\}$ from the insecure channel and tries to construct a new message. Suppose, at current time T_3', he selects a random number r_s'. However, to compute $B_3 = H(SK||T_3'||CID^{**}||B_2)$, he needs CID_i^{**} and B_2. He cannot compute $B_2 = r_s \cdot r_u \cdot G$ even after knowing $A_{2i} = r_u \cdot G$ and $B_1 = r_s \cdot G$, due to the CDHP and DDHP problems. Thus, the adversary cannot impersonate the server in our scheme.

The above discussion clearly states that our scheme is well protected against the user-server impersonation attacks and an adversary cannot build any valid messages for transmission to the desired entity.

6.10 Resistance to Privileged Insider Attack

Due to insider attack, several security systems had been broken. It is therefore essential to keep the user's confidential information secret from the server (though the server is trusted). Some insider of the system (system manager or administrator) may use that information with other accounts on other server, as most of the users use the same password for a set of accounts. In our scheme, a user submits the hashed value $PB_i = H(PW_i \,||\, R_i)$ to the server instead of the original PW_i in the registration phase. Thus, an insider cannot extract the user's password due to non-invertible one way function. Moreover, guessing the password is also infeasible due to two unknown parameters, as discussed earlier.

6.11 Resistance to Replay Attack

When an adversary uses the information that he intercepted from the previous transmission to impersonate as a legal user, it is called as a replay attack. In our scheme, we use timestamp as well as random numbers for sending the authentication messages $\{ID_i, A_{2i}, H(A_{3i}), T_1\}$ and $\{SID_j, B_1, B_3, T_3\}$. The adversary cannot extract random numbers from the messages due to the ECDLP problem.

Case 1: If the adversary sends the same authentication message, the tolerable time delay ΔT will be exceeded and the session will be terminated.

Case 2: When the adversary intercepts the message and later sends it at current time T_1' such as $\{NID_i, A_{2i}, H(A_{3i}), T_1'\}$. The server accepts it and computes $ID_i^{**} = NID_i - d \cdot A_{2i}$, $CID_i^{**} = (ID_i^{**}||d) \cdot G$ and $A_{3i}' = \hat{e}(CID_i^{**}, A_{2i})^{T1',d} \cdot \hat{e}(d \cdot A_{2i}, G)$. Here, $H(A_{3i}') \neq H(A_{3i})$, due to different timestamps $T_1' \neq T_1$. Thus, the login session will be terminated by the server and the replay attack is forbidden in our scheme.

6.12 Resistance to Known Session Specific Temporary Information Attack

In our scheme, the session key SK upon which the user and server agreed in a particular session does not leave any information. Thus, it is not easy for an adversary to compute another session key. The session key is not transmitted as a plaintext on an insecure channel, rather it is

computed by the server and the user using their private keys. So, getting SK is very hard for the adversary without the knowledge of the private keys and random values. However, if the adversary somehow gets r_u and r_s, he cannot compute SK without CID_i.
If the adversary somehow gets the session specific temporary values SK, r_u, and r_s, it cannot affect other session keys. Extracting information from the session key is again a problem of the ECDLP. To compute the session key, the adversary needs a fresh random value of the current session and the secret value of the server. Thus, knowing the temporary value of any session, the adversary cannot find the keys of another session.

6.13 Perfect Forward/Backward Secrecy

In perfect forward/backward secrecy, a session key derived from a set of long-term keys (i.e. ID_i and d) will not be compromised even if one of the long-term keys is compromised in future. Here, we assume that the long-term secret key d of the server is disclosed by some means to an attacker and he tries to compute the previous session key $SK = H(CID_i^{**} \| SID_j \cdot d \cdot A_{2i} \| T_3 \cdot B_2) = H(CID_i^{*} \| SID_j \cdot r_u \cdot Q_d \| T_3 \cdot B_2^{*})$, where $CID_i^{**} = (d \| ID_i) \cdot G$ and $B_2 = r_s$ $A_{2i} = r_u \cdot B_1$. However, knowing only the secret key d, the attacker cannot compute the previous session key due to other secret parameters, namely, ID_i, r_s and r_u as it has already been proved that extracting these values is not possible due to the ECDLP computational problem. Furthermore, if we assume that the session key of the protocol is compromised to the attacker, the attacker tries to compute the previous session key. The attacker cannot extract any secret parameters such as d and B_2 from the session key $SK=H(CID_i^{*} \| SID_j \cdot r_u \cdot Q_d \| T_3 \cdot B_2) = H(CID_i^{**} \| SID_j \cdot d \cdot A_{2i} \| T_3 \cdot B_2)$ due to non-invertible one way hash function and hence he cannot compute the previous session key. Thus, our scheme preserves the perfect forward/backward secrecy property.

6.14 Resistance to Stolen Smart Card Attack

Suppose an attacker steals the smart card and somehow extracts the smart card parameters $\{HPW_i, A_{1i}, G, Q_d, p, H, E_q(a,b)\}$ and wants to generate a login message $\{NID_i, A_{2i}, H(A_{3i}),T_1\}$. To compute A_{3i}, the attacker needs $CID_i^{*} = HPW_i - PB_i^{*} \cdot Q_d$ and for computing PB_i^{*}, the user ID_i, password, and biometric value are needed. In some schemes, if the adversary finds the smart card, he can change the password by password guessing attack. However, in our scheme, ID_i is also kept secret and it has already been proved that guessing attack is infeasible to guess ID_i, PW_i, and B_i. It means that even after getting the smart card's parameters the adversary cannot extract the correct values of ID_i, PW_i and R_i to generate any valid message. Thus, the stolen smart card's attack is not effective in our scheme.

7. Authentication Proof based on BAN logic

In this section, we apply the BAN logic, a tool for analyzing authentication schemes [26]. The BAN - logic uses three objects: principals, encryption keys, and formulas (also called statements for identifying messages with a statement). We use symbols M and N as principals, X and Y range over statements, and K represents the cryptographic key.
We use same notations as in the BAN-logic for our demonstration.

$M \mid\equiv X$: The principal M believes a statement X.

$M \lhd X$: The principal M sees the statement X.

$M \mid\sim X$: M once said X.

$M \Rightarrow X$: M has jurisdiction over X. (Used when the principal has delegated authority over some statement).

$\#(X)$: X is fresh, that is, no principal sent X in a message before the current run of the protocol.

$M \overset{K}{\longleftrightarrow} N$: M and N communicate using shared K. Moreover, K will never be discovered by any principal except M and N, or a principal trusted by either M or N.

$\{X\}_K$: This stands for X encrypted under the K.

$< X >_Y$: This stands for X combined with Y.

$(X)_K$: This stands for X hashed with key K.

$K \mid\rightarrow M$: K is the public key of M and M has a corresponding secret key K^{-1}.

Besides, we present some main logical BAN-logic postulates for proving our scheme.

Message meaning rule: $\dfrac{M \mid\equiv M \overset{K}{\longleftrightarrow} N, M \lhd \{X\}_K}{M \mid\equiv N \mid\sim X}$,

Nonce verification rule: $\dfrac{M \mid\equiv \#(X), M \mid\equiv (N \mid\sim X)}{M \mid\equiv N \mid\equiv X}$

Jurisdiction rule: $\dfrac{M \mid\equiv N \Rightarrow X, M \mid\equiv N \mid\equiv X}{M \mid\equiv X}$

Freshness rule: $\dfrac{M \mid\equiv \#(X)}{M \mid\equiv (X, Y)}$

Believe rule: $\dfrac{M \mid\equiv N \mid\equiv (X, Y)}{M \mid\equiv X, M \mid\equiv Y}$

All authentication schemes need to achieve four main goals between user U_i and server S_j. Following are the required goals:

$G_1 : U_i \mid\equiv S_j \mid\equiv U_i \overset{SK}{\longleftrightarrow} S_j$

$G_2 : S_j \mid\equiv U_i \mid\equiv U_i \overset{SK}{\longleftrightarrow} S_j$

$G_3 : S_j \mid\equiv U_i \overset{SK}{\longleftrightarrow} S_j$

$G_4 : U_i \mid\equiv U_i \overset{SK}{\longleftrightarrow} S_j$

Following are the assumptions made about the initial state of the scheme to analyze the proposed scheme:

$A_1 : U_i \mid\equiv \#r_u$

$A_2 : S_j \mid\equiv \#r_s$

$A_3 \quad U_i \mid\equiv \#T_1$

$A_4: S_j \mid\equiv \#T_3$

$A_5: U_i \mid\equiv S_j \mid\equiv U_i \xleftrightarrow{CID_i} S_j$

$A_6: S_j \mid\equiv U_i \mid\equiv U_i \xleftrightarrow{CID_i} S_j$

$A_7: U_i \mid\equiv Q_d \mid \rightarrow S_j$

$A_8: S_j \mid\equiv U_i \mid\equiv ID_i$

$A_9: U_i \mid\equiv S_j \mid\equiv SID_j$

$A_{10}: S_j \mid\equiv U_i \Rightarrow U_i \xleftrightarrow{SK} S_j$

$A_{11}: U_i \mid\equiv S_j \Rightarrow U_i \xleftrightarrow{SK} S_j$

$A_{12}: U_i \mid\equiv S_j \Rightarrow B_2$

$A_{13}: S_j \mid\equiv U_i \Rightarrow A_2$

We now analyze our scheme's idealized form based on the BAN logic rules and the assumptions:

Message 1: $U_i \rightarrow S_j$: <NID_i, A_{2i}, $H(A_3)$, T_1>
According to seeing rule
R_1: $S_j \lhd$ <$\{ID_i\}_{Qd,} A_2, (\{T_{1,} A_{2i}\}_{CIDi}), T_1$>

According to A_5 and R_1 and message meaning rule, we get
R_2: $S_j \mid\equiv U_i \mid\sim(ID_i, T_1, A_{2i})$

According to A_1, A_4 and R_2 and freshness-conjuncatenation rule and nonce verification rule is applied, we get
R_3: $S_j \mid\equiv U_i \mid\equiv(ID_i, T_1, A_{2i})$

According to A_5, A_6, A_9 and R_3 and Believe rule
R_4: $S_j \mid\equiv U_i \mid\equiv U_i \xleftrightarrow{SK} S_j$ (Goal G_2)

According to A_{10} and R_4 and Jurisdiction rule
R_5: $S_j \mid\equiv U_i \xleftrightarrow{SK} S_j$ (Goal G_3)

Message 2: $S_j \rightarrow U_i$: < SID_j, B_1, B_3, T_3>

According to seeing rule
R_6: $U_i \lhd$ <<SID_j, $r_u \cdot G$,(B_2, $U_i \xleftrightarrow{SK} S_j$, T_3)$_{CIDi}$, T_3>

According to A_6, A_7 and R_6 and message meaning rule, we get
R_7: $U_i \mid\equiv S_j \mid\sim$ (SID_j, B_2, $U_i \xleftrightarrow{SK} S_j$, T_3)

According to A_2, A_3 and R_7 and freshness-conjuncatenation rule and nonce verification rule is applied, we get

R_8: $U_i \mid\equiv S_j \mid\equiv (SID_j, B_2, U_i \xleftrightarrow{SK} S_j, T_3)$

Therefore, according to Believe rule:

R_9: $U_i \mid\equiv S_j \mid\equiv U_i \xleftrightarrow{SK} S_j$ (Goal G_1)

According to A_{11} and R_9 and Jurisdiction rule

R_{10}: $U_i \mid\equiv U_i \xleftrightarrow{SK} S_j$ (Goal G_4)

According to A_{12}, A_9 and R_8 and Jurisdiction rule

R_{12}: $U_i \mid\equiv (SID_i, B_2)$

Since, CID_i, SID_j, B_{2i} are the main factors to compute, SK for smart card, According to R_{12}, A_6, A_7 and message meaning rule

R_{13}: $S_j \mid\equiv U_i \mid\equiv U_i \xleftrightarrow{SK} S_j$ (Goal G_2)

According to A_{10} and R_{13} and Jurisdiction rule

R_{14}: $S_j \mid\equiv U_i \xleftrightarrow{SK} S_j$ (Goal G_3)

The above discussion clearly proves the stated objectives using the BAN logic and it is also proved that the proposed protocol achieves mutual authentication and session key agreement between the U_i and S_j.

8. Simulation of Proposed Scheme using AVISPA Tool

We first briefly discuss about the AVISPA tool and then followed by the basic specification and simulation result of the proposed scheme.

8.1 Brief Description of AVISPA Tool

The Automated Validation of Internet Security Protocols and Applications (AVISPA) [33] is a freeware tool for formal security verification of the security protocols to check if a given security protocol is SAFE or UNSAFE. The basic architecture of the AVISPA tool is shown in **Fig. 3**.

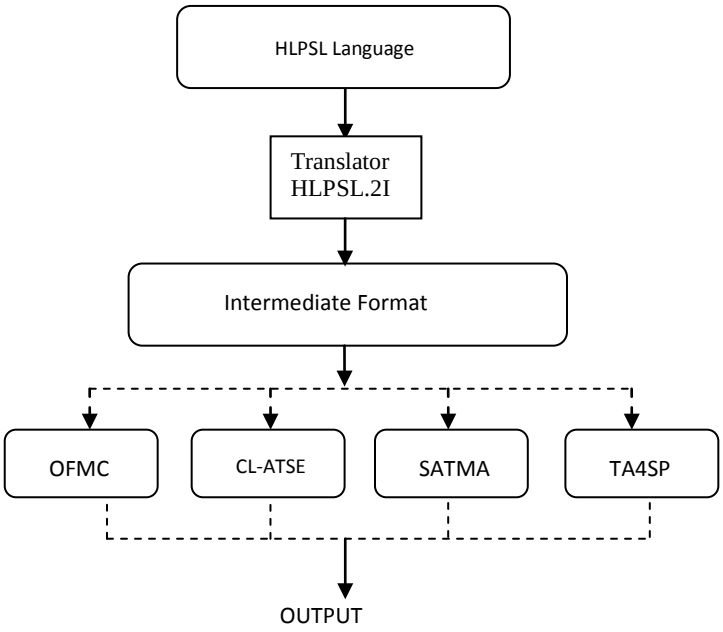

Fig. 3. Basic Architecture of *AVISPA* Tool

The AVISPA, a role-oriented language, is based on the Dolev-Yao [34] intruder model in which each participant plays a role during the protocol execution. It implements four different back-ends and abstraction based methods, called as On-the fly Model-Checker (OFMC), Constraint Logic based Attack Searcher (CL-AtSe), SAT-based Model-Checker (SATMC), and Tree Automata based on Automatic Approximations, for the Analysis of Security Protocol (TA4SP). Based on these four back-ends, the output format (OF) is generated and the successful execution OF reports if the protocol is safe or unsafe and under what condition the output is obtained. The specifications for the protocol to be evaluated are written in High Level Protocol Specification Language (HLPSL) and they are translated into a low-level specification by a translator, called hlpsl2if, that generates the specifications into an intermediate format, called intermediate format (IF), a lower level language, that can directly be read by the back-ends of the AVISPA tool. To analyze a given cryptographic protocol with the AVISPA, the following steps are executed:

Step 1. The protocol is coded in the role based HLPSL specification, which describes each participant's role and the composition roles for representing the scenarios of basic roles.

Step 2. Using the translator HLPSL2IF, the code is translated into Intermediate Format (IF) that contains some information about IF syntax for back-ends, the description of mathematical properties of operators (e.g., exponentiation, bit-wise XOR, etc.), and the intruder's behavior.

Step 3. The IF specifications are given to the back-ends of the AVISPA tool to analyze if there are any active or passive attacks.

8.2 Brief Specification of Proposed Scheme

```
role user(
Ui,Sj:agent,
SK1:symmetric_key,
%Hishashfunction
H:hash_func,
Bilinear,Subtract,Add,Product,Mul:hash_func,
Snd,Rcv:channel(dy))
played_by Ui
def=
local State: nat,
IDi,PWi,Bi, Ri, Pi, PBi, A1i,A3i, G, D, Qd, CIDi,HPWi, B2, Ru,Rs,SK:text,
NIDi, A2i, Ti,SIDj,B1,B3, T1,T3:message,
Inc:hash_func
const user_server,server_user, subs1,subs2,subs3,subs4,subs5,subs6,subs7:protocol_id
init State:=0
transition
% Start registration phase of the user
1.State=0/\Rcv(start)=|>
State':=1/\Ri':=new() /\Pi':=new() /\IDi':=new() /\PWi':=new()
/\PBi':= H(PWi'.Ri')
/\Snd({IDi'.PBi'.Pi'}_SK1)
%Send registration request to the RC
/\secret({IDi}, subs1, {Ui,Sj})
/\secret({PWi,Bi,Ri}, subs2, Ui)
2.State=1/\Rcv({HPWi.A1i.G.Qd.Pi}_SK1)=|>
%Receives smart card information from RC
State':=2/\ Ru':=new() /\T1':= new()
/\CIDi':= Subtract(HPWi,Product(PBi,Qd))
/\A2i':= Product(Ru',G)
/\NIDi':= Add(IDi,Product(Ru',Qd))
/\A3i':= Bilinear(Add(Product(T1,Product(Ru',CIDi')),A2i'),Qd)
/\Snd(NIDi, A2i, T1, H(A3i))
%Send login message to the Sj
/\witness(Ui,Sj,user_server,Ru')
/\secret({Ru'},subs3,{Ui})
/\secret({A3i},subs4,{Ui,Sj})
/\request(Ui,Sj,user_server,Ru)
3.State=2/\Rcv(SIDj,B1,B3,T3)=|>
%Receives messages from Sj
State':=3
```

Fig. 4. Role specification in HLPSL for user U_i of our scheme

To validate and examine the security properties of our proposed scheme, we implement it using the HLPSL language in the AVISPA tool. The role specifications of the user Ui and server Sj are given in **Figs. 4 & 5**, respectively. The proposed scheme is analyzed in the OFMC and CL-AtSe back-ends, and the corresponding results are given in **Figs. 7 & 8**. From these simulation results, the proposed scheme indeed shows its strong security assurance against both the passive and active attacks. The type declaration channel(dy) means that the channel is for the Dolev-Yao threat model [34]. The Bilinear, Product, Subtract, Add, Mul, and H represents bilinear operation, scalar point multiplication, Point subtraction, point addition, scalar multiplication, and hash functions, respectively.

In **Fig. 4**, we have presented the role for the user Ui. Here, Transition 1 starts with the registration of the user. For this, the Ui initially sends the registration message *Snd(IDi'.PBi'.Pi')* to server Sj through a secure channel using *Snd()* operation and symmetric key *SK1*. The declaration *secret({IDi}, subs1, {Ui,Sj})* specifies that the $\{ID_i\}$ is known to user and server only, whereas the *secret({PWi, Bi, Ri}, subs2, Ui)* specifies that the *(PWi, Bi, Ri)* is known only to user Ui. In transition 2, the user Ui receives the smart card information *Rcv({HPWi.A1i.G.Qd.Pi}_SK1)* using *Rcv()* operation securely and generates a random nonce R_u' and timestamp *T1'* using *new()* operation. The user Ui sends *Snd(NIDi,A2i,T1,H(A3i)* to server Sj through a public channel.

The declaration *witness(Ui,Sj,user_server,Ru')* indicates that the user Ui freshly generated the value R_u' for S_j and the declaration *request(Ui,Sj,user_server,Ru')* means that the Sj authenticates user Ui. Furthermore, the declaration *secret({Ru'},subs3,{Ui})* says that the random number Ru' is only known to Ui and the declaration *secret({A3i},subs4,{Ui,Sj})* says that {A3i} is known to {*Ui, Sj*} only. In transition 3, it says about the authentication phase, the user Ui receives *Rcv(SIDj,B1,B3,T3)* through a public channel and after receiving it, the user Ui computes the session key *SK':= H(CIDi.Product(SIDj,Product(Ru,Qd)).Product(T3,B2))* of the protocol.

In **Fig. 5**, we have presented the role of the Server Sj. In transition 1, the Sj chooses a generator G and generates own secret D' using *new()* operation and computes the public key *Qd = Product(D.G).* In transition 2, Sj receives *Rcv({IDi'.PBi'.P_i'}_SK1)* securely from the user U_i as the registration request. After computing the smart card parameters, Sj sends *Snd({HPWi.A1i.G.Qd.Pi}_SK1)* securely to user Ui. The declaration *secret({D},subs5,{Sj})* shows that D is secretly known only to Sj.

```
role server(Ui,Sj:agent,
SK1: symmetric_key, H:hash_func,
Bilinear,Subtract,Add,Product,Mul:hash_func,
Snd,Rcv: channel(dy))
played_by Sj
def=
local State : nat,
IDi,PWi,Bi, Ri, Pi, PBi, A1i,A3i, G, D,Qd,CIDi, HPWi, B2, Ru,Rs,SK:text,
NIDi, A2i, Ti,SIDj,B1,B3, T1,T3:message,
Inc: hash_func
const user_server,server_user,subs1,subs2,subs3,subs4,subs5,subs6,subs7:protocol_id
init State:=0
transition
1.State=0/\Rcv(start)=|>
State':=1
/\SIDj':=new() /\D':=new() /\G':=new()
/\Qd':=Product(D',G')
/\secret({D},subs5,{Sj})
2.State=1/\Rcv({IDi.PBi.Pi}_SK1)=|>
State':=2
/\CIDi':= Product((d.IDi),G)
/\HPWi':= Add(CIDi,Product(PBi,Qd))
/\A1i':= Product((PBi.IDi),G)
/\Snd({HPWi.A1i.G.Qd.Pi}_SK1)
%Send smart card information securely
3.State=3/\Rcv(NIDi, A2i, T1, H(A3i) )=|>
%Receives login message from user
State':=3/\Rs':=new() /\T3':=new()
/\IDi':= Subtract(NIDi,Product(D,A2i))
/\CIDi':= Product((d.IDi'),G)
/\A3i':= Product(exp(Bilinear(CIDi,A2i),Mul(T1,d)),Bilinear(Product(D,A2i),G))
/\request(Ui.Sj,user_server,Ru)
/\B1':= Product(Rs',G)
 /\B2':= Product(Rs',A2i)
 /\SK':= H(CIDi'.Product(SIDj,Product(D,A2i)).Product(T3,B2))
/\B3':= H(SK'.T3.CIDi'.B2')
/\Snd(SIDj, B1',B3',T3')
%Send message to the sensor node
/\witness(Sj,Ui,server_user,Rs')
/\secret({Rs'},subs6,{Sj})
/\secret({B3'},subs7,{Sj,Ui})
/\request(Ui.Sj,server_user,Rs)
 end role
```

Fig. 5. Role specification in HLPSL for the server Sj of our scheme

```
role session(Ui,Sj:agent,
SK1: symmetric_key,
H,Bilinear,Subtract,Add,Product,Mul: hash_func)
def=
local SI,SJ,RI,RJ,TI,TJ,PI,PJ: channel(dy)
composition
user(Ui,Sj,SK1,H,Bilinear,Subtract,Add,Product,Mul,SI,RI)
/\server(Sj,Ui,SK1,H,Bilinear,Subtract,Add,Product,Mul,SJ,RJ)
end role

role environment()
def=
const ui,sj: agent, sk1: symmetric_key,
h,bilinear,subtract,add,product,mul:hash_func,
idi,pwi,bi, ri, pi, pbi, a1i,a3i, g, qd,cidi,d, hpwi, b2, ru,rs,sk:text,
nidi, a2i, ti,sidj,b1,b3, t1,t3:message,
user_server_ru,server_user_rs,subs1,subs2,subs3,subs4,subs5,
subs6,subs7:protocol_id
%Represents Intruder knowledge
intruder_knowledge={hpwi,nidi, a1i,a2i, t1,sidj,b1,qd,b3,t3,h(a3i)}
composition
session(ui,sj,sk1,h,bilinear,subtract,add,product,mul)
/\session(ui,sj,sk1,h,bilinear,subtract,add,product,mul)
end role
goal
%Verifies secrecy of the confidential information
secrecy_of subs1    secrecy_of subs2
secrecy_of subs3    secrecy_of subs4
secrecy_of subs5    secrecy_of subs6
secrecy_of subs7
%Verifies authenticity of the random numbers used in the protocol
authentication_on user_server_ru
authentication_on server_user_rs
end goal
environment()
```

Fig. 6. Role specification in HLPSL for the session and environment of our scheme

In transaction 3, the Sj receives *(NIDi,A2i,T1,H(A3i))* from the user Ui through a public channel. After computing the secret values, it generates a random number Rs' and timestamp T3' with the help of *new()* operation. The Sj computes authentication message and sends *Snd(SIDj,B1',B3',T3')* to the user Ui through a public channel. Here, the declaration *secret({Rs'},subs6,{Sj})* says that the parameter Rs' is known only to Sj. Moreover, *witness(Sj,Ui,server_user,Rs')* shows that Sj freshly generates Rs' for the user Ui and *request(Ui.Sj,server_user,Rs)* shows that the Sj authenticates Ui.

In **Fig. 6**, we have presented the role for the session, and the roles for the goal and environment. In session segment, all the basic roles including the roles for Ui and Sj are given along with actual arguments. The environment section contains the global constant and composition of one or more sessions. The intruder knowledge is also given in this section. It is clearly shown that all the transmitted messages between the entities and smart card parameters are provided. To make

this protocol SAFE, 7 secrecy goals and two authentications are provided between the goal and the end goal that are to be verified in the environment section, which is given as follows:

Security Goals

1. The secrecy_of subs1 represents that IDi is kept secret to {Ui, Sj} only.
2. The secrecy_of subs2 represents that {PWi, Bi, Ri} is known only to Ui.
3. The secrecy_of subs3 represents that the random Ru of the Ui is kept secret to {Ui} only.
4. The secrecy_of subs4 represents that the parameter A3i is known only to {Ui, Sj}.
5. The secrecy_of subs5 represents that the secret parameter D is known only to Sj.
6. The secrecy_of subs6 represents that the random number (Rs) generated by Sj is known only to Sj.
7. The secrecy_of subs7 represents that the parameter B3 is known to {Ui, Sj} only.

Authentication goal

1. The authentication_on user_sensor_ru represents that the Ui generates a random number ru and if the Sj receives it securely through the message, the Sj then authenticates Ui.
2. The authentication_on server_user_rs represents that the Sj generates a random number rs and if Ui receives it securely through the message, Ui then authenticates Sj.

8.3 Simulation Results

The simulation results for formal security verification of our scheme using OFMC and CL-AtSe back-end are shown in Figs. 7 and 8, respectively. It is clear from the SUMMARY (**Figs. 7&8**) of results under OFMC and CL-AtSe back-ends that our method is SAFE. As a result, our scheme is secure against the passive and active attacks such as the replay and man-in-the-middle attacks.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/avispa/web-interface-
computation/./tempdir/workfileELhyfT5zJ5.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.14s
  visitedNodes: 16 nodes
  depth: 4 plies
```

Fig. 7. Simulation result for OFMC back-end

```
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL
PROTOCOL
  /home/avispa/web-interface-
computation/./tempdir/workfileELhyfT5zJ5.if
GOAL
  As Specified
BACKEND
  CL-AtSe
STATISTICS
  Analysed   : 0 states
  Reachable  : 0 states
  Translation: 0.04 seconds
  Computation: 0.00 seconds
```

Fig. 8. Simulation result for CL-AtSe back-end

9. Performance Analysis

In this section, we present a comparative study of our scheme along with other related schemes. The measure of our comparisons is the communication cost (refer **Fig.9**), computation cost (refer **Table 2**), and security features (refer **Table 3**).

For 163-bits elliptic curve cryptosystems and 1024-bits RSA security level, one scalar multiplication of elliptic curve point is roughly 5–15 times as fast as the RSA signing operation depending on the optimization and platform [23]. Also, one MD5/SHA operation is roughly 10 times as fast as one DES encryption/decryption operation and one DES encryption/decryption operation is roughly 1000 times as fast as the 1024-bit RSA signing operation. For fair comparisons, we assume that the identifications can be represented with 32 bits, the size of a timestamp is 32 bits, a point on an elliptic curve can be represented with 163 × 2 = 326 bits, the output size of the secure one-way hash functions is 160 bits, the size of a random number is 64 bits, and the size of an exponent result is 1024 bits. Thus, the communication cost of our scheme is 326+326+32+160 +32+326+160+32 =1394 (refer **Fig.9**). We find that our scheme needs very less communication cost as compared to the schemes [8,22] since we have used the ECC to compute the authentication message. The ECC takes fewer bits as compared to the RSA because it uses exponential function. However, our scheme requires higher cost as compared to the schemes [9,10,13,17,21] since these schemes have not considered the following attacks such as replay attack, insider attack, forward secrecy attack, and denial of service attack. Moreover, the schemes [9,13] do not provide mutual authentication between a user and the server (refer **Table 3**). In authentication schemes, the security is of prime concerned; therefore, paying a little more cost for gaining more security is justifiable.

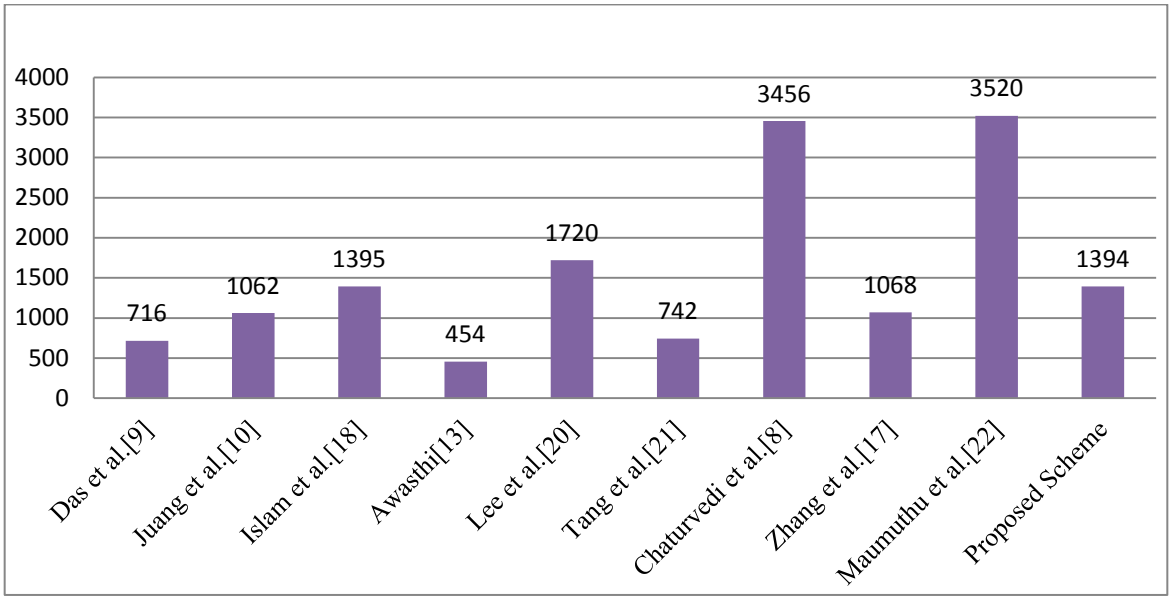

Fig. 9. Graphical representation of communication cost of various schemes

Table 2. Computational cost in Login and Authentication phases of various schemes

Schemes	Computational Cost	
	User Side	Server Side
Das et al.[9]	2PM + 1H	2P + 1PA + 1H
Juang et al.[10]	2PM + 1BP + 1EN + 3H +3C	1PM + 1BP + 2EN + 4H +3C
Islam et al.[18]	3PM + 1EN + 1PA + 2H	2PM +2BP + 1EN + 1PA + 2H
Awasthi [13]	3PM + 2PA + 1EN	1PM + 2BP +1EN
Lee et al. [20]	4H + 1PA + 3PM +4C +EN	2BP + 1EN +1PM +2H + 1PA
Tang et al. [21]	3H + 2PM +10C + 1X	3H + 1PM + 9C
Chaturvedi et al. [8]	3E+ 4H +13C + 2X	3E+ 3H +13C + 1X
Zhang et al. [17]	3PM +4H + 13C + 1X	3PM + 5H + 15C + 2X
Maumuthu et al. [22]	5E + 6H + 6X +1C	4E + 4H + 6X+3C
Proposed Scheme	1BP + 8PM + 3PA + 6C+4H	2BP + 1PA + 7PM+6C+3H

In **Table 2**, we have presented the computational cost of our scheme along with other related schemes [8-10,13,17,18,20-22]. Here, PM, PA, H, C, BP, EN, X, E represent the time Complexity of point multiplication on the Elliptic Curve, point addition on Elliptic Curve, Hash function, Concatenation, Pairing operation, Enc/Dec, XOR operation and Exponentiation, respectively.

In Table 3, we have presented a comparative study of the security features for our scheme and other related schemes. As evident from Table 3, our scheme provides maximum security features as compared to the schemes under consideration. We can also see that the schemes having same security features in **Table 3** [8,22] take much more communication cost (refer **Fig. 9**) to achieve these security goals, which makes our scheme better than other schemes.

Table 3. Security Features of Various Schemes

Schemes	Das et al [9]	Juang et al.[10]	Islam et al. [18]	Awasthi [13]	Lee et al. [20]	Tang et al. [21]	Chaturvedi et al. [8]	Zhang et al. [17]	Maumuthu et al. [22]	Proposed Scheme
Resist Stolen verifier attack	Y	Y	N	Y	Y	Y	Y	N	Y	Y
Efficient login phase	N	N	N	N	N	N	Y	N	Y	Y
Efficient Password Change Phase	N	N	N	N	N	N	Y	N	Y	Y
Mutual Authentication	N	Y	Y	N	Y	Y	Y	Y	Y	Y
Session Key agreement	N	Y	Y	N	N	N	Y	Y	Y	Y
Denial of service	N	N	N	N	N	N	Y	N	Y	Y
User anonymity	N	Y	N	N	N	Y	Y	Y	Y	Y
Withstand offline password guessing attack	N	Y	N	N	Y	Y	Y	Y	Y	Y
Resist User Impersonation attack	N	Y	N	N	Y	Y	Y	Y	Y	Y

Resist Server Impersonation attack	N	Y	Y	N	Y	Y	Y	Y	Y	Y
Resist Insider Attack	N	Y	N	N	Y	Y	Y	N	Y	Y
Resist Replay Attack	N	Y	N	N	Y	Y	Y	Y	Y	Y
Resist session key disclosure attack	-	Y	Y	-	-	-	Y	Y	Y	Y
Forward secrecy	N	N	Y	N	N	N	Y	N	Y	Y
Resist Smart card loss attack	N	Y	N	N	Y	Y	Y	-	Y	Y

Here, Y = Yes, N =No, - =Not Applicable

10. Conclusion

In this paper, we have discussed a new remote login authentication scheme using the bilinear property of a elliptic point and fingerprint that achieves various secure goals and requirements. In this scheme, a user and the sever both authenticate each other to enhance its security. A user can choose and change his password at any time, whenever he wishes. No wastage of communication cost takes place in our scheme if wrong password is entered and the communication cost is also saved during the password change phase. Using elliptic point computation makes the scheme fast as it needs fewer bits as compared to the exponentiation. The bilinear property, use of biometric, and the design of algorithm make it very secure. It is suitable for the applications where high security is required.

References

[1] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770-772, 1981. Article (CrossRef Link).

[2] C. T. Li, M. S. Hwang, "An efficient biometrics-based remote user authentication scheme using smart cards," *Journal of Network and Computer Applications*, vol. 33, no. 1, pp. 1–5, 2010. Article (CrossRef Link).

[3] A. Das, "Analysis and improvement on an efficient biometric-based remote user authentication scheme using smart cards," *Information Security*, IET, vol.5, no.3, pp. 145–151, 2011. Article (CrossRef Link).

[4] C.C. Lee, R.X. Chang, L.A. Chen, "Improvement of Li-Hwang's biometrics-based remote user authentication scheme using smart cards," *WSEAS Transactions on Communications*, vol. 10, no. 7, pp. 193–200, 2011.

[5] Y. An, "Security analysis and enhancements of an effective biometric-based remote user authentication scheme using smart cards," In: *BioMed Research International*, 2012. Article (CrossRef Link).

[6] S. Kumari, M. K. Khan, X. Li, "An improved remote user authentication scheme with key agreement," *Computers and Electrical Engineering*, vol. 40, no. 6, 1997–2012, 2014. Article (CrossRef Link).

[7] X. Li, J. Niu, Z. Wang, C. Chen, "Applying biometrics to design three-factor remote user authentication scheme with key agreement," *Security and Communication Networks*, 2013.

[8] A. Chaturvedi, D. Mishra, S. Mukhopadhyay, "Improved Biometric-Based Three-factor Remote User Authentication Scheme with key Agreement Using Smart Card," *Lecture Notes in Computer Science*, 8303, pp. 63-77, 2013. Article (CrossRef Link).

[9] M.L. Das, A. Saxena, V. P. Gulati, D.B., Phatak, "A novel remote client authentication protocol using bilinear pairings," *Computer and Security*, vol. 25, no. 3, pp. 184–189, 2006. Article (CrossRef Link).

[10] W. S. Juang, W.K. Nien, "Efficient password authenticated key agreement using bilinear pairings," *Mathematical and Computer Modelling*, Elsevier, vol. 47, (11-12), pp. 1238–1245, 2006. Article (CrossRef Link).

[11] G. Fang, G. Huang, "Improvement of recently proposed Remote User Authentication Schemes," http://eprint.iacr.org/2006/200.pdf.

[12] D. Giri, P. D. Srivastava, "An Improved Remote User Authentication Scheme with Smart Card using Billinear Pairings," http://eprint.iacr.org/2006/274.pdf.

[13] A.K. Awasthi, "An improved remote user authentication scheme with smart cards using bilinear pairings," *International Journal of Applied Mathematics and Computation*, vol. 4, no.4, pp. 382–389, 2012. Article (CrossRef Link).

[14] E. J. Yoon, Y. N. Shin, I.S. Jeon, K. Y. Yoo, "Robust mutual authentication with a key agreement scheme for the session initiation protocol," *IETE Technical Review*, vol. 27, no. 3, pp. 203–213, 2010. Article (CrossRef Link).

[15] Q. Xie, "A new authenticated key agreement for session initiation protocol," *Int J Commun Syst*, vol. 25, no. 1, pp. 47–54, 2012. Article (CrossRef Link).

[16] M. S. Farash, M. A. Attari, "An enhanced authenticated key agreement for session initiation protocol," *Information Technology And Control*, vol. 42, no. 4, pp. 333–342, 2013. Article (CrossRef Link).

[17] Z. Zhang, Q. Qi, N. Kumar, N. Chilamkurti, H. Y. Jeong, "A secure authentication scheme with anonymity for session initiation protocol using elliptic curve cryptography," *Multimedia Tools and Application*, Springer, 2014.

[18] S. H. Islam, G. P. Biswas, "Design of improved password authentication and update scheme based on elliptic curve cryptography," *Mathematical and Computer Modelling*, Elsevier, vol. 57, (11-12), pp. 2703–2717, 2013. Article (CrossRef Link).

[19] T. Li, "A new password authentication and user anonymity scheme Based on elliptic curve cryptography and smart card," *IET Information Security*, vol. 7, no.1, pp. 3–10, 2013. Article (CrossRef Link).

[20] C. Lee, C.T. Li, C.Y. Weng, J.J. Jheng, X.Q., Zhu, Y.R. Zhang, "Cryptanalysis and Improvement of an ECC-Based Password Authentication Scheme Using Smart Cards," *Lecture note in computer Science, Springer*, 8300, pp. 338–348, 2013. Article (CrossRef Link).

[21] H. B. Tang, X. S. Liu, L. Jiang, "A Robust and Efficient Timestamp-based Remote User Authentication Scheme with Smart Card Lost Attack Resistance," *International Journal of Network Security*, vol. 15, no. 6, pp. 426-434, 2013.

[22] M. Karuppiah, R. Saravanan, "A secure remote user mutual authentication scheme using smart cards," *Jouranal of information security and application, Elsevier*, vol. 19, no. 11, pp. 282-294, 2014. Article (CrossRef Link).

[23] K. Lauter, "The advantages of elliptic curve cryptography for wireless security," *IEEE Wireless Communications*, vol. 11, no.1, pp. 62–67, 2004. Article (CrossRef Link).

[24] Dodis, Yevgeniy, L. Reyzin, A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," *Advances in cryptology-Eurocrypt 2004. Springer Berlin Heidelberg*, 2004. Article (CrossRef Link).

[25] Boyen, Xavier, 'Reusable cryptographic fuzzy extractors," *Proceedings of the 11th ACM conference on Computer and communications security. ACM*, 2004. Article (CrossRef Link).

[26] M. Burrows, M. Abadi and R Needham, "A logic of authentication," *ACM Transactions on Computer System*, vol. **8**, pp. 18-36, 1990. Article (CrossRef Link).

[27] Y.F. Chang, S.H. Yu, D.R. Shiao, "A uniqueness-and anonymity preserving remote user authentication scheme for connected health care," *J. Med. Syst. vol.*37, no. 2, 9902, 2013. Article (CrossRef Link).
[28] F. Wen, and X. Li, "An improved dynamic id-based remote user authentication with key agreement scheme," *Computers & Electrical Engineering*, vol. 38, no. 2, pp. 381–387, 2012. Article (CrossRef Link).
[29] Juels, and M. Sudan, "A fuzzy vault scheme," *Designs, Codes and Cryptography*, vol. 38, no. 2, pp. 237-257, 2006. Article (CrossRef Link).
[30] Juels, and M. Wattenberg, "A fuzzy commitment scheme," *In Proceedings of the 6th ACM conference on Computer and communications security*, pp. 28-36, 1999. Article (CrossRef Link).
[31] Li, J. Hu, J. Pieprzyk, and W. Susilo, "A New Biocryptosystem-Oriented Security Analysis Framework and Implementation of Multibiometric Cryptosystems Based on Decision Level Fusion," *Information Forensics and Security, IEEE Transactions*, vol. 10, no. 6, pp. 1193-1206, 2015 Article (CrossRef Link).
[32] K. Xi, T. Ahmad, F. Han, and J. Hu, "A fingerprint based bio-cryptographic security protocol designed for client/server authentication in mobile computing environment" *Security and Communication Networks*, vol. 4, no.5, pp. 487-499, 2011. Article (CrossRef Link).
[33] AVISPA. Automated validation of internet security protocols and applications. http://www.avispa-project.org/. Accessed on January 2013.
[34] Dolev and A.C. Yao, "On the security of public key protocols" Information Theory. *IEEE Trans. Vol.* 29, no. 2, pp. 198–208, 1983. Article (CrossRef Link).

Shipra Kumari is currently working as a Research Scholar in the Department of Computer Science & Engineering at Indian school of Mines, Dhanbad, India. She received her Bachelor in Computer Applications and Master in Computer Applications from Indira Gandhi National Open University, New Delhi. Her research interest includes Cryptography and Network Security.

Hari Om is presently working as an Assistant Professor in the Department of Computer Science & Engineering at Indian School of Mines, Dhanbad, India. He did his Ph.D in Computer Science from Jawaharlal Nehru University, New Delhi, M.Tech. in Computer Science & Engineering from Kurukshetra University, Kurukshetra (Haryana), M.Sc. in Mathematics from Institute of Basic Sciences, Khandari, Dr. B. R. Ambedkar University, Agra. He has contributed more than hundred research papers in several International and National journals including IEEE Transactions and conference proceedings of high repute. He is a life member of Indian Society for Technical Education, Indian Mathematical Society, Indian Society of Mathematics and Mathematical Sciences, Cryptology Research Society of India and member of the Institute of Electronics and Telecommunication Engineers, IEEE, ACM, and IEICE. His main research interest includes Cryptography, Data Mining, Network Security, Image Processing.