Wireless Pers Commun (2015) 84:439–462
DOI 10.1007/s11277-015-2616-7

Design and Analysis of Bilinear Pairing Based Mutual Authentication and Key Agreement Protocol Usable in Multi-server Environment

Ruhul Amin[1] · G. P. Biswas[1]

Published online: 6 May 2015

© Springer Science+Business Media New York 2015

Abstract With the increasing popularity and demand for various applications, the internet user accesses remote server by performing remote user authentication protocol using smart card over the insecure channel. In order to resist insider attack, most of the users remember a set of identity and password for accessing different application servers. Therefore, remembering set of identity and password is an extra overhead to the user. To avoid the mentioned shortcoming, many remote user authentication and key agreement protocols for multi-server architecture have been proposed in the literature. Recently, Hsieh–Leu proposed an improve protocol of Liao et al. scheme and claimed that the improve protocol is applicable for practical implementation. However, through careful analysis, we found that Hsieh–Leu scheme is still vulnerable to user anonymity, password guessing attack, server masquerading attack and the password change phase is inefficient. Therefore, the main aim of this paper was to design a bilinear pairing based three factors remote user authentication scheme using smart card for providing security weaknesses free protocol. In order to validate security proof of the proposed protocol, this paper uses *BAN* logic which ensures that the same protocol achieves mutual authentication and session key agreement property securely. Furthermore, this paper also informally illustrates that the proposed protocol is well protected against all the relevant security attacks. The performance analysis and comparison with other schemes are also made, and it has been found that the proposed protocol achieves complete security requirements with comparatively lesser complexities.

Keywords Bilinear pairing · Biometric template · User authentication · Three factor · User anonymity · Security attacks

✉ Ruhul Amin
amin_ruhul@live.com; ruhulamin@cse.ism.ac.in

G. P. Biswas
gpbiswas@gmail.com

[1] Indian School of Mines, Dhanbad 826004, India

1 Introduction

As the number of internet users are increasing day by day exponentially and want to access various application servers for different purposes such as online shopping, online pay-TV, online bill payment, online banking transaction, file sharing, online game, distributed electronic medical records system, etc. In order to access the remote application server, mutual authentication with session key agreement is required in which a client authenticates to the server and vice versa over the insecure communication. To access the remote server, various password based remote user authentication schemes [1, 3, 5, 6, 10, 14, 28, 45] have been proposed in the literature for the single server environment, where the user only can access single server after execution registration phase. But, such type of schemes are not suitable for multi-server architecture, where a user cannot access several application servers on demand. It is noted that remembering set of different secret information(e.g: password, identity) to the U_i is the main complexity for accessing several single servers environment. On the other hand, it is not good practice for the user to use same password for various servers, because of insider attack. To avoid these difficulties, many multi-server architecture based remote user authentication and key agreement schemes [2, 4, 20, 27, 36, 43, 44, 46] have been introduced. But, it has been found that none of the protocols are completely free from security weaknesses [2].

Based on the basic cryptographic algorithms, the existing multi-server authentication schemes can be divided into two types, namely the hash based authentication scheme and the public-key based authentication scheme. It may be noted that the public-key cryptosystem has higher computational cost than hash-based authentication cryptosystem and it provides more security than hash-based authentication cryptosystem. In 2001, Boneh–Franklin [7] defined bilinear pairing (Weil pairing or Tate pairing) on elliptic curve cryptosystem. Bilinear pairing reduces the complexity of discrete logarithm problem (DLP) [16, 34] and also provides many advantages for the bilinear Diffie–Hellman problem (BDH) [13] that is used to design several public-key cryptosystems like encryption/decryption technique, short signature generation, and signcryption technique.

1.1 System Model

To achieve complete mutual authentication property between the entities involved, there are basically five authentication models shown in Fig. 1 for multi-server environment. Apart from this, there may exist several multi-server architectures, where mutual authentication occurs between the U_i and the application server (SS_j). As the messages are transmitted through insecure/open channel, it is a desirable property to achieve mutual authentication between all the entities involved in the system, where the computation cost should be minimum. The computation and communication cost is the directly proportional to the number of communications. Therefore, the mode of communication is an important issue for any multi-server environment. The proposed architecture has also presented in Fig. 1 and comparing with Hsieh–Leu scheme, it may take little more communication cost due to twice authentication of the U_i. As mention in [42], the cost of transmitting 1 Kb information is 3 J (joule) or equal to that of 80,000–600,000 instructions. Therefore, the mode of communication between the entities involved should be minimum and also should achieve mutual authentication property properly. In order to establish secure communication between the entities, the user always initiates the key agreement scheme for all the authentication models and finally shares a common secret session key.

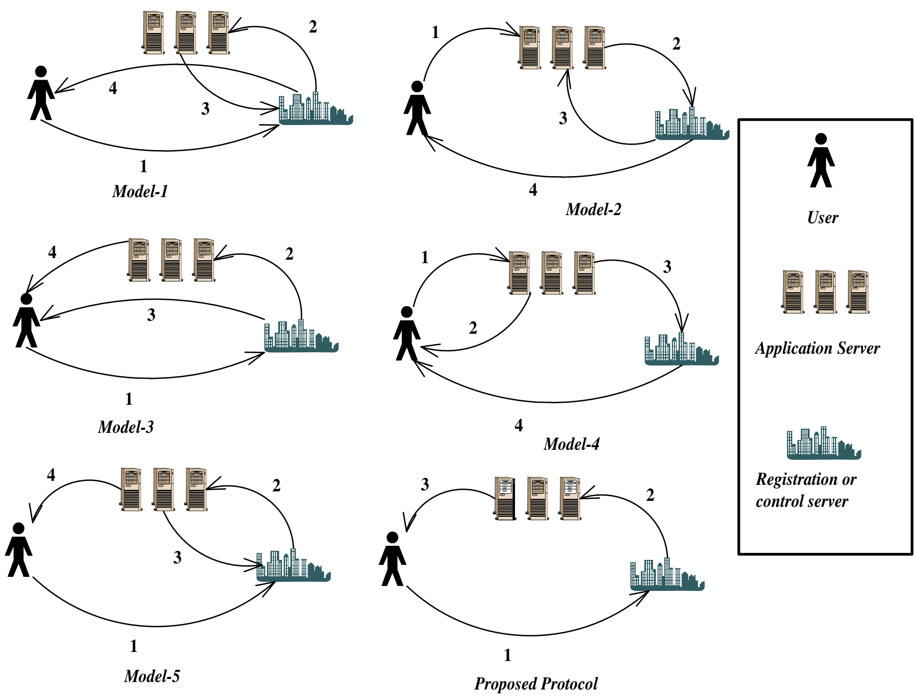

Fig. 1 Five basic authentication model for multi-server environment including proposed protocol

1.2 Related Works

A large number of user authentication and key agreement schemes based on either hash function or public key cryptosystem have been proposed usable in multi-server environment. In 2004, Juang's [22] proposed an efficient hash function and symmetric key cryptosystem based multi-server password authentication and key agreement scheme. In 2009, Hsiang and Shih [18] proposed a hash function based key agreement scheme for the multi-server architecture and then Sood et al.'s [38] demonstrated that the protocol proposed by Hsiang and Shih is vulnerable to replay attack, user impersonation attack, stolen smart card and the password change phase is not user-friendly. To resolve the security flaws of the Hsiang and Shih scheme, Sood et al. [38] proposed a dynamic identity based user authentication protocol usable in multi-server environment. After that, Li et al. [29] pointed out that the Sood et al.'s protocol is susceptible to leak-verifier attack, impersonation attack and stolen smart card attack and presented a another dynamic identity based user authentication and key agreement scheme for the same architecture. It may be noted that all the schemes [18, 29, 38] are dependent on the registration server *RS*. In 2009, Liao and Wang [31] proposed a dynamic *ID*-based remote user authentication scheme which is non-dependent on *RS* and then many researchers propose an authentication system [12, 25, 26, 37] for multi-server environment where the protocol is not dependent on *RS* to achieve full security requirements.

To design authentication system for multi-server architecture another useful technique public key cryptography is used in the literature. Based on the difficulty of the factorization problem of public key cryptography, in 2000, Lee and Chang [24] proposed a user identification and key distribution scheme. Then in 2001, Tsaur et al. [41] proposed an authentication protocol for multi-server architecture based on *RSA* cryptosystem and lagrange interpolating polynomial. Thereafter, Lin et al. [32] proposed a protocol based on the simple geometric properties of the Euclidean and discrete logarithm problem concept. Geng and Zhang [17] proposed a dynamic identity based user authentication and key agreement protocol for multi-server environment using bilinear-pairing. Thereafter, Tseng et al.'s [40] proposed a smart card based authentication system but Liao and Hsiao [30] pointed out that Tseng et al.'s [40] protocol is vulnerable to insider attack, off-line dictionary attack, malicious-server attack and cannot provide mutual authentication property and proposed an improved protocol which has also claimed for its practical implementation. After that, Hsieh–Leu [19] scheme pointed out that the Liao and Hsiao's [30] protocol has several security weaknesses such as trace attack, a burden to update *ID* table and lack of pre-authentication in the login phase and proposed an improvement protocol to resolve the mentioned weaknesses. In this paper, we have described some security weaknesses of the Hsieh–Leu scheme [19] such as, user anonymity problem, off-line password guessing attack, server masquerading attack and the password change phase, is not user-friendly. After reviewing the related related works, it is confirmed that none of the authentication protocols achieve complete security requirements [2]. Therefore, we have presented anonymity preserving user authentication and key agreement protocol usable in multi-server environment using bilinear-pairing.

1.3 Motivation and Contributions

Today's most of the users are dependent on one or more single server for accessing different types of benefits from the remote server. So, it is very necessary to design a user authentication and key agreement protocol which is applicable for practical implementation. In Sect. 1.2, we have observed that none of the protocol achieves complete security requirements [2] and inefficient in terms of complexities. Moreover, we have shown that the protocol proposed by Hsieh–Leu scheme has some security weaknesses such as user anonymity problem, off-line password guessing attack, server masquerading attack and the password change phase is not user-friendly. In order to resolve the above mentioned weaknesses and for achieving complete security requirements, it is very necessary for designing an authentication key agreement protocol and this paper presents an authentication and key agreement protocol for multi-server environment using bilinear pairing. After rigorous security analysis and performance comparison, it is confirm that the proposed protocol achieves complete security requirements and relatively better complexities than existing related works. Additionally, the proposed protocol contributes efficient login phase, user anonymity, user-friendly password change phase and mutual authentication property between the U_i and SS_j.

1.4 Threat Model

In this paper, we assume the following assumptions based on the threat model mentioned in [8, 15, 23, 35]. The following assumptions are:

1. An attacker ($\mathcal{A}$) is able to extract the smart card information by monitoring the power consumption. For example if an attacker gets the smart card of the valid user, he/she then may get all the stored information of the smart card.

2. An attacker may eavesdrops all the communication between the entities involved of the protocol over the public channel. It is also assume that an attacker cannot intercept the message over the secure channel.
3. An attacker can guess low entropy password and identity individually easily but guessing two secret parameters (e.g. password, identity) is computationally infeasible in polynomial time.
4. An attacker can modify, delete and resend, reroute the eavesdrops message.
5. An attacker may be a legitimate user or vice versa.
6. It can be assumed that the protocol used in the authentication system is known to the attacker.
7. If we assume that the length of the user's identity and password is n character, then the probability of guessing approximately composed of n character is $\frac{1}{2^{6n}}$ as pointed out by [11].

1.5 Organization of the Paper

Rest of the paper is sketched as follows: In Sect. 2, we discussed the concept and property of cryptographic one-way hash function, bio-hashing and several definitions along with computational problems related to bilinear pairing as preliminaries of our work. In Sect. 3, we briefly review published Hsieh–Leu scheme and the security weaknesses of the same protocol appears in Sect. 4. The Sect. 5 addresses the proposed protocol and the security validation using *BAN* logic demonstrates in Sect. 6. Moreover, the Sect. 7 shows informal cryptanalysis of the proposed protocol. The performance comparison are also presented in Sect. 8. Finally, we conclude the paper in Sect. 9.

2 Preliminaries

In this section, we briefly review the basic concepts of cryptographic one-way hash function, bio-hashing and bilinear pairing.

2.1 Cryptographic One-Way Hash Function

A cryptographic one-way hash function maps a string of arbitrary length to a string of fixed length called the hashed value. It can be symbolized as: $h : X \to Y$, where $X = \{0,1\}^*$, and $Y = \{0,1\}^n$. X is binary string of arbitrary length and Y is a binary string of fixed length n. It is used in many cryptographic applications such as digital signature, random sequence generators in key agreement, authentication protocols and so on. Cryptographic one-way hash function satisfies the following properties:

1. *Easiness* Given $m \in X$, it can be easily compute y such that $y = h(m)$.
2. *Preimage resistant* It is hard to find m from given y, where $h(m) = y$.
3. *Second-preimage resistant* It is hard to find input $m' \in X$ such that $h(m) = h(m')$ for given input $m \in X$ and $m' \neq m$.
4. *Collision resistant* It is hard to find a pair $(m, m') \in X \times X$ such that $h(m) = h(m')$, where $m \neq m'$.

5. *Mixing-transformation* On any input $m \in X$, the hashed value $y = h(m)$ is computationally indistinguishable from a uniform binary string in the interval $\{0, 2^n\}$, where n is the output length of hash $h(\cdot)$.

2.2 Bio-hashing

In [21], Jina et al. proposed a two factor authenticator based on iterated inner products between tokenised pseudorandom number and the user specific fingerprint feature, which produces a set of user specific compact code that coined as *Bio-hashing*. Later, Lumini and Nanni in [33] proposed an improvement on *Bio-hashing*. As pointed out in [11], *Bio-hashing* is used to map a user/patient's biometric features onto user-specific random vectors in order to generate a code, called the *Bio-code* and then discritizes the projection coefficients into zero or one. *Bio-Hashing* is one-way. *Bio-code* is also as secure as a hashed password.

2.3 Bilinear Pairing

The bilinear pairings namely the Weil pairings or Tate pairings are used in important applications of cryptography and allowed us to construct identity(ID)-based cryptographic schemes. Suppose $< G_1, + >$ be an additive cyclic group and $< G_2, \times >$ a multiplicative cyclic group of prime order q. P is a generator of group G1. A mapping $\hat{e} : G_1 \times G_1 \longrightarrow G_2$ is called a bilinear mapping if it satisfies the following properties:

1. *Bilinear property* For all $S, Q, R \in G_1, \hat{e}(S + Q, R) = \hat{e}(S, R) \times \hat{e}(Q, R)$ and $\hat{e}(S, Q + R) = \hat{e}(S, Q) \times \hat{e}(S, R)$
2. *Non-degeneracy property* There exist $S, Q \in G_1$, such that $\hat{e}(S, Q) \neq 1_{G_2}$, where 1_{G_2} is the identity element of the multiplicative cyclic group G_2
3. *Computability property* There exists an efficient algorithm to compute $\hat{e}(S, Q) \; \forall \; S, Q \in G_1$

3 Brief Review of Hsieh–Leu Scheme

In this section, we present all the phases of the Hsieh–Leu Scheme [19] such as setup, server registration, user registration, login, verification and password change phase for better understanding of the same protocol.

3.1 Setup Phase

The registration server RS chooses a random number $S_{RS} \in Z_q^*$ keeps as the system's private key and computes $Pub_{RS} = S_{RS} \cdot P$ as the public key, where P is the generator of the group G_1 and publishes system parameters $\langle \hat{e}, G_1, G_2, q, P, Pub_{RS}, H(), h() \rangle$ as public.

3.2 Server Registration Phase

In this phase, all the service provider servers SS_j choose desired identity SID_j and then generates a random nonce $v_j \in Z_q^*$ and submits $\langle V_j, SID_j \rangle$ to the RS after computing $V_j = v_j \cdot P$. The RS then generates a random value $w_j \in Z_q^*$ and computes

$W_j = V_j + w_j \cdot P, S_j = (h(SID_j \parallel W_j) \cdot S_{RS} + w_j) \, mod \, q$. Further, the RS issues registration token $\langle T, S_j, W_j \rangle$ for each SS_j. After getting the registration token, the SS_j computes the private key as $s_j = S_j + v_j \, mod \, q$ and checks the validity whether the equation $Pub_j = s_j \cdot P = h(SID_j \parallel W) \cdot Pub_{RS} + W_j$. Finally the SS_j has its own private key s_j, public key Pub_j and the registration token $H(T)$.

3.3 User Registration Phase

In order to join as a legal user to the RS, the user and RS perform the following steps:

Step 1 The U_i chooses a password PW_i, identity ID_i, random number $b \in Z_q^*$ and computes $HPW_i = h(PW_i \parallel b) \cdot P$. Then, the user U_i sends $\langle H(ID_i), HPW_i \rangle$ to the registration server RS. It may be noted that Hsieh–Leu does not define the communication channel (secure/insecure) during execution of the registration phase. However, we have assumed that the channel is secure.

Step 2 After receiving the registration message, the RS checks the existence of $H(ID_i)$ in its database or not. If it is not registered, the RS computes $DID_i = S_{RS} \cdot H(ID_i), Reg_{ID_i} = DID_i \oplus S_{RS} \cdot HPW_i$ and $Auth_i = T \cdot H(ID_i)$. Finally, the RS issues a smart card after storing $\langle Auth_i, Reg_{ID_i}, Pub_{RS}, H(\cdot), h(\cdot) \rangle$ into the memory of smart card through secure channel. At the same time, RS maintains $H(ID_i)$ into the database.

Step 3 After receiving the smart card, the U_i stores $\langle b, CID_i \rangle$ into the smart card after computing $CID_i = h(ID_i) \oplus h(PW_i \parallel b)$.

3.4 Login Phase

In this phase, the user or card reader performs the following steps:

Step 1 Initially, the U_i inserts his/her smart card into the card reader and enters ID_i, PW_i. Then, the smart card computes $CID_i^* = h(ID_i) \oplus h(PW_i \parallel b)$ and matches CID_i^* with the stored CID_i. If it does not match, the card reader terminates the session; otherwise, goto the next step.

Step 2 The smart card generates two numbers $x, r_i \in Z_q^*$ and computes $R_i = r_i \cdot P, DID_i = Reg_{ID_i} \oplus h(PW_i \parallel b) \cdot Pub_{RS}, QID_i = H(ID_i), C_m = x \cdot H(ID_i), xAuth_i = x \cdot Auth_i$, $M_i = xr_i \cdot QID_i, d_{ij} = h(xH(ID_i) \parallel SID_j \parallel M_i \parallel R_i), B_{ij} = x(r_i + d_{ij}) \cdot DID_i$. Finally, the smart card sends login message $\langle xAuth_i, C_m, M_i, R_i, B_{ij} \rangle$ to the SS_j through public channel.

3.5 Verification Phase

Step 1 After receiving the login message, the SS_j first computes $Auth_i^* = x \cdot C_m$ and matches it with the received $xAuth_i$. If it does not match, the SS_j terminates the session; otherwise, computes $d_{ij} = h(C_m \parallel SID_j \parallel M_i \parallel R_i)$ and checks the condition $\hat{e} < B_{i,j}, P > = \hat{e} < M_i + xd_{ij} \cdot QID_i, Pub_{RS} >$. If the condition holds, SS_j accepts the request; otherwise, stops the connection.

Step 2 The SS_j then generates a random point $R_j = r_j \cdot P$ and computes temporary key $TK_{ij} = r_j \cdot R_i, K_{ij} = s_j \cdot R_i, Auth_{ij} = h(C_m \parallel K_{ij} \parallel R_j)$ and sends reply message $\langle Auth_{ij}, K_{ij}, R_j \rangle$ to the user U_i through public channel.

Step 3 After receiving the reply message, the U_i computes $Pub_j = h(SID_j \parallel W_j) Pub_{RS} + W_j$ based on self certified public key of SS_j and further computes $TK_{ij} = r_i \cdot R_j, K_{ij} = r_i \cdot Pub_j$. Then, the U_i checks whether the computed $h(xH(ID_i) \parallel K_{ij} \parallel$

$R_j)$ matches with the received $Auth_{ij}$. If it matches, the U_i computes $Auth_{ij} = h(xH(ID_i) \parallel K_{ij} \parallel R_i \parallel R_j)$ and sends it to the SS_j through open channel.

Step 4 After receiving $Auth_{ij}$, the SS_j checks whether $h(xH(ID_i) \parallel K_{ij} \parallel R_i \parallel R_j) = Auth_{ij}$ holds or not. If the condition is equal, both U_i and SS_j shares a session key $SK = h(xH(ID_i) \parallel TK_{ij})$.

3.6 Password Change Phase

The execution of this phase is performed by the U_i and RS to update the password for the existing user's.

Step 1 The U_i first inserts the smart card and enters ID_i, PW_i to the CR. Then, the SC computes $CID_i^* = h(ID_i) \oplus h(PW_i \parallel b)$ and matches it with the stored CID_i. If it does not match, the CR terminates the session; otherwise, computes $N_i = n_i \cdot P$ after generating a random number n_i. Then, the U_i computes $DID_i = Reg_{ID_i} \oplus h(PW_i \parallel b) \cdot Pub_{RS}, NID_i = DID_i \oplus n_i \cdot Pub_{RS}$ and sends password change request $\langle H(ID_i), NID_i, N_i \rangle$ to the RS.

Step 2 After receiving the request, the RS checks the database to verify the user identity. If it is confirmed, the RS computes $QID_i = H(ID_i), DID_i = NID_i \oplus S_{RS} \cdot N_i$ and whether $\hat{e} < DID_i, P > \; = \hat{e} < QID_i, Pub_{RS} >$ holds or not. If it holds, the RS sends $\langle V_1 \rangle$ to the U_i after computing $V_1 = h(DID_i \parallel S_{RS} \cdot N_i)$.

Step 3 After receiving V_1, the U_i compares V_1 with the computed $h(DID_i \parallel n_i \cdot Pub_{RS})$ to confirm the legality of the RS. Further, the U_i computes $HPW_{new} = h(PW_{new} \parallel b_i), V_2 = HPW_{new} \oplus n_i \cdot Pub_{RS}, V_3 = h(DID_i \parallel n_i \cdot Pub_{RS} \parallel HPW_{new})$ and sends $\langle V_2, V_3 \rangle$ to the RS.

Step 4 After getting $\langle V_2, V_3 \rangle$, the RS computes $V_2 \oplus S_{RS} \cdot N_i$ to extract HPW_{new} and compares the received V_3 with the computed $h(DID_i \parallel S_{RS} \cdot N_i \parallel HPW_{new})$. If the comparison holds, the legality of the user is confirmed. Finally, the RS computes $Reg_{ID_i}^{new} = DID_i \oplus S_{RS} \cdot HPW_{new}, V_4 = Reg_{ID_i}^{new} \oplus S_{RS} \cdot N_i$ and sends $\langle V_4 \rangle$ to the U_i.

Step 5 After getting V_4, the U_i computes $V_4 \oplus n_i \cdot Pub_{RS}$ to extract $Reg_{ID_i}^{new}$ and $CID_{new} = h(ID_i) \oplus h(PW_{new} \parallel b_i)$. Finally, the smart card of the user U_i replaces Reg_{ID_i} and CID_i with $Reg_{ID_i}^{new}$ and CID_{new} respectively.

4 Security Pitfalls and Discussion of Hsieh–Leu Scheme

In this section, we have described that the recently published Hsieh–Leu scheme [19] has several security pitfalls such as off-line identity guessing attack (user anonymity), off-line password guessing attack and server masquerading attack. Additionally, we have also demonstrated that the password change phase of the same scheme is not user-friendly. Based on the threat model assumptions, the mentioned security weaknesses are presented below:

4.1 Off-line Identity Guessing Attack

Generally, user always chooses easy to remember identity for his/her benefit and according to the [2], it is guessable by an attacker. Hsieh–Leu claimed that their protocol achieves user anonymity property that means, an attacker cannot trace the user's original identity. However, we have identified that their protocol suffers from off-line identity guessing attack which is presented below:

Step 1 An attacker first extracts $Auth_i$ parameter from the smart card and intercepts C_m parameter from the login message.

Step 2 The attacker computes $\hat{e}\langle Auth_i, C_m\rangle$ which is equivalent to $\hat{e}{<}H(ID_i), xAuth_i) >$. It may be noted that the attacker knows all the parameters of bilinear pairing operation except the user identity ID_i. Now, the attacker guessess an identity ID_i^g and executes step 3.

Correctness of $\hat{e}{<}Auth_i, C_m > \; = \hat{e}{<}H(ID_i), xAuth_i) >$

$$\begin{aligned} \hat{e}{<}Auth_i, C_m > &= \hat{e}{<}T \cdot H(ID_i), C_m > \text{ since } Auth_i = T \cdot H(ID_i). \\ &= \hat{e}{<}H(ID_i), C_m >^{T} \\ &= \hat{e}{<}H(ID_i), T \cdot C_m > \text{ from the bilinear property.} \\ &= \hat{e}{<}H(ID_i), xAuth_i > \text{ since } xAuth_i = T \cdot C_m \\ &= \hat{e}{<}H(ID_i), xAuth_i > \textbf{ proved} \end{aligned}$$

Step 3 The $\mathcal{A}$ computes $\hat{e}{<}H(ID_i^g), xAuth_i) >$ using the new chosen identity ID_i^g and checks the correctness whether the computed $\hat{e}{<}H(ID_i^g), xAuth_i) >$ is equal to $\hat{e}{<}Auth_i, C_m >$. If the inequality holds, the attacker trace the original identity; otherwise, chooses another identity and executes step 3 until the correct identity is obtained. In this way, the attacker can obtain the original identity of the user.

4.2 Off-line Password Guessing Attack

It can be assumed that most of the user's always use easy to remember password which is guessable for accessing the remote server. Based on the threat model, an attacker knows the algorithm of the protocol proposed by Hsieh–Leu. As a result, the attacker knows the condition $\hat{e}{<}DID_i, P > \; = \hat{e}{<}QID_i, Pub_{RS} >$ used in the password change phase and can guess the user's low entropy password by executing the following steps:

Step 1 As the attacker knows the user's original identity ID_i and public parameter Pub_{RS}, he/she can compute $R = \hat{e}{<}QID_i, Pub_{RS} >$. The description of the condition $\hat{e}{<}DID_i, P > \; = \hat{e}{<}QID_i, Pub_{RS} >$ are given below:

$$\begin{aligned} \hat{e}{<}DID_i, P > &= \hat{e}{<}QID_i, Pub_{RS} > \\ &= \hat{e}{<}DID_i, P > = R, \text{ where } R = \hat{e}{<}QID_i, Pub_{RS} > \\ &= \hat{e}{<}Reg_{ID_i} \oplus S_{RS} \cdot HPW_i, P > = R, \text{ since } DID_i = S_{RS} \cdot HPW_i \\ &= \hat{e}{<}Reg_{ID_i} \oplus S_{RS} \cdot h(PW_i \parallel b) \cdot P, P > = R, \\ \text{since } HPW_i &= h(PW_i \parallel b) \cdot P \\ &= \hat{e}{<}Reg_{ID_i} \oplus h(PW_i \parallel b) \cdot S_{RS} \cdot P, P > = R \\ &= \hat{e}{<}Reg_{ID_i} \oplus h(PW_i \parallel b) \cdot Pub_{RS}, P > = R, \text{ since } Pub_{RS} = S_{RS} \cdot P \\ &= \hat{e}{<}Reg_{ID_i} \oplus h(PW_i \parallel b) \cdot Pub_{RS}, P > = R \end{aligned} \tag{1}$$

Step 2 From the Eq. (1), it is noticeable that the attacker knows all the parameters except the user's password based on the threat model assumption. Now, the attacker chooses either a password PW_i^d from the dictionary or guess a password and checks the correctness whether $\hat{e}{<}Reg_{ID_i} \oplus h(PW_i^d \parallel b) \cdot Pub_{RS}, P > \; = R$.

Step 3 If the above correctness is correct, the attacker's guess password is correct; otherwise, continues step 2, until the correct password is obtained. In this way, the attacker can guess the user's correct password from the protocol description of Hsieh–Leu.

4.3 Server Masquerading Attack

In order to launch server masquerade attack, an attacker first intercepts the *j-th* communicating message $\langle xAuth_i, C_m, M_i, R_i, B_{ij}, Auth_{ij}, K_{ij}, R_j \rangle$ from the protocol. After intercepting these parameter, he/she can launch the server masquerading attack successfully which is presented below:

Step 1 Initially, an attacker generates a random point $R_a = r_a \cdot P$, temporary key $TK_{ia} = r_a \cdot R_i$, $Auth_a = h(C_m \parallel K_{ji} \parallel R_a)$ and sends $\langle Auth_a, K_{ji}, R_a \rangle$ to the user U_i, where the parameter K_{ji} remain unchanged.

Step 2 After receiving the message $\langle Auth_a, K_{ji}, R_a \rangle$, the U_i computes Pub_j based on self-certified public key and then computes temporary key $TK_{ai} = r_i \cdot R_a$, shared secret key $K_i = r_i \cdot Pub_j$ and $Auth^* = h(xH(ID_i) \parallel K_i \parallel R_a)$. It may be noted that the parameters TK_{ia} and TK_{ai} are equal. Then, the U_i checks the condition whether the computed $Auth^*$ matches with the received $Auth_a$. If the condition matches, the attacker launches successfully server masquerading attack.

Step 3 After launching the server masquerading attack, the $\mathcal{A}$ knows all the session key parameters such as $\langle C_m, TK_{ia} \rangle$ and finally computes the session key as, $SK = h(C_m \parallel TK_{ia})$. Therefore, the attacker can read all the confidential information after decrypting using session key.

4.4 Inappropriate Password Change Phase

It is a good practice that if a user U_i change his/her password securely without help of the registration server and this approach reduces the computation and communication cost of the protocol along with network congestion. After reviews carefully Hsieh–Leu's scheme, it is our claim that the password change phase of the same scheme is not efficient and user-friendly because of the following reasons:

1. In step-4 of the password change phase of the Hsieh–Leu scheme, the U_i computes $HPW_{new} = h(PW_{new} \parallel b_i) \in Z_q^*$ and executes xor operation $(\oplus)$ with the $n_i \cdot Pub_{RS} \in G_1$, which is mathematically infeasible. Therefore, the computation $HPW_{new} \oplus n_i \cdot Pub_{RS}$ is not feasible. As a result, the password change phase is not efficient.
2. The password change phase is fully dependent on the RS that means, without help of the RS, U_i is not able to change password.
3. The efficiency of any authentication protocol is inversely proportional to the number of secure channel and it is noted that the number of secure channel should be minimum. However, the Hsieh–Leu scheme uses secure channel in the password change phase to update the user's password, which is not desirable.
4. In order to design a better authentication protocol, the computation and communication cost should be as minimum as possible. To change the password successfully, Hsieh–Leu's protocol communicates one or more times to the entities involved in the protocol. Therefore, the computation and communication cost reduces the protocol's efficiency.

5 Proposed Protocol

This section proposes a user authentication and key agreement scheme usable in multi-server environment using bilinear pairing. Our proposed protocol has three entities user U_i, server SS_j and registration server RS, where RS executes the registration phase for both the

Table 1 List of notations used

Symbol	Description
$\mathcal{A}$	Attacker/adversary
SC	Smart card
U_i	*i-th* user
SS_j	*j-th* application server
RS	Registration or control server
PW_i	Password of the user U_i
ID_i	Identity of the user U_i
B_i	Biometric of the user U_i
$\hat{e} : G_1 \times G_1 \rightarrow G_2$	Bilinear Mapping, where G_1, G_2 are additive and multiplicative cyclic group
P	Generator point of G_1 with the order q;
aP	A times addition of point P
S_{RS}	Secret key of the (RS)
Pub_{RS}	Public key $Pub_{RS} = S_{RS} \cdot P$ of the (RS)
$H(\cdot)$	Map to Point hash function $H(\cdot) : (0,1)^* \rightarrow G_1$.
$h(\cdot)$	Cryptographic one-way hash function.
$H_1(\cdot)$	Bio-hashing technique.
$\|$	Concatenation operation ($a \| b$)
$\oplus$	xor operation ($a \oplus b$)
$+$	Addition of two points ($a + b$)

user and the server, whereas SS_j provides services to the U_i on demand. Like Hsieh–Leu scheme, the proposed scheme has the following six phases: setup phase, server registration phase, user registration phase, login phase, verification and key agreement phase and password change phase as described below, where the notations used throughout this paper are listed in Table 1.

5.1 Setup Phase

Setup phase of the proposed scheme is same as Hsieh–Leu scheme, which is described earlier.

5.2 Server Registration Phase

When a new application server SS_j wants to participate in the multi-server system, initially SS_j chooses an identity SID_j, where $(1 < j < m)$ and sends it to the RS through open channel. It may be noted that m represents the number of SS_j are involved in the multi-server system. After receiving it, the RS computes $S_j = h(SID_j \| S_{RS}) \cdot P \in G_1, Pub_j = S_j \cdot P \in G_1$ and sends secret key S_j to the SS_j through secure channel and declares Pub_j as the public parameter of the application server.

5.3 User Registration Phase

Similar to the Hsieh–Leu scheme, user registration phase is needed before accessing the services of the SS_j. We incorporated biometric template B_i such as fingerprint, in our protocol to achieve top security and applied *Bio-hashing* technique for the B_i described in the preliminary section. The procedures of the user registration phase are presented below:

Step 1 The U_i selects his/her desired low entropy identity ID_i, password PW_i and computes $UID_i = H(ID_i) \in G_1, HPW_i = h(PW_i \parallel ID_i) \cdot P \in G_1$ and then sends registration message $\langle UID_i, HPW_i, B_i \rangle$ to the RS through secure channel.

Step 2 After receiving the registration message, the RS computes the following operations:

$$\begin{aligned} b_i &= H_1(B_i) \\ Reg_i &= h(UID_i \parallel HPW_i) \in Z_q^* \\ DID_i &= S_{RS} \cdot UID_i \in G_1 \\ Z_i &= h(HPW) \cdot P \in G_1 \\ S_i &= DID_i + Z_i \in G_1 \end{aligned}$$

Then, then RS generates one-time an unique identity TID_i for each user U_i and computes $PID_i = h(TID_i \parallel S_{RS})$. After that, the RS stores $\langle PID_i, UID_i \rangle$ and status bit(0,1) shown in the following, where the status bit indicates whether the U_i logged-into the system or not. The status bit one(1) indicates that the U_i is accessing the system; otherwise, zero (0).

PID_i	UID_i	Status bit
PID_1	UID_1	0
PID_2	UID_2	1
PID_3	UID_3	1
PID_n	UID_n	0

Step 3 Finally, the RS stores $\langle TID_i, b_i, Reg_i, S_i, Pub_{RS}, H(.), h(.), H_1(.) \rangle$ into memory of smart card and delivers it through secure channel. After getting it, the U_i keeps it securely for future use. It may be noted that our proposed protocol cannot use any random number to resist insider attack.

5.4 Login Phase

In the login phase of the proposed protocol, the U_i first inserts his/her smart card to the card reader and keys user's confidential information(s). After that, the card reader verifies the user legitimacy and generates the login message and forwards it to the server. The following steps describe the login procedure of the proposed protocol:

Step 1 The U_i inserts the smart card *(SC)* into terminal and keys ID_i^*, PW_i^* and biometric template on the specific devices.

Step 2 The SC then computes $UID_i^* = H(ID_i^*), HPW_i^* = h(PW_i^* \parallel ID_i^*) \cdot P, b_i^* = H_1(B_i)$ and checks the condition whether the computed b_i^* matches with the stored b_i. If it matches, user biometric successfully accepted; otherwise, aborts the connection. After that, the SC computes $Reg_i^* = h(UID_i^* \parallel HPW_i^*)$ and compares with the stored Reg_i. If the comparison holds, it implies that the U_i entered correct $\langle ID_i, PW_i \rangle$ and proceeds to the next steps.

Step 3 The SC generates a random number r_i and performs the following operations after choosing application server's identity SID_j on user's demand.

$$\begin{aligned} Z_i &= h(HPW_i) \cdot P \in G_1 \\ DID_i &= S_i - Z_i \in G_1 \\ R_i &= r_i \cdot P \in G_1 \quad M_i = r_i \cdot Pub_{RS} \in G_1 \\ d_i &= h(UID_i \parallel SID_j \parallel M_i \parallel DID_i) \in Z_q^* \end{aligned}$$

Step 4 Finally, the SC sends the login request message $M_1 = \langle R_i, TID_i, SID_j, d_i \rangle$ to the RS through public channel.

5.5 Verification and Key Agreement Phase

This phase achieves mutual authentication and session key agreement between the U_i and the SS_j after execution all the steps which are presented below:

Step 1 After receiving the login message M_1, the RS computes $PID_i^* = h(TID_i \parallel S_{RS})$ and checks the existence in the server database. If it does not exist, aborts the connection; otherwise, retrieves UID_i from the database.

Step 2 The RS computes $M_i^* = R_i \cdot S_{RS} = M_i, DID_i^* = UID_i \cdot S_{RS}, d_i^* = h(UID_i \parallel SID_j \parallel M_i^* \parallel DID_i^*)$ and then compares the computed d_i^* with the received d_i. Further, the RS checks the condition whether $\hat{e} < UID_i, Pub_{RS} > \; = \hat{e} < DID_i, P >$. If both the condition holds, goto the next steps; otherwise, terminates the connection.

Step 3 The RS generates a random nonce n_i and computes the following operations:

$$\begin{aligned} S_j &= h(SID_j \parallel S_{RS}) \cdot P \in G_1 \\ L_i &= n_i \cdot P \quad w_i = n_i \cdot Pub_j \\ K_i &= UID_i + w_i \\ T_i &= h(UID_i \parallel SID_j \parallel S_j \parallel w_i) \end{aligned}$$

where S_j and Pub_j is the SS_j's secret key and public key respectively.

Step 4 The RS finally sends message $M_2 = \langle L_i, K_i, SID_j, T_i \rangle$ to the SS_j through public channel.

Step 5 After receiving the messages M_2, the SS_j computes $w_i^* = L_i \cdot S_j = w_i, UID_i = K_i - w_i^* \in G_1, T_i^* = h(UID_i \parallel SID_j \parallel S_j \parallel w_i^*) \in Z_q^*$ and checks whether $T_i^* = T_i$ or not. If $T_i^* \neq T_i$, terminates the connection; otherwise, SS_j believes that the RS and U_i are legitimacy entities.

Step 6 The SS_j generates a random nonce c_i and computes $y_i = c_i \cdot P \in G_1, D_i = y_i + UID_i \in G_1, G_i = h(y_i \parallel UID_i) \in Z_q^*, E_i = h(UID_i \parallel SID_j \parallel y_i \parallel G_i) \in Z_q^*$ and sends $M_3 = \langle SID_j, K_i, D_i, E_i \rangle$ to the U_i through public channel.

Step 7 After receiving M_3, the U_i computes $y_i^* = D_i - UID_i, w_i^* = K_i - UID_i, G_i^* = h(y_i^* \parallel UID_i)$ $E_i^* = h(UID_i \parallel SID_j \parallel y_i^* \parallel G_i^*)$ and checks the correctness whether $E_i^* = E_i$. If the correctness holds, it implies that SS_j and RS is authentic entities. The U_i further computes $SK = h(UID_i \parallel SID_j \parallel M_i \parallel y_i^* \parallel w_i^*), V_i = h(''0'' \parallel SK), O_i = M_i + y_i^*$ and sends $M_4 = \langle V_i, O_i \rangle$ to the SS_j through public channel.

Step 8 After receiving M_4, the SS_j computes $M_i^* = O_i - y_i, SK^* = h(UID_i \parallel SID_j \parallel M_i^* \parallel y_i \parallel w_i^*), V_i^* = h(''0'' \parallel SK^*)$ and compares the verification of the V_i^* with the received V_i. If the verification holds, the SS_j and the U_i can securely communicate using the shared secret session key SK; otherwise, terminates the connection.

5.6 Password Update Phase

It may happen that the user's password may disclose or leak to the third party by some means. At that moment, it is very necessary to change the password immediately. Therefore, the password change phase should be included with the authentication protocol. Our proposed protocol provides user-friendly password change phase, where the execution does not depend on the registration server. All the steps of this phase are presented below:

Step 1 The U_i inserts the smart card *(SC)* into terminal and keys $\langle ID_i, PW_i \rangle$ and biometric template on the specific devices.

Step 2 The *(SC)* then computes $UID_i = H(ID_i) \in G_1, HPW_i = h(PW_i \parallel ID_i) \cdot P \in G_1, b_i^* = H_1(B_i)$ and checks the condition whether the computed b_i^* matches with the stored b_i. If it matches, the SC believes that the biometric template is from the same user; otherwise, aborts the connection. After that, the *(SC)* computes $Reg_i^* = h(UID_i \parallel HPW_i)$ and compares with the stored Reg_i. If the comparison holds, it implies that the U_i entered correct $\langle ID_i, PW_i \rangle$ and asks to input to U_i for a new password PW_i^{new}.

Step 3 The SC computes $DID_i = S_i - Z_i = S_{RS} \cdot UID_i$ using user's old password and further computes $HPW^{new} = h(PW_i^{new} \parallel ID_i) \cdot P, Reg_i^{new} = h(UID_i \parallel HPW^{new}), Z_i^{new} = h(HPW^{new}) \cdot P, S_i^{new} = DID_i + Z_i^{new}$ and replaces $\langle Reg_i, S_i \rangle$ with $\langle Reg_i^{new}, S_i^{new} \rangle$ into memory of smart card and keeps rest of the parameters unchanged.

6 Authentication Proof Based on BAN Logic

This section addresses the security analysis of our proposed protocol using Burrows–Abadi–Needham logic [3, 9, 39], generally called as *BAN* logic. The *BAN* logic is well-known formal model used to analyze the security of authentication and key distribution protocols in the literature. Some preliminaries and notations of the *BAN* logic are described as follows:

Principals are those agents which are involved in the protocol (usually people or programs).
Keys are used to encrypt messages symmetrically.
Public Keys are similar to Keys except that they are used in pairs.
Nonces are message parts that are not meant to be repeated.
Timestamps are similar to nonces in that they are unlikely to be repeated.

Some *BAN* statements which are helpful for analyzing security of the proposed protocol are given below:

$P \mid\equiv X$: P believes X, or P would be entitled to believe X. In particular, P can take X as true.
$P \triangleleft X$: P sees X. P has received some message X and is capable of reading and repeating it (seeing rule).
$P \mid\sim X$: P once said X. P at some time sent a message including the statement X. It is not known whether this is a replay, though it is known that P believed X when he sent it.
$P \Rightarrow X$: P has jurisdiction over X. The principal P is an authority on X and should be trusted on this matter.

$\sharp(X)$: The message X is fresh.
(X, Y) : The formulae X or Y is one part of the formulae (X,Y).
$<X>_Y$: The formulae X combined with the formulae Y.
$\{X\}_K$: The formulae X is encrypted under the key K.
$(X)_K$: The formulae X is hashed with the key K.
$P \overset{K}{\longleftrightarrow} Q$: Principals P and Q communicate via shared key K.
$P \overset{X}{\rightleftharpoons} Q$: The formula X is a secret known only to P and Q, and possibly to principals trusted by them.
$\overset{K}{\mapsto} P$: Principal P has K as its public key.
SK: The session key used in the current session.

Some main logical postulates of the *BAN* logic are as follows:

- The message-meaning rule: $\frac{P|\equiv P \overset{K}{\rightleftharpoons} Q,\, P \triangleleft <X>_K}{P|\equiv Q|\sim X}$
  If the principal P believes that the secret K is shared with Q and sees $\langle X \rangle_K$, then P believes that Q once said X.
- The freshness-conjuncatenation rule: $\frac{P|\equiv\sharp(X)}{P|\equiv\sharp(X,Y)}$
  If the principal believes that X is fresh, then the principal P believes freshness of (X,Y).
- The belief rule: $\frac{P|\equiv(X),P|\equiv Y}{P|\equiv(X,Y)}$
  If the principal P believes X and Y, then the principal P believes (X,Y).
- The nonce-verification rule: $\frac{P|\equiv\sharp(X,\, P|\equiv Q|\sim X}{P|\equiv Q|\equiv X}$
  If the principal P believes that X is fresh and the principal Q once sent X, then principal P believes that Q believes X.
- The jurisdiction rule: $\frac{P|\equiv Q \Rightarrow X,\, P|\equiv Q|\equiv X}{P|\equiv X}$
  If the principal believes that Q has jurisdiction over X and Q believes X, then P believes that X is true.
- The session keys rule: $\frac{P|\equiv\sharp(X),P|\equiv Q|\equiv X}{P|\equiv P \overset{K}{\longleftrightarrow} Q}$
  If the principal P believes that the session key is fresh and the principal P and Q believes X,which are the necessary parameters of the session key, then principal P believes that he/she shares the session key K with Q.

To prove an authentication protocol secure, the following process should be performed:

- First, idealize the proposed authentication scheme in the language of formal logic.
- Second, identify the assumptions about the initial state of the proposed authentication scheme.
- Third, use the production and use of rules of the logic to deduce new predicates.
- Fourth, use logic to discover the beliefs held by the parties in the proposed scheme.

In order to prove the proposed protocol secure, the proposed protocol must satisfy the following goals based on the *BAN* logic which are given as follows:

- **Goal 1** $U_i \mid\equiv U_i \overset{SK}{\longleftrightarrow} RS$
- **Goal 2** $U_i \mid\equiv RS \mid\equiv U_i \overset{SK}{\longleftrightarrow} RS$
- **Goal 3** $SS_j \mid\equiv SS_j \overset{SK}{\longleftrightarrow} U_i$
- **Goal 4** $SS_j \mid\equiv U_i \mid\equiv SS_j \overset{SK}{\longleftrightarrow} U_i$

First the proposed protocol is transformed into idealized form:

M_1: $U_i \rightarrow RS : R_i, TID_i, SID_j, d_i : \langle M_i \rangle_{DID_i}$
M_2: $SS_j \rightarrow U_i : SID_j, D_i, E_i : \langle y_i \rangle_{G_i}$

Second, the following assumptions about the initial state of the protocol are made to analyze the proposed protocol:

$A_1 : U_i \mid\equiv \sharp(M_i, y_i, w_i)$
$A_2 : RS \mid\equiv \sharp(M_i, w_i, y_i)$
$A_3 : U_i \mid\equiv U_i \overset{DID_i}{\longleftrightarrow} RS$
$A_4 : SS_j \mid\equiv SS_j \overset{G_i}{\longleftrightarrow} U_i$
$A_5 : RS \mid\equiv U_i \Rightarrow M_i$
$A_6 : U_i \mid\equiv SS_j \Rightarrow y_i$

Third, the idealized form of the proposed protocol is analyzed based on the BAN logic rules and the assumptions. The main proofs are stated as follows:

M_1: $U_i \rightarrow RS : R_i, TID_i, SID_j, d_i : \langle M_i \rangle_{DID_i}$

According to seeing rule, we get

$S1 : RS \triangleleft R_i, TID_i, SID_j, d_i : \langle M_i \rangle_{DID_i}$

According to A3, S1 and message meaning rule, we get

$S2 : RS \models U_i \mid\sim M_i$

According to A2, S2 and freshness-conjuncatenation rule and nonce verification rule is applied, we get

$S3 : RS \mid\equiv U_i \mid\equiv M_i$, where M_i is the necessary parameter of the session key of the proposed protocol.

According to A5, S3 and the jurisdiction rule is applied, we get

$S4 : RS \mid\equiv M_i$

According to A2, S3 and the session key rule is applied, we get

$S5 : RS \mid\equiv U_i \overset{SK}{\longleftrightarrow} RS$ **(Goal 1)**

According to A2, S5 and nonce verification rule is applied, we get

$S6 : RS \mid\equiv U_i \mid\equiv U_i \overset{SK}{\longleftrightarrow} RS$ **(Goal 2)**
M_2: $SS_j \rightarrow U_i : SID_j, D_i, E_i : \langle y_i \rangle_{G_i}$

According to seeing rule, we get

$S7 : U_i \triangleleft SID_j, D_i, E_i : \langle y_i \rangle_{G_i}$

According to A4, S7 and message meaning rule, we get

$S8 : U_i \mid\equiv SS_j \mid\sim y_i$

According to A1, S8 and freshness-conjuncatenation rule and nonce verification rule is applied, we get

$S9 : U_i \mid\equiv SS_j \mid\equiv y_i$, where y_i is the necessary parameter of the session key of the proposed protocol.

According to A6, S9 and the jurisdiction rule is applied, we get

$S10 : U_i \mid\equiv w_i$

According to A1, S9 and the session key rule is applied, we get

$S11 : U_i \mid\equiv U_i \overset{SK}{\longleftrightarrow} SS_j$ **(Goal 3)**

According to A1, S11 and nonce verification rule is applied, we get

$S12 : U_i \mid\equiv SS_j \mid\equiv U_i \overset{SK}{\longleftrightarrow} SS_j$ **(Goal 4)**

The above discussion clearly proves the mentioned objectives using *BAN* logic and it is also clear that the proposed protocol achieves mutual authentication and session key agreement between the U_i and the SS_j.

7 Further Security Analysis and Discussion of the Proposed Protocol

In order to achieve complete security requirements [2], this section presents security analysis of the proposed protocol based on the valid assumptions mentioned in the threat model.

Theorem 1 *The proposed protocol preserves user anonymity and off-line password guessing attack based on the threat model and success probability of the $\mathcal{A}$ is enormously negligible.*

Proof It is our assumption that the proposed protocol uses guessable user's password PW_i and identity ID_i. Since PW_i and ID_i are protected by the non-invertible cryptographic one-way hash function, so the extraction of $\langle PW_i, ID_i \rangle$ is not feasible by the attacker $\mathcal{A}$. However, he/she may guess $\langle PW_i, ID_i \rangle$ from the known parameters of the proposed protocol description. Based on the threat model, an $\mathcal{A}$ knows lots of parameters $\langle Reg_i, S_i, d_i, K_i, T_iE_i \rangle$ during execution of the protocol. Now, it is our challenge that the $\mathcal{A}$ cannot guess in polynomial time and also cannot derive the $\langle ID_i, PW_i \rangle$.

- *From Reg_i*: We can define the parameter Reg_i as $Reg_i = h(UID_i \parallel HPW_i) = h(H(ID_i) \parallel H(PW_i \parallel ID_i))$. As the cryptographic hash function is non-invertible, it is not possible to derive ID_i from the Reg_i. It is also confirm that if an $\mathcal{A}$ wants to guess the ID_i, he/she has to guess ID_i and PW_i at a time which is not feasible in polynomial time described in the threat model. The probability to guess $\langle ID_i, PW_i \rangle$ at a time is approximately $\frac{1}{2^{12n}}$, which is enormously negligible.
- *From S_i*: We can define the parameter S_i as $S_i = DID_i + Z_i = S_{RS} \cdot UID_i + h(HPW_i) \cdot P$. In this case, the $\mathcal{A}$ has no knowledge of $\langle ID_i, PW_i \rangle$ and S_{RS}. Therefore, to obtain ID_i, he/she has to guess three parameters at a time which is more infeasible and the probability of guessing is more less than the previous case.

Similar to the above description, it is confirmed that the $\mathcal{A}$ cannot guess user's ID_i as $\langle d_i, K_i, T_i, E_i \rangle$ has three (3), two(2), three(3), two(2) respectively unknown parameters to the attacker. Hence, the probability of guessing is very very less.

On the other hand, the user's password is involved with the parameters $\langle Reg_i, S_i \rangle$ of the proposed protocol. So, we can proof in the similar way that $\mathcal{A}$ cannot derive or guess user's low entropy password in polynomial time.

The above description clearly states that an $\mathcal{A}$ has no way to derive user's original identity or password and the probability of guessing is enormously negligible. Hence, the Theorem 1 is proved. □

Theorem 2 *The proposed protocol protects user-server impersonation attack and achieves mutual authentication property based on the threat model.*

Proof It is our assumption that an $\mathcal{A}$ can trap the communicating message, as it is transmitted through the public channel and after some modification of the messages, he/she can re-transmit the message to the authenticator. If the re-transmitted message is authenticated by somehow, the $\mathcal{A}$ can smash the security system and access the server. We now assume that the login message $\langle R_i, TID_i, SID_j, d_i \rangle$ of the proposed protocol is transmitted to the RS and trapped it by an $\mathcal{A}$. At this time, he/she ($\mathcal{A}$) tries to re-compute the login message by changing the random nonce. However, the attacker cannot launch valid message without the knowledge of the parameters $\langle UID_i, DID_i \rangle$ of the proposed protocol. As described earlier, if $\mathcal{A}$ wants to guess the unknown parameters $\langle UID_i, DID_i \rangle$, the probability is enormously negligible. Therefore, the attacker cannot forge valid message $\langle R_i, TID_i, SID_j, d_i \rangle$ to the RS.

On the other hand, if an $\mathcal{A}$ wants to impersonate other communicating messages like $M_2 = \langle L_i, K_i, SID_j, T_i \rangle$ and $M_3 = \langle SID_j, K_i, D_i, E_i \rangle$ of the proposed protocol, he/she fails to launch valid messages, because the parameters $\langle UID_i, S_j, G_i \rangle$ is unknown to the attacker $\mathcal{A}$.

The above description clearly states that the proposed protocol is well protected against the user-server impersonation attacks, that means the attacker cannot forge valid transmitted messages to the desired entities. Therefore, the protocol achieves most desirable mutual authentication property between the U_i and the SS_j. Hence, the Theorem 2 is proved. □

7.1 Privilege Insider Attack

Most of the today's security system breaks due to insider attack. So, it is an important task to keep user's confidential information(s) secret from the server (though the server is trusted). If an insider of the system (system manager or administrator) gets the user's correct password by some means, then he/she may use the same password in others account of the others server, as most of the users use same password for a set of accounts. In our proposed protocol, we provide $HPW_i = h(PW_i \parallel ID_i) \cdot P \in G_1$ instead of original password to the RS during the registration phase. Therefore, the insider adversary cannot extract PW_i from the given HPW_i, as it is protected by the non-invertible cryptographic one-way hash function.

7.2 Session Key Discloser Attack

The authenticated session key is used for secure communication between the entities involved, and an attacker upon disclosure of the key can decrypt the secret information. So, the secrecy of session key is the mandatory property of any key agreement protocol. However, the session key of our proposed protocol is protected by the non-invertible cryptographic one-way hash function. Besides it, the computation of the session key is dependent upon the parameters $\langle UID_i, y_i, w_i \rangle$ which all are unknown to the $\mathcal{A}$. Therefore, the session key computation is not feasible by an attacker.

7.3 Efficient Login Phase

During the login procedure, the protocol uses wrong information detection mechanism [2], where the card reader verifies the user's identity and password after verifying the user's biometric template. However, if a valid user inputs wrong information by mistake, it will be quickly detected by the card reader and immediately rejects the session. Therefore, the protocol avoids extra computation and communication overheads as well as network congestion. Thus, the proposed protocol provides efficient login phase.

7.4 Efficient and User-Friendly Password Change Phase

In order to maintain security system safely, the user should change the password, and for doing that the proposed protocol has provided efficient and user-friendly password update phase. An user can change or update his/her password either without the help of the registration server or with the help of the registration server. But, it is most desirable to change the password without the help of the registration server, as it reduces the extra computation, communication overheads and network congestion as well. The proposed protocol provides password change phase to the user without the help of the *RS* and it is more efficient comparing to the Hsieh–Leu's scheme, as their protocol changes the password with the help of the *RS* and uses secure channel. Therefore, the password change phase of the proposed protocol is more efficient and user-friendly.

7.5 Single Registration

After executing user registration phase successfully to the *RS*, an user can access the several registered application servers based on his/her demand and it avoids multiple registration. Hence, the proposed protocol provides single registration feature to the system.

8 Performance Comparison

In this section, we compare the performance of the proposed protocol with some other existing related schemes [17–19, 25, 30, 31, 37, 40, 46] in terms of several security functionalities and computation cost. It is noted that the authentication protocol generally executes registration phase ($\langle U_i, SS_j \rangle$) only one-time. Therefore, we compare the complexity of the login and authentication phase in terms of computation. Though, the password change phase executes on user's demand, we have compare it with the Hsieh–Leu scheme. This paper applies some cryptographic operations whose notations are presented below:

T_{bp}: The time of executing a bilinear map operation.
T_h: The time of executing a cryptographic one-way hash operation.
T_{pm}: The time of executing a point multiplication operation on the group G_1.
T_m: The time of executing a integer multiplication operation.
T_{bh}: The time of executing a bio-hashing operation.

Table 2 Functionality and security comparison of the proposed scheme with the related schemes

Schemes ⇒	[30]	[31]	[18]	[17]	[40]	[19]	[25]	[37]	Proposed
A1	×	×	√	×	×	×	×	×	√
A2	×	×	×	×	×	×	√	×	√
A3	×	×	×	×	×	×	×	×	√
A4	√	×	√	√	×	√	×	×	√
A5	√	√	×	√	√	×	√	√	√
A6	√	√	√	√	√	√	√	√	√
A7	√	√	×	×	√	√	×	×	√

A1, resist off-line password guessing attack; A2, resist user anonymity; A3, resist impersonation attack; A4, achieves mutual authentication with session key agreement; A5, provides efficient and user-friendly password change phase; A6, single registration; A7, pre-authentication; √, resists corresponding attack or satisfy corresponding property; ×, cannot resist corresponding attack or not satisfy corresponding property

Table 3 Computation cost comparison of the proposed protocol with other related protocols

Schemes ⇓	Computation cost
Liao–Hsiao [30]	$2T_{bp} + 10T_{pm} + 7T_h + 1TG_H + 2T_{padd}$
Tseng et al. [40]	$2T_{bp} + 4T_{pm} + 1TG_H + 2T_{padd} + 3T_h$
Geng–Zhang [17]	$4T_{bp} + 10T_{pm} + 3TG_H + 3T_{padd} + 8T_h$
Zhao et al. [46]	$2T_{bp} + 12T_{pm} + 1TG_H + 2T_{padd} + 7T_h$
Hsieh–Leu [19]	$2T_{bp} + 14T_{pm} + 4T_m + 15T_h$
Proposed	$2T_{bp} + 11T_{pm} + 18T_h + 1T_{bh}$

T_{bp}, bilinear pairing operation; T_{pm}, scalar point multiplication operation; T_m, multiplication operation; T_{padd}, point addition on group; G_1, T_{bh}, bio-hashing operation; T_h, one-way hash function; TG_H, map to point hash function

TG_H: The time of executing a map to point hash function.
T_{padd}: The time of executing a point addition on the group G_1.

In Table 2, we have presented several functionalities comparison of the proposed protocol with other related protocols and it is noticeable that the proposed protocol resists relevant security attacks and achieves several security attributes than other schemes. Additionally, the Table 2 confirms that the proposed protocol resists the mentioned security weaknesses of the Hsieh–Leu scheme.

In Table 3, we have presented computation cost comparison of the proposed protocol with several related protocols. In order to measure computation cost, the time complexity associated with these operation can be roughly expressed as $T_{bp} \gg T_{pm} \gg T_m \gg T_h \approx T_{bh}$. The computation cost including (login and authentication phases) of the Hsieh–Leu scheme and proposed scheme are $(2T_{bp} + 15T_h + 14T_{pm} + 4T_m)$ and $(2T_{bp} + 18T_h + 1T_{bh} + 11T_{pm})$ respectively. Moreover, the computation cost for the password change phase of the Hsieh–Leu scheme takes $(2T_{bp} + 11T_h + 7T_{pm})$, whereas the proposed protocol takes negligible computation, as it does not compute any bilinear pairing operation. It is noticeable that the proposed protocol is relatively better than others specially Hsieh–Leu scheme in terms of computation complexity. This paper uses *SHA-2* cryptographic one-way hash function for achieving top security, and its message digest is 160 bits. As the protocol provides strong security protection on the relevant security attacks and authenticates the U_i twice, the proposed protocol takes little more communication cost than Hsieh–Leu's scheme.

9 Conclusion

This paper discusses several security attacks including inefficient password change phase of the scheme proposed by Hsieh–Leu. Thereafter, we propose a three factors based user authentication and key agreement scheme using bilinear pairing usable in multi-server architecture to remove the mentioned security pitfalls of the Hsieh–Leu scheme. The security validation of the proposed scheme has shown using *BAN* logic which confirms that the proposed protocol achieves mutual authentication and session key agreement securely. Further, the informal cryptanalysis proves the resilience of relevant security attacks. The performance analysis section shows that the protocol is more secure and efficient than other related schemes. Moreover, it is also noticeable that the proposed protocol can change the password on user's demand without the help of the registration server and also it takes negligible computation cost. The overall performance and different security aspects of the proposed protocol make the authentication system so efficient that it can be implemented in practical application.

Acknowledgments The authors are grateful to the editor and anonymous reviewers for their valuable suggestions and comments which improved the paper.

References

1. Amin, R. (2013). Cryptanalysis and an efficient secure ID-based remote user authentication using smart card. *International Journal of Computer Applications*, *75*(13), 43–48.
2. Amin, R., & Biswas, G. P. (2015). A novel user authentication and key agreement protocol for accessing multi-medical server usable in TMIS. *Journal of Medical Systems*, *39*(3), 33. doi:10.1007/s10916-015-0217-3.
3. Amin, R., & Biswas, G. P. (2015). Remote access control mechanism using rabin public key cryptosystem. In *Information systems design and intelligent applications, advances in intelligent systems and computing* (vol. 339, pp. 525–533). Springer. doi:10.1007/978-81-322-2250-7_52.
4. Amin, R., Maitra, T., & Giri, D. (2013). An improved efficient remote user authentication scheme in multi-server environment using smart card. *International Journal of Computer Applications*, *69*(22), 1–6.
5. Awasthi, A. K., & Lal, S. (2004). An enhanced remote user authentication scheme using smart cards. *IEEE Transactions on Consumer Electronics*, *50*(2), 583–586.
6. Badra, M., & Urien, P. (2004). Introducing smartcards to remote authenticate passwords using public key encryption. In *Advances in wired and wireless communication, 2004 IEEE/Sarnoff symposium on* (pp. 123–126). doi:10.1109/SARNOF.2004.1302856.
7. Boneh, D., & Franklin, M. (2001). Identity-based encryption from the weil pairing. In *Advances in cryptology CRYPTO 2001, lecture notes in computer science* (vol. 2139, pp. 213–229). Springer, Berlin. doi:10.1007/3-540-44647-8_13.
8. Boyd, C., & Mathuria, A. (2003). In *Protocols for authentication and key establishment*. doi:10.1007/978-3-662-09527-0.
9. Burrows, M., Abadi, M., & Needham, R. (1990). A logic of authentication. *ACM Transactions on Computer Systems*, *8*(1), 18–36. doi:10.1145/77648.77649.
10. Chang, C. C., & Wu, T. C. (1991). Remote password authentication with smart cards. *IEE Proceedings E Computers and Digital Techniques*, *138*(3), 165–168.
11. Chang, Y. F., Yu, S. H., & Shiao, D. R. (2013). A uniqueness-and-anonymity-preserving remote user authentication scheme for connected health care. *Journal of Medical Systems*, *37*(2), 1–9. doi:10.1007/s10916-012-9902-7.

12. Lee, C.-C., Lai, Y.-M., & Li, C. T. (2011). An improved secure dynamic ID based remote user authentication scheme for multi-server environment. *Expert Systems with Applications*, *38*(11), 203–209.
13. Cheon, J. H., & Lee, D. H. (2002). Diffie-hellman problems and bilinear maps. Cryptology ePrint Archive: Report 2002.
14. Chien, H. Y., Jan, J. K., & Tseng, Y. M. (2002). An efficient and practical solution to remote authentication: Smart card. *Computers and Security*, *21*(4), 372–375. doi:10.1016/S0167-4048(02)00415-7.
15. Eisenbarth, T., Kasper, T., Moradi, A., Paar, C., Salmasizadeh, M., & Shalmani, M. (2008). On the power of power analysis in the real world: A complete break of the keeloq code hopping scheme. In *Advances in cryptology CRYPTO 2008, lecture notes in computer science* (vol. 5157, pp. 203–220). Springer, Berlin. doi:10.1007/978-3-540-85174-5_12.
16. Frey, G., & Rück, H. G. (1994). A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. *Mathematics of computation*, *62*(206), 865–874. doi:10.2307/2153546.
17. Geng, J., & Zhang, L. (2008). A dynamic ID-based user authentication and key agreement scheme for multi-server environment using bilinear pairings. In *Power electronics and intelligent transportation system, 2008. PEITS '08. Workshop on* (pp. 33–37). doi:10.1109/PEITS.2008.35.
18. Hsiang, H. C., & Shih, W. K. (2009). Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment. *Computer Standards and Interfaces*, *31*(6), 1118–1123. doi:10.1016/j.csi.2008.11.002.
19. Hsieh, W. B., & Leu, J. S. (2014). An anonymous mobile user authentication protocol using self-certified public keys based on multi-server architectures. *The Journal of Supercomputing*, *70*(1), 133–148. doi:10.1007/s11227-014-1135-8.
20. Islam, S. (2014). A provably secure ID-based mutual authentication and key agreement scheme for mobile multi-server environment without esl attack. *Wireless Personal Communications*, *79*(3), 1975–1991. doi:10.1007/s11277-014-1968-8.
21. Jin, A. T. B., Ling, D. N. C., & Goh, A. (2004). Biohashing: Two factor authentication featuring fingerprint data and tokenised random number. *Pattern Recognition*, *37*(11), 2245–2255.
22. Juang, W. S. (2004). Efficient multi-server password authenticated key agreement using smart cards. *IEEE Transactions on Consumer Electronics*, *50*(1), 251–255. doi:10.1109/TCE.2004.1277870.
23. Kocher, P., Jaffe, J., & Jun, B. (1999). Differential power analysis. In *Advances in cryptology CRYPTO 99, lecture notes in computer science* (vol. 1666, pp. 388–397).
24. Lee, W. B., & Chang, C. C. (2000). User identification and key distribution maintaining anonymity for distributed computer network. *Computer and System Science*, *15*(4), 211–214.
25. Lee, C. C., Lin, T. H., & Chang, R. X. (2011). A secure dynamic ID based remote user authentication scheme for multi-server environment using smart cards. *Expert Systems with Applications*, *38*(11), 13,863–13,870. doi:10.1016/j.eswa.2011.04.190.
26. Li, X., Ma, J., Wang, W., Xiong, Y., & Zhang, J. (2010). A novel smart card and dynamic ID based remote user authentication scheme for multi-server environments. Mathematical and Computer Modelling 58(1–2), 85–95 (2013). doi:10.1016/j.mcm.2012.06.033. Financial IT and security and international symposium on computational electronics.
27. 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. doi:10.1007/s11277-014-2002-x.
28. Li, X., Qiu, W., Zheng, D., Chen, K., & Li, J. (2010). Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards. *IEEE Transactions on Industrial Electronics*, *57*(2), 793–800. doi:10.1109/TIE.2009.2028351.
29. Li, X., Xiong, Y., Ma, J., & Wang, W. (2012). An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards. *Journal of Network and Computer Applications*, *35*(2), 763–769. doi:10.1016/j.jnca.2011.11.009.
30. Liao, Y. P., & Hsiao, C. M. (2013). A novel multi-server remote user authentication scheme using self-certified public keys for mobile clients. *Future Generation Computer Systems*, *29*(3), 886–900.
31. Liao, Y. P., & Wang, S. S. (2009). A secure dynamic ID based remote user authentication scheme for multi-server environment. *Computer Standards and Interfaces*, *31*(1), 24–29. doi:10.1016/j.csi.2007.10.007.
32. Lin, I. C., Hwang, M. S., & Li, L. H. (2003). A new remote user authentication scheme for multi-server architecture. *Future Generation Computer Systems*, *19*(1), 13–22. doi:10.1016/S0167-739X(02)00093-6.
33. Lumini, A., & Nanni, L. (2007). An improved biohashing for human authentication. *Pattern Recognition*, *40*(3), 1057–1065. doi:10.1016/j.patcog.2006.05.030.
34. Menezes, A., Vanstone, S., & Okamoto, T. (1991). Reducing elliptic curve logarithms to logarithms in a finite field. In *Proceedings of the twenty-third annual ACM symposium on theory of computing, STOC '91* (pp. 80–89). doi:10.1145/103418.103434.

35. Messerges, T. S., Dabbish, E. A., & Sloan, R. H. (2002). Examining smart-card security under the threat of power analysis attacks. *IEEE Transactions on Computers*, *51*(5), 541–552.
36. Pippal, R., Jaidhar, C., & Tapaswi, S. (2013). Robust smart card authentication scheme for multi-server architecture. *Wireless Personal Communications*, *72*(1), 729–745. doi:10.1007/s11277-013-1039-6.
37. Shao, M.H., & Chin, Y.C. (2010). A novel approach to dynamic ID-based remote user authentication scheme for multi-server environment. In *Proceedings of the 2010 fourth international conference on network and system security, NSS '10* (pp. 548–553). IEEE Computer Society, Washington, DC, USA. doi:10.1109/NSS.2010.95.
38. Sood, S. K., Sarje, A. K., & Singh, K. (2011). A secure dynamic identity based authentication protocol for multi-server architecture. *Journal of Network and Computer Applications*, *34*(2), 609–618. doi:10.1016/j.jnca.2010.11.011. Efficient and Robust Security and Services of Wireless Mesh Networks.
39. Tsai, J. L., Wu, T. C., & Tsai, K. Y. (2010). New dynamic ID authentication scheme using smart cards. *International Journal of Communication Systems*, *23*(3), 1449–1462.
40. Tseng, Y. M., Wu, T. Y., & Wu, J. (2008). A pairing-based user authentication scheme for wireless clients with smart card. *Informatics*, *19*(2), 285–302.
41. Tsuar, W. J., Chang, C. C., & Wu, W. L. (2001). A flexible user authentication scheme for multi-server internet services. In *Networking ICN 2001, lecture notes in computer science* (vol. 2093, pp. 174–183). Springer, Berlin. doi:10.1007/3-540-47728-4_18.
42. Turkanovic, M., Brumen, B., & Hölbl, M. (2014). A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the internet of things notion. *Ad Hoc Networks*, *20*, 96–112. doi:10.1016/j.adhoc.2014.03.009.
43. Wang, B., & Ma, M. (2013). A smart card based efficient and secured multi-server authentication scheme. *Wireless Personal Communications*, *68*(2), 361–378. doi:10.1007/s11277-011-0456-7.
44. Wei, J., Liu, W., & Hu, X. (2014). Cryptanalysis and improvement of a robust smart card authentication scheme for multi-server architecture. *Wireless Personal Communications*, *77*(3), 2255–2269. doi:10.1007/s11277-014-1636-z.
45. Xu, J., Zhu, W. T., & Feng, D. G. (2009). An improved smart card based password authentication scheme with provable security. *Computer Standards and Interfaces*, *31*(4), 723–728. doi:10.1016/j.csi.2008.09.006.
46. Zhao, D., Peng, H., Li, S., & Yang, Y. (2013) An efficient dynamic ID based remote user authentication scheme using self-certified public keys for multi-server environment. CoRR abs/1305.6350. http://arxiv.org/abs/1305.6350

Ruhul Amin received his B.Tech. and M.Tech. degree from West Bengal University of Technology in computer science and engineering department in 2009 and 2013 respectively. Currently, he is a pursing Ph.D. in the Department of CSE, Indian school of mines, Dhanbad, India. He has published several research papers in Journals and Conference proceedings of International reputes. His current research interests include Cryptographic authentication protocol and security in wireless sensor network.

G. P. Biswas received B.Sc. (Engg.) and M.Sc. (Engg.) degrees in Electrical and Electronics Engineering and Computer Science and Engineering, respectively. He completed his PhD degree in Computer Science and Engineering from Indian Institute of Technology, Kharagpur, India. He is currently working as a Professor in the Department of Computer Science and Engineering, Indian school of Mines, Dhanbad, Jharkhand, India. His main research interests include Cryptography, Computer Network and Security, Cellular Automata, VLSI Design.