A provable and secure mobile user authentication Scheme for Mobile Cloud Computing Services

Azeem Irshad[1], Shehzad Ashraf Chaudhry*[1], Muhammad Shafiq[2], Muhammad Usman[3],
Muhammad Asif [4], Anwer Ghani[1]

irshadazeem2@gmail.com, shahzad@iiu.edu.pk, shafiq.pu@gmail.com,
musman@qau.edu.pk, asif@ntu.edu.pk, anwer.ghani@iiu.edu.pk

[1]*Department of Computer Science & Software Engineering, International Islamic University, Islamabad, Pakistan*
[2]*Department of Information and Communication Engineering, Yeungnam University, Gyeongsan, South Korea*
[3]*Department of Computer Science, Faculty of Natural Science, Quaid-I-Azam University, Islamabad, Pakistan*
[4] *Department of Computer Science, National Textile University, Faisalabad, Pakistan*
.

Abstract.

The mobile cloud computing (MCC) has enriched the quality of services that the clients access from remote cloud-based servers. The growth in the number of wireless users for MCC has further augmented the requirement for a robust and efficient authenticated key agreement mechanism. Formerly, the users would access cloud services from various cloud-based service providers and authenticate one another only after communicating with the trusted third party (TTP). This requirement for the clients to access the TTP during each mutual authentication session, in earlier schemes, contributes to the redundant latency overheads for the protocol. Recently, Tsai et al. have presented a bilinear pairing based Multi-Server Authentication (MSA) protocol, to bypass the TTP, at least during mutual authentication. The scheme construction works fine, as far as the elimination of TTP involvement for authentication has been concerned. However, Tsai et al. scheme has been found vulnerable to server spoofing attack, De-synchronization attack, and lack smart card-based user verification, which renders the protocol inapt for practical implementation in different access networks. Hence, we have proposed an improved model designed with bilinear pairing operations, countering the identified threats as posed to Tsai scheme. Additionally, the proposed scheme is backed up by performance evaluation and formal security analysis.

Keywords: authentication, mobile cloud computing, cryptanalysis, attacks, security, bilinear pairing

1. INTRODUCTION

The number of wireless gadgets is going to exceed the wired ones due to the flourishing mobile cloud computing environment, for about 50 percent of the total IP based traffic, by the end of year 2016. The mobile cloud computing environment enables the subscribers on the fly, to access the online cloud-based applications and services, which not only enhances the quality of service for clients, but also help generating revenues for the service providers. According to a report conducted by ABI [4-5], the number of mobile broadband users will exceed 5 billion till 2017, and this can be attributed to MCC. In MCC [1, 3], the cloud-based services may be accessed with the use of mobile devices employing Ethernet or 3G/4G based telecommunication links. A user may prompt for the use of cloud computing service, by employing a web browser or any sort of cloud service application installed on its mobile device. In that case, the user application and MCC service application mutually authenticate one another. In this connection, we can witness many authentication techniques introduced so far, for MCC [10-13]. As a matter of fact, these protocols should be designed with the consideration of low computing devices, which also meet the security requisites [2, 6-9]. The security is of vital concern, as the messages pass through an insecure domain of WLAN or telecommunication links, where the attackers may easily intercept the messages to initiate several attacks. Besides, these authentication protocols should consider the privacy and identity based concerns.

It is hard to register all of the service providers and maintain multiple passwords for those services when there is more than one cloud computing service providers. The management of tens and hundreds of passwords for different service providers might be pretty troublesome for the users, in a distributed mobile cloud environment. In this regard, conventional single sign-on (SSO) protocols e.g., Passport and OpenID are one of the likely key management techniques [14-22]. The users can avail multiplicity of services, in those systems, by remembering just a single secret key or password. However, the majority of SSO based techniques engage a trusted party to establish an authenticated session. At the same time, OpenID a decentralized SSO protocol [26-33] is still being used by some major web-based service providers i.e., Yahoo and Google, for the management of nearly more than 50000 websites for authentication purpose. The three of the entities, i.e., user, relaying partner i.e., service provider (SP) and Identity provider (IdP) take part in the mutual authentication phase between user and service provider [53]. The RC and service provider can act as RC and SP to serve the user on alternate basis, in OpenID. A user who gets registered with IdP for OpenID identifier, could log into several websites, based on OpenID, that use Secure Socket Layer (SSL) protocol on a secure channel [32]. The user, while performing mutual authentication phase with SP, employ IdP, and it initially has to send a login request towards SP. The SP, after checking the OpenID identifier, forwards that authentication request towards IdP for further verification. The IdP verifies and acknowledges to user and SP, in case the identity is found legitimate in its repository. Next, the user and SP mutually authenticate each other. However, it might add further delay, if the IdP is burdened with too many awaiting authentication requests, which may further disturb the service provisioning of server. The employment of SSO would require a secure message transmission protocol to operate reliably in this environment. As we know that SSL rely on Rivest, Shamir and Adleman (RSA) for the purpose of authentication, which is public key cryptographic costly technique regarding computation. So it will be an expensive technique to employ in the system, in its current state.

1.1 Objectives

Thus, the objectives for this work are described as under:

1) We need a less computational-intensive protocol for better practical implications.
2) We require a control authority (RC in our scenario) that registers the subscribers initially, while those subscribers may benefit from the services of service providers, onwards.
3) We need to select a low entropy password for user so it could be used to avail many services from various service providers, as warranted by multi-server authentication paradigm. The hassle of maintaining many passwords for several servers is relieved by employing this paradigm.
4) In the last, we need to develop the protocol that does not rely on database verifiers at server's end.

1.2 Related Work

The authentication being the most significant component in security properties enables the subscribers to avail secure network services. We may witness many recent public key cryptography-based authenticated key agreements involving DLP (Discrete Logarithm Problem) [52] and RSA etc. Nonetheless, these protocols were less effective for large key sizes. The Elliptic Curve Cryptography (ECC) offers the same security strength in much lesser key size. i.e., the public-key size with 3072-bit for RSA offers an equivalent security strength as 256-bit public-key size of ECC does. The MCC gadgets requires less power consuming-security solutions, at the same time the elliptic curve cryptography is regarded as one of the best candidates for the frameworks requiring efficient wireless gadgets with less powerful processor.

Thus besides efficient cryptographic algorithms like ECC, DLP or Chebyshev map, we require a protocol that engages trusted third party merely in registration phase, but not in login and authentication phase. Likewise, we do not need to maintain a repository of password verifiers on the side of RC, At the same time, no password verifier table or database should be being maintained at RC's end as maintaining those certificates needs overhead cost and may involve risk of stealing by any adversary.

Meanwhile, to meet the above objectives, an identity based cryptosystem employing bilinear pairing operations is demonstrated by the researchers to achieve the aforementioned goals. In identity-based cryptography, the identity of subscriber acts as its public key, where the corresponding private key is constructed by a key central entity employing the related subscriber's identity and is forwarded to that subscriber in registration procedure.

The identity based cryptography foregoes the requirement to confirm the validity of public key of some subscriber using any public key certificate or asking any assistance from some external source, or keeping the certificate in its database for some period. We could witness several applications of identity based cryptography in internet of things, cloud computing, grid computing, wireless sensor networks, and group signatures, etc. Initially, in 2004 for grid computing the related identity based scheme was pioneered by Lim and Robshaw [21, 22]. Thereafter, Mao [23] also presented an ID-based protocol in Grid. Thereafter, Li et al. [24] came up with another ID-based authenticated key agreement protocol for cloud environment, yet it does not provide anonymity-based features [25, 35, 51].

Mostly, the authentication protocols are based on single-server authentication paradigms that render these protocols unable to fit in multi-server environment. In such multi-server environment-based protocols, a subscriber does not need to remember more than one password as much as the number of servers [56]. A user might benefit from services of various service providers using a single password. Previously, a user seeks to consult TTP each time it acquires the services of service provider. In addition, we observe that there are few protocols that share a single master key among all servers in a network that might enable the adversary, a malicious server, to initiate impersonation attack. So these previous schemes are unable to address the problems, although these were efficient in terms of computation. In multi-server authentication, Li et al. [61] presented the pioneer concept about this authentication paradigm. Thereafter, Lin et al. [62] criticized the above scheme for neural network-based implementation and presented its own scheme. Next, Cao and Zhong [63] found impersonation attacks in [62] and presented another improved scheme. Likewise, many symmetric key schemes were presented in a row. Later, public-key cryptography based schemes were demonstrated involving ECC and RSA-based operations. Regarding this, Yoon and Yoo [64] presented a novel multi-server authentication scheme. Nonetheless, Yoon and Yoo's protocol is vulnerable to impersonation attacks. Afterwards, He and Wang [65] put another efficient protocol employing ECC operations. Thereafter, Odelu et al. [66] discovered that He and Wang protocol does not provide anonymity and is also susceptible to few attacks. All of the above mentioned schemes could not fulfill the requirement of multi-server authentication. Still there was requirement for a secure, efficient and anonymous protocol in multi-server paradigm. In this context, lately Tsai et al. [34] proposed a bilinear pairing-based authentication scheme for mobile cloud computing. We discover that it suffers from server spoofing or impersonation threat and de-synchronization threat. Besides, it also lacks smart card verification which renders the protocol ineffective for implementation. In this study, we review Tsai et al.'s protocol besides crypt-analyzing the scheme. Then we propose an enhanced mobile user authenticated key agreement protocol for mobile cloud computing environment that employs the bilinear pairing operations. Finally our contributed scheme presents performance evaluation analysis along with formal security analysis.

1.3 Threat Model

We assume few assumptions regarding an adversary Ă under the current threat model:

1) An adversary may intercept and examine the contents exchanged on a public channel among the legitimate parties.

2) The adversary might could manipulate the contents, i.e. delete, replay or alter the contents during the communication.
3) The adversary might be a privileged or legitimate malicious insider in an organization.
4) It is also supposed that adversary knows about the protocol.
5) The adversary is capable of guessing low-entropy strings including passwords and identities; nonetheless, it might not guess high-entropy random numbers in polynomial amount of time.
6) In the last, the adversary could steal and manipulate the smart card details.

1.4 Paper layout

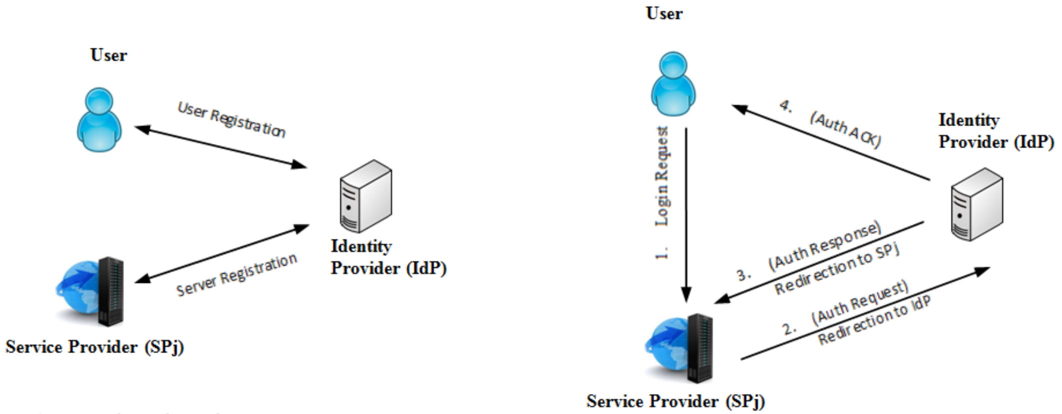
This paper is organized as follows: The section 2 illustrates the preliminary details regarding this paper. The section 3 relates to reviewing and cryptanalysis for Tsai et al.'s protocol. Section 4 demonstrates our contributed scheme. Section 5 and 6 depicts informal and formal security discussion and analysis, along with performance evaluation analysis. Finally, the conclusion is presented.

2 PRELIMINARIES

We briefly take a review of ID-based protocol working, bilinear pairing operation, bio-hashing operation, and hash digest function in this section.

2.1 ID-based protocol framework

The Figure 1 depicts the ID-based protocol environment, where each user, in the beginning, registers with the Registration Centre (RC). In registration process, smart card is issued by RC bearing the private key as generated by the RC based on the identity of the user. Thereafter, the user avails the required services from various service providers, by login and authentication procedure using the same account as established with RC. Here, the subscriber may authenticate itself with server S, only through engaging RC during each session. At times, this frequent communication on regular basis becomes a bottleneck, as the user has to face extra communication delay, whenever it needs to avail or acquire any service from any service provider.



a) Registration Phase

b) Authentication Phase

Fig. 1. User and Server Authentication Steps employing OpenID

2.2 Bilinear Pairing Operation

In bilinear pairing [62], there are two well-known types, the Weil pairings or Tate pairings that are utilized for identity based cryptography. We take (G_1, +) as an additive cyclic group, while (G_2, ×) as multiplicative cyclic group, and the symbol P being a generator of G_1 group. The bilinear mapping such as $e: G_1 \times G_1 \rightarrow G_2$ bears the understated properties:

1. Bilinear: For all $D, E, F \in G_1$, $e(D+E, F) = e(D,F) \times e(E,F)$ and $e(D, E+F)=e(D,E) \times e(D,F)$.
2. Non-degeneracy: Supposedly, 1 being an identity element for multiplicative cyclic-group G_2, then this group holds the feature as $D, E \in G_1$, where $e(D, E) \neq 1$.
3. Computability: There exist an algorithm to compute $e(D, E)$ for D, E corresponding to G_1.

2.3 Bio-hashing

The Biohashing function [49] takes the subscriber's biometric properties and maps it on randomly generated vectors that further enable to produce a particular user code called as biocode. This code enables in generating one or zero-based discrete projection coefficients. This Bio-hashing function works much alike hashed password, however more secure for covering biometric aspects. Jina et al. first employed two-factor-based authenticator and iterated inner products to generate token-based pseudo random integers and unique biometric features and ultimately the compact codes. After Jina et al. [49], Lumini and Nanni [50] developed this concept as an updated biohashing technique.

2.4 Hash digest function

The hash digest function, say h: ($x \rightarrow y$) bears the under-mentioned characteristics:

1. The hash digest function h takes input an integer or character-based string of arbitrary length and generates another string of fixed length.
2. With the given function $h(x)=y$, it is hard to calculate $h^{-1}(y)=x$ in polynomial amount of time.
3. With the given integer u, it is hard to output x', such that x' is not equal to x, nonetheless $h(x')$ remains equal to $h(x)$;
4. Lastly, it is hard to find a pair x, x' provided x' is not equal to x, and $h(x')$ is equal to $h(x)$.

3 REVIEW OF TSAI ET AL. SCHEME WITH CRYPTANALYSIS

The Tsai et al.'s protocol [34] consists of three participants in multi-server system, i.e. user (Ui), server (SPj) and Registration centre (RC). Here, we use the term RC instead of IdP or smart card generator (SCG) in Tsai et al. scheme. The participants, i.e. Ui and SPj perform registration procedure initially on secret channel, and then these may mutually authenticate one another on public channel without engaging the registration centre. In this section, we first illustrate the system setup for Tsai et al.'s protocol [34] and then describe its working and critical analysis. We narrate few symbols in Table I that have been used in this paper.

3.1 System Initialization

We take a cyclic additive-group as G_1 which is build on P generator, while a cyclic multiplicative group as G_2, where the integer p describes the prime order for both groups. First, the registration centre chooses s as its master key and generates the corresponding public key i.e. $P_{pub} = sP$. Afterwards, it calculates $e(P, P)$ and bilinear pairing functions such as $e: G_1 \times G_1 \rightarrow G_2$, besides some hash digest functions as $H_1: Z_p \rightarrow Z_p$, $H_2: G_2 \rightarrow Z_p$, $H_3: Z_p \rightarrow Z_p$, $H_4: Z_p \rightarrow Z_p$, $h: Z_p \rightarrow G_1$. Ultimately, registration centre make public these parameters as $\{e, h, P_{pub}, P, H_1 - H_4, e(P,P)\}$.

Table I. Notations description

Symbols	*Meanings*
Ui, SPj, RC	i^{th} User, j^{th} Service provider, Registration Centre
IDi , *IDj*	Identities of user and server
PWi, bi	Password and biometric finger impression of Ui
$e\colon G_1 \times G_1 \rightarrow G_2$	G_1 and G_2 are additive and multiplicative-cyclic groups in a bilinear mapping
K_i, K_j:	User's private secret , Server's private secret
$H_1(IDi)$, $H_1(IDj)$	User's public key, Server's public key
s, P_{pub}	Private and public key of Registration Centre
u, v	Server's and user's temporary secrets
SC	Smart Card
$H_b()$	Bio-hashing function
$H()$	Private hash-digest function
$h(.)$	A secure hash function
$\oplus$, $\|\|$	XOR operation, Concatenation operation
+	Point Addition operation

3.2 Tsai et al.'s scheme working [34]

The Tsai et al. based protocol comprises three sub-phases such as registration, mutual authentication phase.

3.2.1 Registration Phase

In this procedure, user or server submits registration request towards registration centre. Once the latter receives the registration request, it constructs the private keys for user and server by utilizing master key *s* receiving the request as given under:

$$K_i = \frac{1}{s + H_1(IDi)} P$$

Subsequently, the registration centre submits the K_i factor to user or server using a confidential channel. The Ui, after getting the private secret from registration centre, calculates $Ei = Ki \oplus h(PWi \,||\, bi)$. Onwards, it deposits the parameter *Ei* on smart card, here *PWi* symbolizes as password and *bi* as the user's fingerprint. Similarly, SPj stores the received private key in its memory for future use.

3.2.2 Mutual Authentication Steps:

1. Primarily, in this sub-phase, Ui submits login query towards server.
2. Then, the server calculates $\boldsymbol{Z = e(P, P)^u}$ and submits towards user.
3. The user then, calculates $\boldsymbol{Kij}$, $\boldsymbol{L_2}$, $\boldsymbol{X}$, $\boldsymbol{R_i}$ and $\boldsymbol{M_1}$ as follows:

$$Kij = H_2(Z^{v}) = H_2(e(P,P)^{uv}) \tag{1}$$

$$L_2 = vP_{pub} + H_1(IDj)vP, \tag{2}$$

$$X = vP_{pub} + H_1(IDi)vP, \tag{3}$$

$$R_i = \frac{1}{v + H_3(IDi \,||Z||\, IDj \,||\, X \,||\, Kij)} K_i \tag{4}$$

$$M_1 = Kij \oplus (IDi \,||\, R_i \,||\, X) \tag{5}$$

In the above equation, $\boldsymbol{v}$ is taken as a random integer, while the user constructs the above factors and submits <$\boldsymbol{L_2}$, $\boldsymbol{M_1}$> to server. The factor $\boldsymbol{v}$ might be pre-selected, and similarly the other computed factors as

vP_{pub}, vP, and $vH_1(IDi)P$ before the login and authentication process is initiated, for reducing the computational cost of protocol.

4. After getting the message $<L_2, M_1>$ of user, the server calculates session key Kij as shown below.

$$Kij = H_2\,(e\,(L_2, K_j)^u) = H_2\,(\,e(P, P)^{uv}\,) \tag{6}$$

Next, the server extracts $(IDi \,||\, R_i \,||\, X)$ after calculating $(IDi \,||\, R_i \,||\, X) = Kij \oplus (M_1)$. The server, thereafter, calculates $e(R_i,\ X + H_3\ (IDi \,||\, Z \,||\, IDj \,||\, X \,||\, Kij)\ Yi)$ and verifies this with the parameter $e(P, P)$, for instance,

$$e(R_i, X + H_3\,(IDi\,||Z||IDj\,||X\,||Kij)Y_i) \overset{?}{=} e(P, P) \tag{7}$$

While, Y_i is calculated as $Y_i = (P_{pub} + H_1\,(\,IDi\,)\,P)$. Then, the server calculates $G_i = H_4\,(Kij \,||\, Z \,||\, IDi \,||\, IDj)$ and submits G_i to Ui.

5. The user gets G_i and calculates G_i' as

$$G_i' = H_4\,(Kij \,||\, Z \,||\, IDi \,||\, IDj) \tag{8}$$

Further, it matches the parameter Gi' against Gi. If the equality is successful, then the Ui authenticates SPj as a legitimate server.

3.3 Review of Tsai et al.'s protocol

The Tsai et al.'s protocol is a MCC-based mobile user authenticated key agreement protocol employing bilinear pairing operations. Nonetheless, the Tsai et al.'s protocol is discovered to be susceptible to server impersonation attack, de-synchronization attack, and also lack smart card-based user verification, as described below.

3.3.1 Server Impersonation Threat

The attacker Ă might initiate an impersonation or spoofing attack against any subscriber after impersonating as SPj, employing the under-mentioned steps.

1. An attacker, upon intercepting the login query from any subscriber, may construct Z by calculating a bilinear map as given in equation (9). The factor Z is then submitted to Ui.

$$Z = e(P_{pub} + H_1\,(IDj)\ P, P)^u \tag{9}$$

2. Then, Ui gets Z from attacker, considering it as a legal SPj, and calculates Kij, L_2, X, R_i and M_1.

$$Kij = H_2\,(Z^{\,v}) = H_2\,(e(P, P)^{uv}), \tag{10}$$

$$L_2 = vP_{pub} + H_1\,(IDj)vP, \tag{11}$$

$$X = vP_{pub} + H_1\,(IDi)vP, \tag{12}$$

$$R_i = \frac{1}{v + H_3\,(IDi\,||Z\,||IDj\,||\,X\,||\,Kij)}K_i \tag{13}$$

$$M_1 = Kij \oplus (IDi \,||\, R_i \,||\, X), \tag{14}$$

Next, Ui submits $<L_2, M_1>$ towards SPj in order to verify and authenticate it, as seized by Ă.

3. Afterwards, Ă gets the message $<L_2, M_1>$ and calculates the parameter Kij^* using L_2, u and P as depicted in **(15)**.

$$Kij^* = H_2(e(L_2, P)^u) \tag{15}$$

4. The attacker computes the identity of user as IDi using equation (**16**).

$$(IDi \parallel R_i \parallel X) = Kij^* \oplus M_1 \qquad (16)$$

Then the attacker calculates G_i^* and submits toward user in response to Ui's challenge as depicted in equation (17).

$$G_i^* = H_4\,(Kij^* \parallel Z \parallel IDi \parallel IDj) \qquad (17)$$

5. Thereafter, Ui gets G_i^* from the attacker Ă, and calculates $G_i = H_4\,(Kij \parallel Z \parallel IDi \parallel IDj)$. Now Ui verifies the authenticity by comparing parameters G_i^* and G_i. If it is successful, Ui authenticates the attacker as a legitimate server, however fake.

3.3.2 Loose synchronization

In Tsai et al.'s protocol, registration sub-phase employs the biometric smart card input without using any type of biometric capturing algorithm such as biohashing [49, 50] or fuzzy extractor [55]. Since, the Tsai et al.'s protocol calculates the parameter $Ei = Ki \oplus h(PWi \parallel bi)$ through utilizing the biometric input bi barely in the hash operation escaping any kind of pre-dealing algorithm or tool. This becomes the basis for the de-synchronization threat [56], since the stored biometric-template does not match with the current biometric-input; this is because the pre-dealing tools were not used. The use of such pre-dealing tools is crucial as we always observe significant amount of noise in the capturing of biometric inputs during registration and mutual authentication phase. In this manner we may avoid de-synchronization attacks.

3.3.3 Non-verification of smart card

In Tsai et al.'s protocol, the smart card is unable to authenticate the user in login phase before submitting the authentication request message to server. Due to this limitation, the service provider might come under Denial-of-Service (DoS) threat by inputting invalid password (PWi^*) and biometric input (Bi^*), in case the attacker steals over the smart card contents. Then, the smart card may construct Ki^* if the fake input parameters are given as input. The card then further constructs Ri^*, and finally produces $<L_2^*, M_1^*>$ message. Here, one point is worth mentioning that the service provider could refuse the authentication request after comparing equation (7), yet the server's energy gets drained for number of repeated computations. There might even be some legal users behaving maliciously to affect the server performance.

3.3.4 Other drawbacks in Tsai et al.'s protocol

In mutual authentication phase of Tsai et al.'s protocol, the smart card employs the private secret 'Ki' in various calculations without extracting it from the parameter Ei i.e., $K_i = Ei \oplus h(PWi \parallel bi)$.

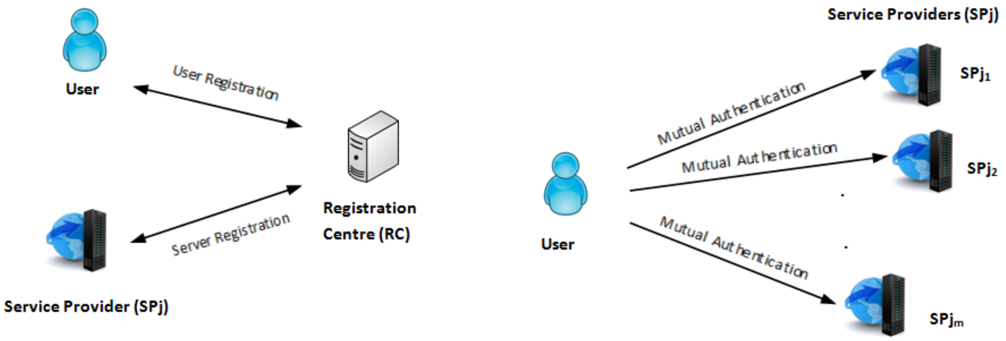
4 PROPOSED MODEL

We propose this enhanced and improved protocol after ascertaining few limitations in Tsai et al.'s scheme. In proposed model, likewise the user and service provider perform registration process prior to becoming part of the system. Thereafter, these entities could authenticate one another without engaging the registration centre as depicted in Figure 2. This contributed scheme is composed of three sub-sections, such as Registration, mutual authentication phase, and password modification sub-phase. The system set up of the proposed model takes the same assumptions as illustrated in section 3.1.

4.1 Registration Phase

All users and service providers are registered in this phase. For this purpose, the candidate users or servers submit their requests to registration centre. Once the RC receives the corresponding the request, it further constructs the private key for the respective user or server candidates, employing its own master secret key *s* as illustrated below.

$$\boldsymbol{K_i = \frac{1}{s+H_1(h(IDi))} P} \qquad \textbf{(18)}$$



a) Registration Phase

b) Authentication Phase

Figure 2. Proposed MSA protocol architecture eliminating RC from mutual authentication phase

The registration now submits K_i or K_j factors towards user or server on confidential channel. Once, these entities receive their respective private key from registration centre, the user computes $Ei = K_i \oplus h(PWi \,||\, H_b(bi))$. Then, the user stores the parameter *Ei* on smart card. In the calculation of *Ei*, *PWi* represents password, and *bi* be the biometric input from Ui. Similarly, the server upon getting the private secret key from registration centre, stores in its memory safely.

4.2 Login and Authentication procedure

The user performs the following steps when it wants to get mutually authenticated with server.

1. For mutual authentication, the user initially inputs its identity (***IDi)*** and password (***PWi***). Next, it imprints its biometric parameter (***Bi****) and also calculates $\boldsymbol{Di' = h(h(PWi \,||IDi) \,||\, H_b\,(bi))}$. Next, it compares the equality for $\boldsymbol{Di' \mathrel{?=} Di}$. If it fails to match, the protocol shall be aborted. On the other hand, the user submits the login query towards server SPj.
2. Onwards, the server constructs $\boldsymbol{Z = e(P, P)^u}$ and forwards to user.
3. The user, then calculates the parameters ***Kij***, $\boldsymbol{L_2}$, ***X***, $\boldsymbol{R_i}$ and $\boldsymbol{M_1}$ as given below:

$$\boldsymbol{K_i = Ei \oplus h(PWi \;||\, H_b(bi))} \qquad \textbf{(19)}$$

$$\boldsymbol{Kij = H_2\,(Z^{\,v}) = H_2\,(e(P,P)^{uv})} \qquad \textbf{(20)}$$

$$\boldsymbol{L_2 = vP_{pub} + H_1\,(h(ID_j))vP,} \qquad \textbf{(21)}$$

$$\boldsymbol{X = vP_{pub} + H_1\,(h(ID_i))vP,} \qquad \textbf{(22)}$$

$$\boldsymbol{R_i = \frac{1}{v+H_3\,(h(IDi)||L_2\,||Z\,||\,h(IDj)||\,X\,||\,Kij)} K_i} \qquad \textbf{(23)}$$

$$\boldsymbol{M_1 = Kij \oplus (h(ID_i)||\,R_i\,||\,X)} \qquad \textbf{(24)}$$

The user constructs the above factors and forwards <$\boldsymbol{L_2, M_1}$> to server, where *v* is a random integer. Even, ***v*** could be selected beforehand, likewise, other parameters $\boldsymbol{vP_{pub}}$, ***vP***, and $\boldsymbol{H_1(IDi)vP}$ may also be constructed before the mutual authentication phase, which helps to minimize the computational cost of the system protocol.

4. After getting the message $< L_2, M_1 >$ from user, the server calculates the session key Kij primarily, as following.

$$Kij = H_2 (e (L_2, K_j)^u) = H_2(e(P, P)^{uv}) \qquad (25)$$

Then server computes $(h(ID_i) || R_i || X)$ by calculating $(h(ID_i) || R_i || X) = Kij \oplus M_1$. The server, thereafter, calculates $e(R_i, X + H_3 (h(ID_i) || Z || h(ID_j) || X || Kij) Yi)$ and checks the equality against the already calculated parameter $e(P, P)$, for instance,

$$e(R_i, X + H_3 (h(ID_i) || L_2 || Z || h(ID_j) || X || Kij) Y_i) ?= e(P, P) \qquad (26)$$

Where the factor Y_i is calculated as $Y_i = (P_{pub} + H_1(h(ID_i))P)$. Here, the server authenticates the subscriber positively as shown in Eq (**26**). Then, server calculates L_3 and Q_i as shown in the Eq **(27)** and **(28)**. Next, it submits the content $< L_3, Q_i >$ towards user, in this manner the service provider could also be authenticated by Ui.

$$L_3 = uP_{pub} + H_1 (h(ID_i))uP \qquad (27)$$

$$Q_i = H_4(Kij || Z || L_3 || h(ID_i) || h(ID_j)) \qquad (28)$$

5. Then, the user gets $< L_3, Q_i >$ and computes Kij' and verifies Q_i with the computation as shown in Eq **(30)**

$$Kij' = H_2(e (L_3, K_i)^v) \qquad (29)$$

$$Q_i ?= H_4(Kij' || Z || L_3 || h(ID_i) || h(ID_j)) \qquad (30)$$

If the Eq **(30)** holds valid, the subscriber authenticates the service provider. Alternatively, it will terminate the session. Therefore, both of the participating entities authenticate one another and construct the agreed session key, i.e. $SK = Kij$.

4.3 Password Alteration Procedure

The subscriber may transform its old password (PWi) into a new one as (PWi^{new}) after initiating the under-mentioned steps. The former may alter PWi without seeking any help from registration centre. The corresponding steps for this modification are given as under:

1) The user initially enters its IDi and PWi as input in smart card. Then, it imprints its biometric impression bi in biometric scanner and chooses to modify its password.
2) After that, the smart card calculates $Di^* = h(h(PWi || IDi) || H_b (bi))$ and confirms the equation for $Di^* ?= Di$. In case, it is not valid the smart would decline the password modification request. On the other hand, it shall allow the user to progress with the password updation procedure.
3) Next, the smart card calculates the parameter $K_i = Ei \oplus h(PWi || H_b(bi))$ and asks the user to enter its new password (PWi^{new}).
4) Then, it calculates the values $Di^{new} = h(h(PWi^{new} || IDi) || H_b (bi))$ and $Ei^{new} = Ki \oplus h(PWi^{new} || H_b (bi))$.
5) Ultimately, the card shall replace the parameters Di, Ei against the modified Di^{new} and Ei^{new} parameters.

5. SECURITY ANALYSIS

The security analysis for the proposed model has been described below:

5.1 Resists Replay Attack

The replay attacks could be instigated while an adversary reproduces the same content in various timings to betray any valid member entity. The attacker adversary Ă intercepts publicly available messages $< Z >$, <

$L_2, M_1>$, $< L_3, Q_i>$ and might attempt to replay those contents to any one of the legitimate members. If an attacker replays either $< Z>$ or $< L_3, Q_i >$ contents towards user then the latter may recognize in 4th step of login and authentication procedure after matching the equation $Q_i \; ?= H_4(Kij' \,\|\, Z \,\|\, L_3 \,\|\, h(ID_i) \| \, h(ID_j))$ as depicted in ***Eq*** **(30).** If the equation does not match, Ui would consider this message as a replayed message or attack. Similarly, on replaying the content $<L_2, M_1>$, the server checks the validity after matching the equation **(26)**, i.e. $e(R_i, X + H_3\,(h(ID_i) \| \, L_2 \,\|\, Z \,\|\, h(ID_j) \| \, X \,\|\, Kij)\; Y_i) \; ?= e(P, P)$**.** In case, this equality fails to match, this will be taken as replay threat from server. Thus, our protocol could comfortably counter this attack.

5.2 Resists Modification

The modification or man-in-the-middle attack may be initiated if the adversary alters the message details, being unauthorized, for presenting it to some legal server or user so that those legitimate members misleadingly take them as original participants. If an adversary attempts to alter the contents $< Z>$, $< L_2$, $M_1>$ or $< L_3$, $Q_i>$, then user or server might comfortably prevent this threat after matching the equation $e(R_i, X + H_3\,(h(ID_i) \| \, L_2 \,\|\, Z \,\|\, h(ID_j) \| \, X \,\|\, Kij)\; Y_i) \; ?= e(P, P)$ and $Q_i \; ?= H_4\,(Kij' \,\|\, Z \,\|\, L_3 \,\|\, h(ID_i) \| \, h(ID_j))$ for SPj and Ui, respectively, as depicted in equations **(26)** and **(30).** In this context, we might infer that our protocol is protected from MiTM attack from both sides.

5.3 Resists offline-Password Guessing Attack

An attacker Ă might attempt to guess the password (***PWi***) out of the intercepted or stolen contents of smart card [45]. The smart card comprises $Di=h(h(PWi \,\|IDi) \,\|H_b\,(bi))$ and $Ei = K_i \oplus h(PWi \,\|\, H_b\,(bi))$ factors, nonetheless, Ă still cannot extract or compute the password out of ***Ei*** or ***Di***, as the attacker has no knowledge about a high entropy random secret ***bi***. Therefore, if we concatenate the identity and ***bi***, the password could not be calculated in polynomial amount of time by the attacker. In this context, our scheme is protected from any sort of password or identity guessing attack.

5.4 Session key security

The adversary Ă could steal the contents of smart card or it may intercept the public messages. Then, it may use these contents to calculate the legitimate session key as $SK =\{Kij\}= \{Kij'\}$ from the available parameters. Nonetheless, Ă may not be able to calculate ***SK*** as the computation needs the availability of x and y factors, while the computation of those factors by the attacker is bound by the hardness of ECDLP problem [15]. Therefore, the contents stolen by the adversary on an insecure channel could not expose the valid session keys constructed between the legal participants.

5.5 Resists Impersonation / Server spoofing attack

An attacker Ă, being a malicious server, might attempt to originate a spoofing attack against user. Nonetheless, unlike the scheme Tsai et al., if Ă forwards a fabricated factor ***Z*** i.e., $Z = e(\, P_{pub}+ H_1(h(ID_j))P,\, P)^u$ to Ui, the later could comfortably discern the possibility of threat in 4th step of login and authentication phase of our scheme, after matching the equation **(30)**. If Ă would attempt to initiate this attack, the adversary would not be able to pass the equality for $Q_i \; ?= H_4\,(Kij' \,\|\, Z \,\|L_3\| \, h(ID_i) \| \, h(ID_j))$, and then Ui will terminate the session. Therefore, in our scheme the entities validate the authenticity of one another while the adversary may not be able to initiate any kind of server impersonation attack.

5.6 Known-Key Security

This feature warrants the incapability of the adversary of guessing past session keys if the current session key is exposed. In our scheme, even if the attacker becomes familiar about the session key $SK= \{Kij\}$ of any session, it might not assist the attacker in determining the rest of the session keys among the similar members, since a unique random integer involves in every session establishment phase between the same members. For this reason, it becomes an intractable problem for the adversary to approach the related secret factors, as nearly as ECDLP-based hard problem. Consequently, our protocol supports the feature of known-key security.

5.7 Perfect Forward secrecy

This feature warrants the secrecy of session keys exposure once the attacker is able to approach the private secret keys of control authority, for example, the registration centre in our scenario.
In our protocol, in case the adversary approaches the private secret key (***s***) of RC, it may calculate the private secrets of the involved Ui and SPj, i.e. K_i and K_j, after getting access to identities (***IDi***, ***IDj***) of user or server ***IDi***, ***IDj*** also depicted in equation **(31)** and **(32)**.

$$K_i = \frac{1}{s + H_1(h(IDi))} P \qquad \textbf{(31)}$$

$$K_j = \frac{1}{s + H_1(h(IDj))} P \qquad \textbf{(32)}$$

However, it may not compute the session key ***SK = {Kij}*** as the calculation for ***Kij*** factors, regardless of the information about K_i and K_j, needs additionally the knowledge about the parameters of a specific session, such as ***x*** or ***y*** to calculate the session key ***Kij***. In addition, the attacker might not be able to recover those factors out of M_1 i.e., $M_1 = Kij \oplus (h(ID_i) \| R_i \| X)$ due to ignorance of information about R_i and ***X*** factors**.** Thus, our protocol supports the feature of perfect forward secrecy.

5.8 Mutual Authentication

This property suggests that both of the interacting participants verify and validate one another within the same protocol. In our protocol, the server authenticates the user after receiving the request <***A***> submitted by the user and the challenge response as received from user. The server calculates bilinear map and verifies it with the computer parameter ***e(P, P)*** as depicted in equation **(26)**, in this manner it validates the user. Likewise, Ui verifies the server after calculating bilinear map and matching the parameter Q_i with $H_4(Kij' \| Z \| h(ID_i) \| h(ID_j))$ as depicted in equation **(30)**. Following this, the user and server may authenticate one another in the protocol.

5.9 Anonymous Authentication

This feature affords anonymity to user besides getting it authenticated out of service provider. After a successful anonymous authentication the adversary may not infer about the identity of any of the participating entities by using the intercepted parameters. In our scheme, the adversary might not be able to recover the Ui's identity out of the publicly available messages of various sessions, as the identity (***IDi***) is included in the construction of M_1, i.e., $M_1 = Kij \oplus (h(ID_i) \| R_i \| X)$, besides it is embedded after taking hash in M_1 parameter. This is least probable to recover and guess those secrets, i.e. ***u, v*** and ultimately the session key is computed or approached in polynomial amount of time. As a result, the demonstrated scheme affords sufficient authenticity and anonymity to the user.

5.10 Resist de-synchronisation threat

This attack might be possible by the attacker if the latter alters the contents in a manner that the valid members could not authenticate each other, and then they have to terminate the session. This may lead to desynchronization attack while the adversary changes the contents such that the legitimate members may not verify one another, and will have to terminate the session in mutual authentication. In our scheme, if attacker attempts to alter the content < Z>, <L_2, M_1>, <L_3, Q_i>, the Ui could counter this threat after calculating $Kij' = H_2(e(L_3, K_i)^v)$ and checking the equation $Q_i \ ?= H_4(Kij' \,||\, Z \,||\, L_3 \,||\, h(ID_i) \,||\, h(ID_j))$ as revealed in equation **(29).** This is because of the fact that the calculated factor Kij' does not match Qi against $H_4(Kij' \,||\, Z \,||\, L_3 \,||\, h(ID_i \,||\, h(ID_j)$, which prevents the attackers to launch the modification attack. At the same time, the De-synchronization attack might happen on the mismatch of stored biometric template with the same biometric imprint in the login process. In proposed scheme, we employed bio-hashing to remove the probability of mismatch out of noise in direct biometric application without any pre-dealing tool. Hence, the de-synchronization attack, in both ways, may be detected and foiled successfully in proposed scheme.

6 FORMAL SECURITY ANALYSIS

We demonstrate the security analysis formally in this section by employing the Burrows-Abadi-Needham logic (BAN) logic [36, 37] technique and also random oracle model. The first model analyzes the protocol on account of few parameters including key distribution, mutual authentication, and immunity strength for session key exposure. Next, we illustrate few notations that are utilized in proving the protocol using BAN logic.

f |≡ ʐ: ***f*** believes the message ɱ.

f ⊲ ɱ: ***f*** sees the message ɱ.

f |~ ɱ: ***f*** once said ɱ.

f ⇒ ɱ: ***f*** has jurisdiction over ɱ.

♯ **(**ɱ**)**: ɱ is fresh.

$\langle ɱ \rangle_{ɱ'}$: ɱ is combined with another ɱ'.

(ɱ**,** ɱ'**)**: ɱ or ɱ' are parts of a message (ɱ, ɱ').

{ ɱ**,** ɱ'**}**$_K$: ɱ or ɱ' is encrypted using the key K.

$f \xleftrightarrow{K} f'$: ***f*** and ***f'*** communicate utilizing a shared key K.

(ɱ**,** ɱ'**)**$_K$: ɱ or ɱ' is hashed by key K.

Some rules that are used in the proof of BAN logic are given as following:

Rule 1. Message-meaning rule: $\frac{ʃ \,|\!\equiv ʃ \overset{K}{\leftrightarrow} ʃ',\ ʃ \lhd \langle ɱ \rangle_{ɱ'}}{ʃ \,|\!\equiv ʃ' \,|\!\sim ɱ}$

Rule 2. Nonce-verification rule: $\frac{ʃ \,|\!\equiv \sharp(ɱ),\ ʃ \,|\!\equiv ʃ' \,|\!\sim ɱ}{ʃ \,|\!\equiv ʃ' \,|\!\equiv ɱ}$

Rule 3. Jurisdiction rule: $\frac{ʃ|\!\equiv ʃ' \Rightarrow ɱ,\ ʃ|\!\equiv ʃ' \,|\!\equiv ɱ}{ʃ|\!\equiv ɱ}$

Rule 4. Freshness-conjuncatenation rule: $\frac{ʃ|\!\equiv \sharp(ɱ)}{ʃ|\!\equiv \sharp(ɱ,\ ɱ')}$

Rule 5. Belief rule: $\frac{ʃ \,|\!\equiv (ɱ),\ ʃ|\!\equiv (ɱ')}{ʃ \,|\!\equiv (ɱ,\ ɱ')}$

Rule 6. Session-keys rule: $\frac{ʃ|\!\equiv \sharp(ɱ),\ ʃ|\!\equiv ʃ' \,|\!\equiv ɱ}{ʃ|\!\equiv ʃ \overset{K}{\leftrightarrow} ʃ'}$

The contributed scheme requires satisfying the under-mentioned goals (G1-G4) for ensuring the security using BAN logic, employing the postulates as mentioned above.

G1 : $S \,|\!\equiv Ui \xleftrightarrow{SK} S$

G2 : $S \,|\!\equiv Ui \,|\!\equiv Ui \xleftrightarrow{SK} S$

G3 : $Ui \,|\!\equiv Ui \xleftrightarrow{SK} S$

G4 : $Ui \,|\!\equiv S \,|\!\equiv Ui \xleftrightarrow{SK} S$

We may transform the communicated messages in our scheme into idealized form as given below:

m_1: Ui → S: $L_2, M_1: \langle IDi, R_i, vP_{pub} + H_1\ (h(IDi)).vP \rangle_{Kij}$

m_2: S → Ui: $L_3, Q_i: \langle IDj,\ uP_{pub} + H_1\ (h(IDi)).uP \rangle_{Kij'}$

Secondly, the following premises have been established to prove the security of our scheme.

P1 : $Ui \mid\equiv \sharp v$

P2 : $S \mid\equiv \sharp (u, Z)$

P3 : $Ui \mid\equiv S \xleftrightarrow{Kij} Ui$

P4 : $S \mid\equiv S \xleftrightarrow{Kij'} Ui$

P5 : $Ui \mid\equiv S \Rightarrow (uP_{pub} + H_1\ (h(IDi)).uP)$

P6 : $S \mid\equiv Ui \Rightarrow (R_i, vP_{pub} + H_1\ (h(IDj)).vP)$

Thirdly, the mentioned idealized form such as m_1 and m_2 in the contributed scheme may be evaluated and verified with the help of postulates as illustrated above.

By using the above mentioned symbols, postulates, assumptions and idealized forms, we come to the understated derivations:

Regarding the first idealized form, we have:

m_1: Ui → S: $L_2, M_1: \langle IDi, R_i, vP_{pub} + H_1\ (h(IDj)).vP \rangle_{Kij}$

On the basis of the seeing rule, we have the following derivation

D1: $S \lhd L_2, M_1: \left\langle IDi, R_i, vP_{pub} + H_1\ (h(IDj)).vP \right\rangle_{Kij}$

According to D1, P3 and message-meaning-rule,

D2: $S \mid\equiv Ui \sim (R_i, vP_{pub} + H_1\ (h(IDj)).vP)$

In relation to P2, D2, Rule4, and Rule2, we have

D3: $S \mid\equiv Ui \mid\equiv (R_i, vP_{pub} + H_1\ (h(IDj)).vP)$

Here, $(IDi, R_i, vP_{pub}+H_1\ (h(IDj)).vP)$ are few significant factors required to mutually authenticate the participants and calculating the session key $SK = \{Kij\}$.

In relation to P6, D3, and Rule3

D4: $S \mid\equiv (R_i, vP_{pub} + H_1\ (h(IDj)).vP)$

In relation to P3, D4, and Rule 6, we have

D5: $S \mid\equiv Ui \mid\equiv S \xleftrightarrow{SK} Ui$ **(G2)**

In relation to P6, D5, and Rule 3

D6: $S \mid\equiv Ui \xleftrightarrow{SK} S$ **(G1)**

Regarding the 2nd message of constructed idealized form, we have:

m_2: S → Ui: $L_3, Q_i: \langle IDj,\ uP_{pub} + H_1\ (h(IDi)).uP \rangle_{Kij'}$

On the application of Rule the seeing rule, we have

D7: $Ui \lhd S \rightarrow Ui: L_3, Q_i: \langle IDj, uP_{pub} + H_1(h(IDi)).uP \rangle_{Kij'}$

In relation to D7, P4 and Rule 1,

D8: $Ui \mid\equiv S \sim (uP_{pub} + H_1\ (h(IDi)).uP)$

In relation to D8, P1, Rule 4, and Rule 2, we have,

D9: $Ui \mid\equiv S \mid\equiv (uP_{pub} + H_1\ (h(IDi)).uP)$

Here, ($uP_{pub} + H_1\ (h(IDi)).uP$) are significant factors utilized in authenticating the participants and verifying the computer factor Kij' that is utilized in the computation of session key $SK = \{Kij\} = \{ Kij'\}$.

In relation to P5, D9, and Rule 3

D10: Ui $|\equiv$ ($uP_{pub} + H_1\ (h(IDi)).uP$)

In relation to P4, D10, and Rule 6, we have

D11: Ui $|\equiv$ S $|\equiv$ Ui $\xleftrightarrow{SK}$ S **(G4)**

In accordance with P5, D11, and the Jurisdiction rule

D12: Ui $|\equiv$ Ui $\xleftrightarrow{SK}$ S **(G3)**

We prove formally by using BAN logic analysis that our contributed scheme may attain the property of mutual authentication, while the constructed session key (SK) is established on mutual basis between user and server.

Besides the above proof, we might use a random oracle model (ROM) which is generally known as a generic contradiction model in cryptography [48], to prove that the existing scheme is secure enough to construct a mutually authenticated session key. To prove the protocol by using the above defined model, we employed two oracles such as *Reveal1* and *Reveal2* as given below:

$Reveal_1$: The oracle $Reveal_1$ will generate a out of the related bilinear map $Z = e(P,P)^a$ in absolute terms.
$Reveal_2$: The oracle $Reveal_2$ produces t out of related hash value $u=h(t)$, absolutely.

The $Reveal_1$ oracle is employed for the Algorithm 1. $EXP1^{Key}_{IMSADMCCS}$ specifying the exposure of session key SK if the oracle $Reveal_1$ is utilized by taking the inverse hash function.

Algorithm 1. $\boldsymbol{EXP1^{Key}_{\mathrm{BMSAMCC}}}$

```
1.  Eavesdrop login request, i.e. $<Z>$ in mutual authentication phase, while $Z = e(P, P)^u$ .
2.  Call $Reveal_1$ oracle on the input of $Z = e(P, P)^u$ to get $u' \leftarrow reveal1\ (e(P, P)^u)$.
3.  Eavesdrop $<L_2, M_1>$ and $<L_3, Q_i>$ in mutual authentication phase, where $L_2 = vP_{pub} + H_1\ (h(IDj))vP$, $L_3 = uP_{pub} + H_1\ (h(IDi))uP$, $M_1 = Kij \oplus (h(IDi) \| R_i \| X)$ and $Q_i = H_4(Kij \| Z \| h(IDi) \| h(IDj))$.
4.  Call $Reveal_2$ on the input of factor $Z_i$ to get $(Kij^*, Z', h(IDi'), h(IDj))$ as $(Kij \| Z' \| h(IDi) \| h(IDj)) \leftarrow reveal_1(Q_i)$.
5.  Onwards, it calculates $Kij^* \oplus M_1$ and then recovers $(h(IDi'') \| R_i' \| X')$.
6.  Afterwards, it calculates $Y_i' = (P_{pub} + H_1\ (h(IDi))P)$.
7.  If [$(h(IDi') == h(IDi''))$ AND $e(R_i', X' + H_3(h(IDi') \| L_2 \| Z' \| h(IDj') \| X' \| Kij^*)Y_i') == e(P,P)$]
      Accept the identity $IDi'$ as valid for Ui, and also accept $SK = Kij^*$ as a legitimate session key for user and server,
      Return 1 (true)
8.      Else
9.    Return 0 (false)
10. End if
```

Theorem1

The contributed protocol would be secure, if a crafty adversary attempts to extract the corresponding session key among the legal participants, given the hash digest function h() acts strongly as a random oracle.

Proof.

In this proof, the adversary Ă who is competent enough to recover the session key among the legal participants, employs the oracles *Reveal1* and *Reveal2* to implement the algorithm $EXP1^{Key}_{\mathbf{BMSAMCC}}$. The probability for success of the algorithm $EXP1^{Key}_{\mathbf{BMSAMCC}}$ is Suc1=Pr.2[$EXP1^{Key}_{\mathbf{BMSAMCC}}$=1] - 1, whereas, Pr [E] indicates the expected probability for the relevant event (E). Here, the advantage function of $EXP1^{Key}_{\mathbf{BMSAMCC}}$ is calculated as $Adv^{Key}_{\mathbf{BMSAMCC}}$ (t_1, q_1, q_2) = $\max_{Ă}$ [$Suc1^{Key}_{\mathbf{BMSAMCC}}$], in execution time ($t_1$) as well as the Reveal query q_1 and q_2 maximized on adversary Ă. We may term the contributed protocol as verifiably secure against the adversary to extract the agreed session key SK among the participants, provided the advantage $Adv^{Key}_{\mathbf{BMSAMCC}}$($t_1$, q_1, q_2) $\leq \varepsilon'$ for any adequately small $\varepsilon' > 0$. In accordance with the above experiment, in case the adversary Ă is capable of approaching private secrets of involved participants, and accessing the bilinear map components, it could comfortably extract the original session key SK and ultimately Ă wins this game. Nonetheless, referring to [15], it becomes infeasible as far as computation is concerned, for breaking the corresponding bilinear map as $Adv^{Key}_{\mathbf{BMSAMCC}}$ (t_1) $\leq \varepsilon'$ for adequately small $\varepsilon' > 0$. Thus, our contributed protocol might safely be deemed to be secure since the related security features for the hash operation are polynomially too hard to break.

7. PERFORMANCE EVALUATION ANALYSIS

In this section, we present the comparative analysis of the contributed scheme with Tsai et al. and few contemporary schemes. The Table II shows the functionality comparison for different protocols [24], [38], [40]-[43], [44], [34] and our protocol. The proposed scheme is an extended and improved model for Tsai et al.'s protocol. Our contributed model is proven to be a robust authenticated key agreement protocol in the formal security analysis as demonstrated above. We can witness that the three of those schemes abide by the anonymity requirement, i.e. [34], [44] and our scheme. Likewise, majority of the schemes do not fulfill the traceability feature excluding [34] and our scheme. We may notice the time-synchronization problem, stolen smart card threat, and modification attack in [38], and [40]-[42] as discovered in later schemes. While, [38] and [40] do not resist replay attack, and the scheme [42] is found to be vulnerable to password-guessing attack. In earlier schemes, mostly the protocols were found susceptible to impersonation attacks except [24]. In the last, the protocols that enables the mobile user authentication setting to client employing identity based cryptography are [24, 34], including our protocol.

Table II: Comparison of security features for different ID-based cryptographic protocols

	[24]	**[38]**	**[40]**	**[41]**	**[42]**	**[44]**	**[34]**	**Ours**
Supports Anonymity	×	×	×	×	×	√	√	√
Mutual Authentication	√	×	×	×	√	×	×	√
Known key secrecy	√	×	√	×	√	√	√	√
Untraceable	√	×	×	×	×	×	√	√
Defy Modification threat	√	×	×	×	×	√	√	√
Defy offline-password guessing threat	√	√	√	√	×	√	√	√
Defy Stolen card threat	√	×	×	×	×	√	√	√
Defy Impersonation threat	√	×	×	×	×	×	×	√
Defy Replay threat	√	×	×	√	√	√	√	√
Multi-server Paradigm	√	×	×	×	×	×	√	√
Resistance to Time Synchronization issues	√	×	×	×	×	√	√	√
De-synchronization attack	√	√	√	√	√	√	×	√

Smart card-based user verification	√	√	√	√	√	√	×	√

√: Could resist threat ×: Could not resist threat

Table III. Number of operations in Tsai and Proposed protocol

	Tsai et al. protocol [34]	**Proposed protocol**
Registration messages	1 T_{PM}	1 T_{PM}
User	4 T_{PM}	4 T_{PM} + 1T_{BP}
Service provider	2 T_{PM} + 3T_{BP}	3 T_{PM} + 2T_{BP}
Computational delay (ms)	6 T_{PM} + 3T_{BP} $\approx$ 30.75	7 T_{PM} + 3T_{BP} $\approx$ 32.98

The comparative analysis for the computational costs of Tsai et al.'s and the contributed protocol is depicted in Table III. We define the symbol T_{BP} as the time needed to complete for the bilinear pairing-based operation, and the symbol T_{PM} be the time required to complete scalar point multiplication operation. We assume that some computations on the end of user are calculated already, so these are excluded from computational cost in the comparison such as $yH_1(IDi)P$, yP, $yPub$, as shown in Table III. We calculated the computational costs by simulating and employing the MIRACL library [67] on a desktop computer (HP E8300 Core i5 with 2.93 Ghz processor using Ubuntu 16.10 OS having 4GB RAM), while the time latency for T_{PM} and T_{BP} are calculated as 2.214*ms* and 5.79*ms*, respectively. The registration procedure for Tsai et al. and proposed scheme takes 1T_{PM} of time latency to register the user and server in constructing their respective private keys. In mutual authentication step, the user consumes the total time as 4T_{PM} in Tsai et al. scheme, whereas in our contributed scheme, it consumes 4 T_{PM} + 1T_{BP} in mutual authentication phase. The server consumes 2 T_{PM} + 3T_{BP} overall time delay for Tsai et al., while in our proposed scheme, it would take 3T_{PM} + 2T_{BP} time latency. Although the contributed protocol consumes an extra operation of 1T_{BP} on Ui's side, and 1T_{PM} on the server's side, nonetheless, the contributed protocol is not susceptible to server spoofing or impersonation threat, as Tsai et al.'s protocol stands vulnerable to the same attack. The timing for computational cost is calculated for Tsai et al. and our scheme is as 30.75 and 32.98, respectively. The cost of the contributed protocol is almost 8% higher than Tsai et al.'s protocol owing to extra point multiplications as shown in Table 3 and Figure 3, nonetheless, the former is secure against possible impersonation threats. The Figure 3 depicts the graph that demonstrates that although our scheme's computational cost is bit high, yet it is more secure. In contributed protocol, the operation for bilinear map is dominant in identity-based cryptography, and lets the server and user in verifying one another's authenticity to establish multiple mutual sessions without involving the registration centre.

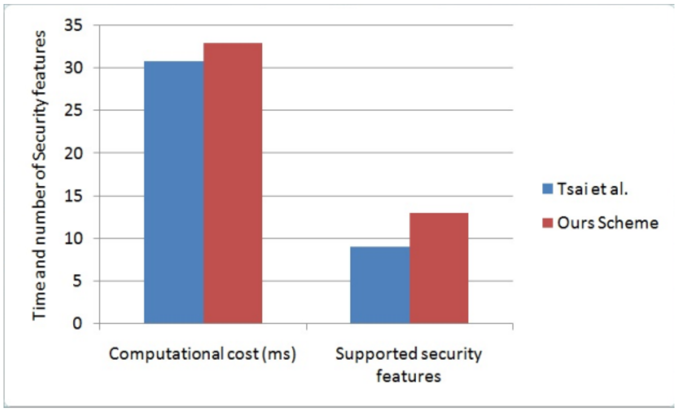


Fig. 3. Comparison graph for Tsai et al. and our scheme

In view of the fact that, our contributed protocol is immune of impersonation threats that Tsai et al. could not embed in its protocol, therefore, in view of the illustrated performance evaluation analysis, we may deduce that the contributed model is far more secure than Tsai et al.'s protocol although it incurs a little more necessary additional cost. This is also mention-worthy that the security or immunity from attacks for any key agreement protocol is more significant element for practical implementation, while for enhancing the security, to a certain extent an additional overhead may be afforded.

8. CONCLUSION

The mobile cloud computing (MCC) is increasingly finding ways for being embedded in the mobile subscriber-based services. Lately, Tsai et al. presented a new mobile user authentication scheme employing bilinear pairing operations, to abandon the involvement of trusted third party in the authentication process between the participants. Nonetheless, the Tsai et al.'s protocol is found to be vulnerable to server-based impersonation and desynchronization attacks. Besides, it also lacks smart card-based user verification in the login phase that makes the protocol inapplicable for implementation in access networks. In this context, we have put forward an enhanced and secure authentication scheme based on the operation of bilinear pairing, enabling to counter many attacks as discovered in Tsai et al. scheme. The contributed protocol, in this study, demonstrates the security analysis or formal and informal basis which warrants that the contributed protocol is immune of particular known attacks and limitations as faced by earlier protocols.

REFERENCES

[1] G. Le, K. Xu, M. Song, and J. Song (2011). “A survey on research on mobile cloud computing”, in Proc. 10th IEEE/ACIS/Int. Conf. Comput. Inf. Sci., pp. 387–392.

[2] X. F. Qiu, J.W. Liu, and P. C. Zhao, “Secure cloud computing architecture on mobile Internet,” in Proc. 2nd Int. Conf. AIMSEC, 2011, pp. 619–622.

[3] N. Fernando, S. W. Loke, and W. Rahayu, “Mobile cloud computing: A survey,” Future Gen. Comput. Sys., vol. 29, no. 1, pp. 84–106, Jan. 2013.

[4] ABI Research Report, Mobile Cloud Applications. [Online]. Available: http://www.abiresearch.com/research/1003385-Mobile+Cloud+Computing

[5] W. G. Song and X. L. Su, “Review of mobile cloud computing,” in Proc. IEEE 3rd ICCSN, 2011, pp. 1–4.

[6] W. Itani, A. Kayssi, and A. Chehab, “Privacy as a service: Privacy-aware data storage and processing in cloud computing architectures,” in Proc. IEEE Int. Conf. Dependable Auton. Secure Comput., 2009, pp. 711–716.

[7] S. Pearson, “Taking account of privacy when designing cloud computing services,” in Proc. CLOUD ICSEWorkshop Softw. Eng. Challenges Cloud Comput., 2009, pp. 44–52.

[8] H. Takabi, J. B. D. Joshi, and G. Ahn, “Security and privacy challenges in cloud computing environments,” IEEE Security Privacy, vol. 8, no. 6, pp. 24–31, Nov./Dec. 2010.

[9] Z. Xiao and Y. Xiao, “Security and privacy in cloud computing,” IEEE Commun. Surveys Tuts., vol. 15, no. 2, pp. 843–859, Jul. 2012.

[10] H. Ahn, H. Chang, C. Jang, and E. Choi, “User authentication platform using provisioning in cloud computing environment,” in Proc. ACN CCIS, 2011, vol. 199, pp. 132–138.

[11] P. Urien, E. Marie, and C. Kiennert, “An innovative solution for cloud computing authentication: Grids of EAP-TLS smart cards,” in Proc. 5th Int. Conf. Digit. Telecommun., 2010, pp. 22–27.

[12] J. L. Tsai, N. W. Lo, and T. C. Wu, “Secure delegation-based authentication protocol for wireless roaming service,” IEEE Commun. Lett., vol. 16, no. 7, pp. 1100–1102, Jul. 2012.

[13] H. Chang and E. Choi, “User authentication in cloud computing,” in Proc. UCMA CCIS, 2011, vol. 151, pp. 338–342.

[14] OpenID Foundation, OpenID Authentication 2.0, 2007. [Online]. Available: http://openid.net/specs/openid-authentication-2_0.html

[15] N. Koblitz, “Elliptic curve cryptosystems,” Math. Comput., vol. 48, no. 177, pp. 203–209, 1987.

[16] V. Miller, “Use of elliptic curves in cryptography,” in Proc. CRYPTO, 1986, pp. 417–426.

[17] “Recommendation for key management—Part 1: General,” Gaithersburg, MD, USA, Aug. 2005, Special Publication 800-57.

[18] D. Boneh and M. Franklin, “Identity-based encryption from the Weil pairing,” in Advances in Cryptology-CRYPTO, vol. 2139, LNCS. Berlin, Germany: Springer-Verlag, 2001, pp. 213–229.

[19] J. C. Cha and J. H. Cheon, “An identity-based signature from gap Diffie–Hellman groups,” in Public Key Cryptography PKC, vol. 2139, LNCS. Berlin, Germany: Springer-Verlag, 2003, pp. 18–30.

[20] H. Z. Du and Q. Y. Wen, “An efficient identity-based short signature scheme from bilinear pairings,” in Proc. Int. Conf. CIS, 2007, pp. 725–729.

[21] H. W. Lim and M. Robshaw, “On identity-based cryptography and grid computing,” in Proc. ICCS, 2004, pp. 474–477.

[22] H.W. Lim and M. Robshaw, “A dynamic key infrastructure for GRID,” in Proc. EGC, 2005, pp. 255–264.

[23] W. Mao, “An identity-based non-interactive authentication framework for computational grids,” HP Labs, Palo Alto, CA, USA, Tech. Rep. HPL-2004-96, Jun. 2004.

[24] H. Li, Y. Dai, L. Tian, and H. Yang, “Identity-based authentication for cloud computing,” in Proc. CloudCom, 2009, pp. 157–166.

[25] V. S. Hughes, “Information hiding, anonymity and privacy a modular approach,” J. Comput. Security, vol. 12, no. 1, pp. 3–36, Jan. 2004.

[26] OASIS, SAML version 2.0 errata 05, May 2012. [Online]. Available: http://docs.oasis-open.org/security/saml/v2.0/sstc-saml-approved-errata-2.0.pdf

[27] OpenID Foundation, The OpenID User Interface Extension 1.0, Draft0.5, 2009. [Online]. Available: http://svn.openid.net/repos/specifications/user_interface/1.0/trunk/openid-user-interface-extension-1_0.html

[28] OpenID Foundation, OpenID Specifications, 2007.[Online].Available: http://openid.net/developers/specs/

[29] OpenID Foundation, The OpenID User Interface Extension Best Practices for Identity Providers 2009. [Online]. Available: http://wiki.openid.net/w/page/12995153/Details-of-UX-Best-Practices-for-OPs

[30] Google, Google Security and Product Safety, 2009.[Online].Available: http://www.google.com/about/company/security.html

[31] Google, SAML Single Sign-On (SSO) Service for Google Apps, 2008. [Online]. Available: https://developers.google.com/google-apps/sso/saml_reference_implementation?hl=zh-tw

[32] A. Armando et al., “An authentication flaw in browser-based single sign-on protocols: Impact and remediations,” Comput. Security, vol. 33, pp. 41–58, Mar. 2013.

[33] Microsoft, Windows Live ID, 2011. [Online]. Available: https://account.live.com/

[34] J. L. Tsai and N. W. Lo, “A privacy-aware authentication scheme for distributed mobile cloud computing services,” IEEE Syst. J. ,vol.9, no. 3, pp. 805–815, Sep. 2015.

[35] J. L. Tsai, N. W. Lo, and T. C. Wu, “Novel anonymous authentication scheme using smart cards,” IEEE Trans. Ind. Informat., vol. 9, no. 4, pp. 2004–2013, Nov. 2013.

[36] M. Burrows, Abadi, M., & Needham, R. (1990). A logic of authentication. ACM Transactions on Computer Systems, 8(1), 18–36. doi:10.1145/77648.77649.

[37] M. Burrows, Abadi, M., & Needham, R. M. (1871). A logic of authentication. Proceedings of the Royal Society of London A-Mathematical and Physical Sciences, 1989(426), 233–271.

[38] M. L. Das, A. Saxena, V. P. Gulati, and D. B. Phafstak, “A novel remote user authentication scheme using bilinear pairings,” Comput. Security, vol. 25, no. 3, pp. 184–189, May 2006.

[39] J. S. Chou, Y. Chen, and J. Y. Lin, “Improvement of Das et al.’s remote user authentication scheme,” Cryptology ePrint Archive, 2005. [Online]. Available: http://eprint.iacr.org/2005/450.pdf

[40] T. Goriparthia, M. L. Das, and A. Saxena, “An improved bilinear pairing based remote user authentication scheme,” Comput. Std. Interfaces, vol. 31, no. 1, pp. 181–185, Jan. 2009.

[41] A. S. Khan Pathan, C. S. Hong, and K. Hee, “Bilinear-pairing-based remote user authentication schemes using smart cards,” in Proc. 3rd Int. Conf. Ubiquitous Inf. Manage. Commun., 2009, pp. 356–361.

[42] T. H. Chen, H. L. Yeh, and W. K. Shih, “An advanced ECC dynamic ID based remote mutual authentication scheme for cloud computing,” in Proc. 5th FTRA Int. Confe. Multimedia Ubiquitous Eng., 2011, pp. 155–159.

[43] D. Wang, Y. Mei, C. G. Ma, and Z. S. Cui, “Comments on an advanced dynamic ID-based authentication scheme for cloud computing,” in Web Information Systems and Mining, vol. 752, LNCS. Berlin, Germany: Springer-Verlag, 2012, pp. 246–253.

[44] H. Sun, Q.Wen, H. Zhang, and Z. Jin, “A novel remote user authentication and key agreement scheme for mobile client–server environment,” Appl. Math. Inf. Sci., vol. 7, no. 4, pp. 1365–1374, 2013.

[45] M. Bellare, D. Pointcheval, and P. Rogaway, “Authenticated key agreement secure against dictionary attacks,” in Proc. EUROCRYPT, 2000, pp. 139–155.

[46] M. Jakpbsson and D. Pointcheval, “Mutual authentication for low-power mobile devices,” in Proc. FC, Feb. 19–22, 2001, pp. 178–195.

[47] F. Bao, R. H. Deng, and H. Zhu, “Variations of Diffie–Hellman problem,” in Proc. 5th ICICS, 2003, pp. 301–312.

[48] Stallings W. Cryptography and network security: principles and practices, third ed. Prentice Hall: Peason, 2003.

[49] Jin, A. T. B., Ling, D. N. C., & Goh, A. (2004). Biohashing: Two factor authentication featuring fingerprint data and tokenized random number. Pattern Recognition, 37(11), 2245–2255.

[50] Lumini, A., & Nanni, L. (2007). An improved bio-hashing for human authentication. Pattern Recognition, 40(3), 1057–1065. doi:10.1016/j.patcog.2006.05.030.

[51] M.Alizadeh, S.Abolfazli, M.Zamani, S. Baharun, & K. Sakurai, (2015). Authentication in mobile cloud computing: A survey. Journal of Network and Computer Applications.

[52] Hellman, M. E. (2002). An overview of public key cryptography. IEEE Communications Magazine, 40(5), 42-49.

[53] Ranchal, R., Bhargava, B., Othmane, L. B., Lilien, L., Kim, A., Kang, M., & Linderman, M. (2010, October). Protection of identity information in cloud computing without trusted third party. In Reliable Distributed Systems, 2010 29th IEEE Symposium on (pp. 368-372). IEEE.

[54] Boneh, D., Lynn, B., & Shacham, H. (2001). Short signatures from the Weil pairing. In Advances in Cryptology—ASIACRYPT 2001 (pp. 514-532). Springer Berlin Heidelberg.

[55] http://www.cs.ucla.edu/~rafail/PUBLIC/89.pdf

[56] Wu, F., Xu, L., Kumari, S., & Li, X. (2015). A novel and provably secure biometrics-based three-factor remote authentication scheme for mobile client–server networks. Computers & Electrical Engineering, 45, 274-285. [57] Amin, R., & Biswas, G. P. (2015). Design and analysis of bilinear pairing based mutual authentication and key agreement protocol usable in multi-server environment. Wireless Personal Communications, 84(1), 439-462.

[58] Irshad, A., Sher, M., Rehman, E., Ch, S. A., Hassan, M. U., & Ghani, A. (2015). A single round-trip SIP authentication scheme for voice over internet protocol using smart card. Multimedia Tools and Applications, 74(11), 3967-3984.

[59] Irshad, A., Sher, M., Faisal, M. S., Ghani, A., Ul Hassan, M., & Ashraf Ch, S. (2014). A secure authentication scheme for session initiation protocol by using ECC on the basis of the Tang and Liu scheme. Security and Communication Networks, 7(8), 1210-1218.

[60] Irshad, A., Sher, M., Chaudhary, S. A., Naqvi, H., & Farash, M. S. (2016). An efficient and anonymous multi-server authenticated key agreement based on chaotic map without engaging Registration Centre. The Journal of Supercomputing, 72(4), 1623-1644.

[61] L. H. Li, L. C. Lin, and M. S. Hwang,“A remote password authentication scheme for multi-server architecture using neural networks,” IEEE Trans. Neural Netw., vol. 12, no. 6, pp. 1498–1504, Nov. 2001.

[62] I. C. Lin, M. S. Hwang, and L. H. Li, “A new remote user authentication scheme for multi-server architecture,” Future Gener. Comput. Syst., vol. 19, no. 1, pp. 13–22, 2003.

[63] X. Cao and S. Zhong, “Breaking a remote user authentication scheme for multi-server architecture,” IEEE Commun. Lett., vol. 10, no. 8, pp. 580–581, Aug. 2006.

[64] E. J. Yoon and K. Y. Yoo, “Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem,” J. Supercomput., vol. 63, no. 1, pp. 235–255, 2013.

[65] D. He and D. Wang, “Robust biometrics-based authentication scheme for multiserver environment,” IEEE Syst. J., vol. 9, no. 3, pp. 816–823, Sep. 2015.

[66] V. Odelu, A. K. Das, and A. Goswami, “A secure biometrics-based multi-server authentication protocol using smart cards,”IEEE Trans. Inf. Forens. Security, vol. 10, no. 9, pp. 1953–1966, Sep. 2015.

[67] Shamus Software Ltd., Miracl library, http://www.shamus.ie/index.php?page=home

Azeem Irshad received Master’s degree from Arid Agriculture University, Rawalpindi, Pakistan. Currently, he is pursuing his PhD in security for multi-server architectures, from International Islamic University, Islamabad, Pakistan. He authored more than 40 international journal and conference publications, including 23 SCI-E journal publications. He has served as a reviewer for more than 27 reputed journals including IEEE Systems Journal, IEEE Communications Magazine, IEEE Transactions on Industrial Informatics, IEEE Consumer Electronics Magazine, Computer Networks, Information Sciences, CAEE, Cluster Computing, AIHC, Journal of Supercomputing and Wireless Personal Communications, notably. His research interests include strengthening of authenticated key agreements in SIP multimedia, IoT, WBAN, TMIS, WSN, Ad hoc Networks, e-health clouds and multi-server architectures.