Multimed Tools Appl (2016) 75:12705–12725
DOI 10.1007/s11042-015-3194-0

A secure biometric based multi-server authentication scheme for social multimedia networks

Shehzad Ashraf Chaudhry[1]

Received: 3 July 2015 / Revised: 12 November 2015 / Accepted: 18 December 2015 /
Published online: 11 January 2016

Abstract Social networking is one of the major source of massive data. Such data is not only difficult to store, manipulate and maintain but it's open access makes it security prone. Therefore, robust and efficient authentication should be devised to make it invincible against the known security attacks. Moreover, social networking services are intrinsically multi-server environments, therefore compatible and suitable authentication should be designed accordingly. Sundry authentication protocols are being utilized at the moment and many of them are designed for single server architecture. This type of remote architecture resists each user to get itself register with each server if multiple servers are employed to offer online social services. Recently multi-server architecture for authentication has replaced the single server architecture, and it enable users to register once and procure services from multiple servers. A short time ago, Lu et al. presented two authentication schemes based on three factors. Furthermore, both Lu et al.'s schemes are designed for multi-server architecture. Lu et al. claimed the schemes to be invincible against the known attacks. However, this paper shows that one of the Lu et al.'s scheme is susceptible to user anonymity violation and impersonation attacks, whereas Lu et al.'s second scheme is susceptible to user impersonation attack. Therefore an enhanced scheme is introduced in this paper. The proposed scheme is more robust than subsisting schemes. The proposed scheme is thoroughly verified and validated with formal and informal security discussion, and through the popular automated tool ProVerif. The in-depth analysis affirms that proposed scheme is lightweight in terms of computations while attaining mutual authentication and is invincible against the known attacks, hence is more suitable for automated big data analysis for social multimedia networking environments.

Keywords Social multimedia networking · Big data analysis · Biometrics · Authentication · Multi server · Impersonation attack · Anonymity · ProVerif

✉ Shehzad Ashraf Chaudhry
shahzad@iiu.edu.pk

[1] Department of Computer Science and Software Engineering, International Islamic University Islamabad, Islamabad, Pakistan

1 Introduction

Big data refers to the huge amount of data with complicated and diverse structure to be stored and analyzed for retrieving results. This kind of result retrieval is known as big data analysis, which is performed by disclosing concealed pattern and correlations present in the colossal data. Big data analysis is playing a vital role in present day businesses and contemporary science, because it helps organizations and companies to attain competitive benefits through deeper and wealthier insights into precious gigantic data. There are numerous sources for such gigantic data, social networking interaction is one of them. Huge social networking data storage, manipulation and transfer becomes difficult to manage and can be compromised by various security attacks therefore efficient authentication mechanism should be developed to make it more secure and reliable. Moreover social networking services are inherently multi-server environments, therefore authentication schemes must be specifically designed for multi-server architecture in order to maintain compatibility.

The first step was taken up by Lamport [35], by proposing password based authentication scheme. After that researchers proposed numerous authentication schemes based on password for various applications [36, 37, 49, 52]. Although password based authentication schemes are susceptible to a number of attacks but they laid the foundation for advance research in this area. Therefore, two factor authentication scheme are introduced in order to mitigate the security concerns of single factor authentication schemes [3–7, 14–22, 24–31, 44, 50, 53]. Two factor authentication utilize smart card along with password. Moreover, three factor authentication schemes are also introduced not only to improve the security of the transmission between the authentic users but also to provide the integrity and authenticity of the exchanged messages [1, 11, 23, 38–40, 45, 54]. Three factor authentication is achieved by utilizing biometrics along with smart card and password. However most of the authentication schemes are designed specifically for single server architecture, making it incompatible for multi-server architectures.

In 2014 [8] Chuang et al. introduced authentication scheme utilizing biometrics and smart card. They declared the scheme to be secure against the known attacks. Soon, Mishra et al. [46] identified that Chuang et al.'s scheme is not invincible to server spoofing, smart card stolen and impersonation attacks. Mishra et al. proposed authentication scheme using smart card and biometric and declared it to be secure against all security threats. Later on, Lu et al. [41, 42] recognized that Mishra et al.'s scheme is vulnerable to server spoofing and impersonation attacks and fails to provide forward secrecy. In response to Mishra et al.'s scheme Lu et al. introduced two independent three factors authentication schemes [41, 42] for multi-server architecture. Furthermore, Lu et al. declared that their schemes are invincible against the known attacks. However, this paper provide an evidence that Lu et al.'s both schemes can be compromised by the known attacks. We show that Lu et al.'s scheme-1 [41] is insecure against user anonymity violation and impersonation attacks, whereas Lu et al.'s scheme-2 [42] is insecure against user impersonation attack. This paper exhibits that by knowing the public identity of any other user, the unfair user of the system can impersonate him easily.

Rest of the paper is structured as follows: Section 2 presents notations used within the paper and primitive notions concerning one-way hash functions, BioHashing, basics of elliptic curve cryptography and the considered adversarial model. Section 3 presents review of two Lu et al.'s authentication schemes based on three factor for multi server environments, followed by their cryptanalysis performed in Section 4. The proposed scheme is discussed in Section 5. The formal and informal security analysis is performed in

Section 6 followed by automated security validation in Section 7. The performance evaluation is shown in Section 8. The paper is concluded in Section 9.

2 Preliminaries

This section elaborates the notations user through out the paper and some basics relating to hash functions, BioHashing, elliptic curve cryptography and the common adversarial model.

2.1 Notations

We have listed all the notations used in the paper in Table 1.

2.2 BioHashing

The biometrics is the unique and quantifiable characteristic commonly utilized to identify and designate or recognize a particular human. Biometric is practically utilized for authentication purpose and demands the physical presence of a particular person in order to be authenticated. At each imprint, biometric features (such as fingerprint, retina, face recognition and iris recognition etc) may faintly differ from the actual one, leading towards frequent false rejections of legitimate user. Frequent false rejections of legitimate user in turn degrade the performance of the latent system. In 2004, Jin et al. [32] proposed a scheme to tenacity the problem of false rejection. Jin et al.'s scheme implements two factor authentication based on iterated inner product amid biometric characteristics and tokenized pseudo-random numbers. Moreover, in order to implement Jin et al.'s scheme multiple and explicit user codes are engendered and these explicit user codes are designated as BioHash codes. Recently, numerous BioHashing schemes has been introduced [2, 43]. BioHashing is verified to be the most suitable and compatible technique that can be utilized in tiny smart devices such as smart card and smart phone etc.

2.3 Hash functions

A collision resistant hash function $H : \{0, 1\}^* \rightarrow Z_q^*$ takes arbitrary size string Str as input and produces a fixed length code/value $V = H(Str)$. A secure hash function should posses following attributes:

- A minor change in input (Str) results a substantial change in out put V.
- It is computationally easy to find $V = H(Str)$, given $H(.)$ and Str.

Table 1 Notation guide

Notations	Description
$RC, \mathcal{S}_y, \mathcal{U}_x, \mathcal{A}$	Registration center, Server, User, Attacker
$SID_y, ID_{ux}, PW_{ux}, BIO_{ux}$	identities of $\mathcal{S}_y, \mathcal{U}_x, \mathcal{U}_x$'s password, and Biometrics
$x_{ux}, Pub_{sy}, Pri_{sy}$	$\mathcal{U}_x$'s secret key, Public and private key pair of $\mathcal{S}_y$
y_{rs}, PSK_{rs}	RC's secret key, Secret key between $\mathcal{S}_y$ and RC
$SC_{ux}, h(.), H(.), \Vert, \oplus$	$\mathcal{U}_i$'s smart card, Hash, BioHash functions, Concatenation, XOR operators

- For given hash code $V = H(Str)$ and hash function $H(.)$, finding the input Str is computationally infeasible.
- It is difficult to find two inputs $Str_1 \neq Str_2$ such that $H(Str_1) = H(Str_2)$. This property is known as collision resistance property.

Definition 1 [Collision resistant property for secure hash functions] Given a collision resistant secure hash function $H(.)$. The probability that an adversary $\mathcal{A}$ can find a pair $(Str_1 \neq Str_2)$ such that $H(Str_1) = H(Str_2)$ is defined as $Adv_{\mathcal{A}}^{HASH}(t) = Prb[(Str_1, Str_2) \Leftarrow_r \mathcal{A} : (Str_1 \neq Str_2)\ and\ H(Str_1) = H(Str_2)]$, where $\mathcal{A}$ is allowed to select a pair (Str_1, Str_2) at random. $\mathcal{A}$'s advantage is computed over the random choices made during polynomial time (t). The collision resistant property implies that $Adv_{\mathcal{A}}^{HASH}(t) \leq \epsilon$ for any sufficiently small $\epsilon > 0$.

2.4 Elliptic curve cryptography

A non singular elliptic cure $y^2 = x^3 + ax + b \mod p$ is the set of finite solutions $E_p(a, b)$ such that $(x, y) \in Z_p^* \times Z_p$, a, b are chosen carefully to accommodate $4a^3 + 27b^2 \mod p \neq 0$ while p is a selected large prime number such that$|p| \geq 160\ bits$. The scalar multiplication over the curve is solicited as repeated addition i.e. $kS = S + S + S + ... + S$ $(k\ times)$, for a given point S and a scalar k. The parameters (a, b, p, S, k) must belong to finite field F_p. E is considered as abelian group and a point at infinity O is termed as the identity element.

Definition 2 [Elliptic curve discrete logarithm problem (ECDLP)] Given two random point $U, V \in E_p(a, b)$, find a scalar x such that $U = xV$. The probability that a polynomial time (t) bound adversary $\mathcal{A}$ can compute x is as follows: $Adv_{\mathcal{A}}^{ECDLP}(t) = Prb[(\mathcal{A}(U = xV, V) = x : x \in Z_p]$. The ECDLP assumption implies that $Adv_{\mathcal{A}}^{ECDLP}(t) \leq \epsilon$.

2.5 Adversarial model

In this paper, we consider the common adversarial model as mentioned in [4, 9, 12, 13]. Where according to capabilities of the adversary $\mathcal{A}$, following assumptions are made:

1. $\mathcal{A}$ completely controls the public communication link. $\mathcal{A}$ is able to intercept, replay, modify, remove or can send a new fabricated message.
2. The information stored in a smart card can be extracted by $\mathcal{A}$ using power analysis [33, 47] provided he has possession of the card.
3. $\mathcal{A}$ may be some outsider or some dishonest user of the system and knows all public parameters.
4. $\mathcal{A}$ knows the identities and public keys of the registered users and servers.
5. It is assumed that all servers of the system are honest and $\mathcal{A}$ is not allowed to compromise any server.

3 Review of Lu et al.'s schemes

In this section, we briefly review Lu et al.'s multi-server biometric based authentication schemes [41, 42] in Subsections 3.1 and 3.2 respectively.

3.1 Review of Lu et al.'s scheme-1 [41]

Lu et al.'s biometric based authentication scheme for multi-server environments [41] is illustrated in Fig. 1 and is elaborated in following three phases:

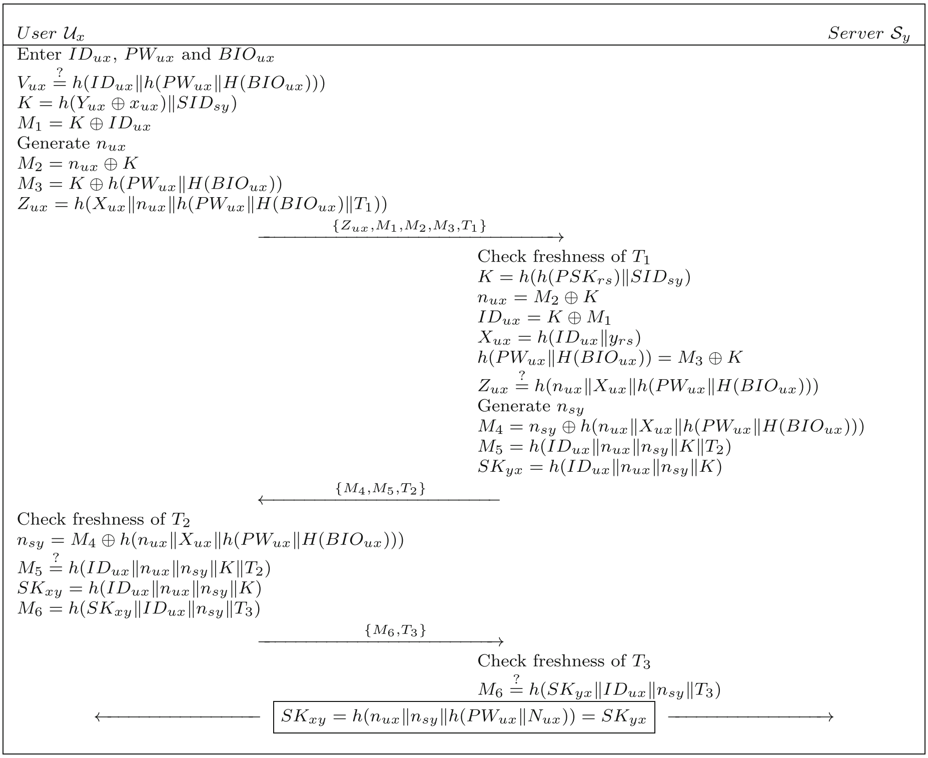
3.1.1 *Registration phase*

$\mathcal{U}_x$ selects his identity ID_{ux}, password PW_{ux} and imprints his biometrics BIO_{ux}. Further $\mathcal{U}_x$ sends $\{ID_{ux}, h(PW_{ux}\|H(BIO_{ux}))\}$ to RC on a private channel. Upon reception, RC computes $X_{ux} = h(ID_{ux}\|y_{rs})$ and $V_{ux} = h(ID_{ux}\|h(PW_{ux}\|H(BIO_{ux})))$ and stores $X_{ux}, h(PSK_{rs})$ and V_{ux} in the smart card SC_{ux}. RC sends smart card (SC_{ux}) to $\mathcal{U}_x$. Upon reception of smart card, $\mathcal{U}_x$ computes $Y_{ux} = h(PSK_{rs}) \oplus x_{ux}$. Finally, smart card contains $\{X_{ux}, Y_{ux}, V_{ux}, h(.)\}$.

3.1.2 *Login and authentication phase*

$\mathcal{U}_x$ enters his smart card in specialized reader and inputs his biometric BIO_{ux} , password PW_{ux} and identity ID_{ux}. Following steps are performed between the smart card (SC_{ux}) and the server $\mathcal{S}_y$:

Step L1A1: SC_{ux} checks $V_{ux} \overset{?}{=} h(ID_{ux}\|h(PW_{ux}\|H(BIO_{ux})))$, if it is not true, session is aborted by SC_{ux}. Otherwise, SC_{ux} computes $K = h(Y_{ux} \oplus x_{ux})\|SID_{sy})$ and



User $\mathcal{U}_x$ *Server* $\mathcal{S}_y$

Enter ID_{ux}, PW_{ux} and BIO_{ux}
$V_{ux} \overset{?}{=} h(ID_{ux}\|h(PW_{ux}\|H(BIO_{ux})))$
$K = h(Y_{ux} \oplus x_{ux})\|SID_{sy})$
$M_1 = K \oplus ID_{ux}$
Generate n_{ux}
$M_2 = n_{ux} \oplus K$
$M_3 = K \oplus h(PW_{ux}\|H(BIO_{ux}))$
$Z_{ux} = h(X_{ux}\|n_{ux}\|h(PW_{ux}\|H(BIO_{ux})\|T_1))$

$\xrightarrow{\{Z_{ux}, M_1, M_2, M_3, T_1\}}$

Check freshness of T_1
$K = h(h(PSK_{rs})\|SID_{sy})$
$n_{ux} = M_2 \oplus K$
$ID_{ux} = K \oplus M_1$
$X_{ux} = h(ID_{ux}\|y_{rs})$
$h(PW_{ux}\|H(BIO_{ux})) = M_3 \oplus K$
$Z_{ux} \overset{?}{=} h(n_{ux}\|X_{ux}\|h(PW_{ux}\|H(BIO_{ux})))$
Generate n_{sy}
$M_4 = n_{sy} \oplus h(n_{ux}\|X_{ux}\|h(PW_{ux}\|H(BIO_{ux})))$
$M_5 = h(ID_{ux}\|n_{ux}\|n_{sy}\|K\|T_2)$
$SK_{yx} = h(ID_{ux}\|n_{ux}\|n_{sy}\|K)$

$\xleftarrow{\{M_4, M_5, T_2\}}$

Check freshness of T_2
$n_{sy} = M_4 \oplus h(n_{ux}\|X_{ux}\|h(PW_{ux}\|H(BIO_{ux})))$
$M_5 \overset{?}{=} h(ID_{ux}\|n_{ux}\|n_{sy}\|K\|T_2)$
$SK_{xy} = h(ID_{ux}\|n_{ux}\|n_{sy}\|K)$
$M_6 = h(SK_{xy}\|ID_{ux}\|n_{sy}\|T_3)$

$\xrightarrow{\{M_6, T_3\}}$

Check freshness of T_3
$M_6 \overset{?}{=} h(SK_{yx}\|ID_{ux}\|n_{sy}\|T_3)$

$\longleftarrow \boxed{SK_{xy} = h(n_{ux}\|n_{sy}\|h(PW_{ux}\|N_{ux})) = SK_{yx}} \longrightarrow$

Fig. 1 Lu et al.'s Scheme-1[41]

$M_1 = K \oplus ID_{ux}$. Then SC_{ux} generates a nonce $M_2 = n_{ux} \oplus K$, $M_3 = K \oplus h(PW_{ux} \| H(BIO_{ux}))$ and $Z_{ux} = h(X_{ux} \| n_{ux} \| h(PW_{ux} \| H(BIO_{ux}) \| T_1))$, where T_1 is the fresh time stamp.

Step L1A2: Smart card SC_{ux} sends $\{M_1, M_2, M_3, Z_{ux}, T_1\}$ to $\mathcal{S}_y$.

Step L1A3: $\mathcal{S}_y$ upon receiving login message, checks the freshness of T_1, aborts the session if T_1 is not fresh. Otherwise, computes $K = h(h(PSK_{rs}) \| SID_{sy})$, $n_{ux} = M_2 \oplus K$, $ID_{ux} = K \oplus M_1$, $X_{ux} = h(ID_{ux} \| y_{rs})$ and $h(PW_{ux} \| H(BIO_{ux})) = M_3 \oplus K$.

Step L1A4: $\mathcal{S}_y$ verifies $Z_{ux} \underset{=}{?} h(n_{ux} \| X_{ux} \| h(PW_{ux} \| H(BIO_{ux})))$, if it is not true, $\mathcal{S}_y$ aborts the session. Otherwise, $\mathcal{S}_y$ selects a random number n_{sy} and computes $M_4 = n_{sy} \oplus h(n_{ux} \| X_{ux} \| h(PW_{ux} \| H(BIO_{ux})))$, $M_5 = h(ID_{ux} \| n_{ux} \| n_{sy} \| K \| T_2)$ and the session key $SK_{yx} = h(n_{ux} \| n_{sy} \| h(PW_{ux} \| N_{ux}))$. Further $\mathcal{S}_y$ sends $\{M_4, M_5, T_2\}$ to $\mathcal{U}_x$, where T_2 is current time stamp.

Step L1A5: Upon reception, $\mathcal{U}_x$ checks the freshness of T_2, if T_2 is fresh $\mathcal{U}_x$ computes $n_{sy} = M_4 \oplus h(n_{ux} \| X_{ux} \| h(PW_{ux} \| H(BIO_{ux})))$ and checks validity of $M_5 \underset{=}{?} h(ID_{ux} \| n_{ux} \| n_{sy} \| K \| T_2)$. If it is not valid $\mathcal{U}_x$ aborts the session. Otherwise, $\mathcal{U}_x$ computes the session key $SK_{xy} = h(ID_{ux} \| n_{ux} \| n_{sy} \| K)$ and $M_6 = h(SK_{xy} \| ID_{ux} \| n_{sy} \| T_3)$. Finally $\mathcal{U}_x$ sends M_6, T_3 to $\mathcal{S}_y$, where T_3 is current time stamp.

Step L1A6: $\mathcal{S}_y$ upon receiving the message checks $M_6 \underset{=}{?} h(SK_{yx} \| ID_{ux} \| n_{sy} \| T_3)$ if it holds, $\mathcal{S}_y$ considers $\mathcal{U}_x$ as authenticated. The session key shared among both is:

$$SK_{xy} = h(ID_{ux} \| n_{ux} \| n_{sy} \| K) = SK_{yx} \tag{1}$$

3.1.3 Password change phase

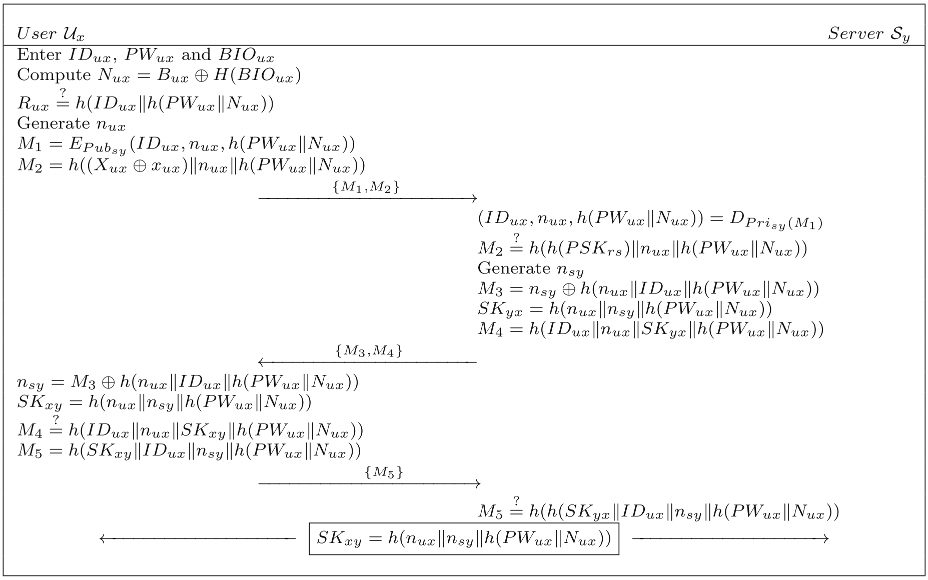
To change password, $\mathcal{U}_x$ enters his smart card in the reader, then inputs his password PW_{ux}, identity ID_{ux} and biometrics BIO_{ux}. The smart card verifies $V_{ux} \underset{=}{?} h(ID_{ux} \| h(PW_{ux} \| H(BIO_{ux})))$, if it is true $\mathcal{U}_x$ is asked to enter his new password PW_{ux}^{new} the smart card computes $V_{ux}^{new} = h(ID_{ux} \| h(PW_{ux} \| H(BIO_{ux})))$ and replaces V_{ux} by V_{ux}^{new}.

3.2 Review of Lu et al.'s scheme-2 [42]

In this section, we briefly review Lu et al.'s scheme-2 [42] . Lu et al. employed public key techniques to achieve user anonymity and forward secrecy. Their scheme involves three participants: a user $\mathcal{U}_x$, a server $\mathcal{S}_y$ and the registration center RC. The scheme is illustrated in Fig. 2. We further elaborate Lu et al.'s scheme by following three phases:

3.2.1 Registration phase

Registration involves following three steps: $\mathcal{U}_x$ selects his identity ID_{ux}, password PW_{ux}, a random number N_{ux} along with his master private key x_{ux}. Then $\mathcal{U}_x$ scans his biometrics BIO_{ux}. Further $\mathcal{U}_x$ sends $\{ID_{ux}, h(PW_{ux}, N_{ux})\}$ to RC on a private channel. RC computes $R_{ux} = h(ID_{ux} \| h(PW_{ux} \| N_{ux}))$ and personalizes the smart card SC_{ux} by $\{R_{ux}, h(PSK_{rs})\}$, where PSK_{rs} is the shared secret key between RC and $\mathcal{S}_y$. RC using private channel sends SC_{ux} to $\mathcal{U}_x$. Upon receiving smart card, $\mathcal{U}_x$ computes



User $\mathcal{U}_x$ *Server* $\mathcal{S}_y$

Enter ID_{ux}, PW_{ux} and BIO_{ux}
Compute $N_{ux} = B_{ux} \oplus H(BIO_{ux})$
$R_{ux} \overset{?}{=} h(ID_{ux}\|h(PW_{ux}\|N_{ux}))$
Generate n_{ux}
$M_1 = E_{Pub_{sy}}(ID_{ux}, n_{ux}, h(PW_{ux}\|N_{ux}))$
$M_2 = h((X_{ux} \oplus x_{ux})\|n_{ux}\|h(PW_{ux}\|N_{ux}))$

$\xrightarrow{\{M_1,M_2\}}$

$(ID_{ux}, n_{ux}, h(PW_{ux}\|N_{ux})) = D_{Pri_{sy}(M_1)}$
$M_2 \overset{?}{=} h(h(PSK_{rs})\|n_{ux}\|h(PW_{ux}\|N_{ux}))$
Generate n_{sy}
$M_3 = n_{sy} \oplus h(n_{ux}\|ID_{ux}\|h(PW_{ux}\|N_{ux}))$
$SK_{yx} = h(n_{ux}\|n_{sy}\|h(PW_{ux}\|N_{ux}))$
$M_4 = h(ID_{ux}\|n_{ux}\|SK_{yx}\|h(PW_{ux}\|N_{ux}))$

$\xleftarrow{\{M_3,M_4\}}$

$n_{sy} = M_3 \oplus h(n_{ux}\|ID_{ux}\|h(PW_{ux}\|N_{ux}))$
$SK_{xy} = h(n_{ux}\|n_{sy}\|h(PW_{ux}\|N_{ux}))$
$M_4 \overset{?}{=} h(ID_{ux}\|n_{ux}\|SK_{xy}\|h(PW_{ux}\|N_{ux}))$
$M_5 = h(SK_{xy}\|ID_{ux}\|n_{sy}\|h(PW_{ux}\|N_{ux}))$

$\xrightarrow{\{M_5\}}$

$M_5 \overset{?}{=} h(h(SK_{yx}\|ID_{ux}\|n_{sy}\|h(PW_{ux}\|N_{ux}))$

$\longleftarrow \boxed{SK_{xy} = h(n_{ux}\|n_{sy}\|h(PW_{ux}\|N_{ux}))} \longrightarrow$

Fig. 2 Lu et al.'s scheme-2 [42]

$X_{ux} = h(PSK_{rs}) \oplus x_{ux}$, $B_{ux} = N_{ux} \oplus H(BIO_{ux})$. Then $\mathcal{U}_x$ deletes $h(PSK_{rs})$ from smart card (SC_{ux}) and stores X_{ux} and B_{ux} in the smart card (SC_{ux}). Finally the smart card (SC_{ux}) contains $\{R_{ux}, X_{ux}, B_{ux}, h()\}$.

3.2.2 Login and authentication phase

During login phase $\mathcal{U}_x$ inserts his SC_{ux} into card reader, imprints his biometrics (BIO_{ux}) and submits ID_{ux} and PW_{ux}. The steps performed by SC_{ux} and $\mathcal{S}_y$ are as follows:

Step L1A1: SC_{ux} computes $N_{ux} = B_{ux} \oplus H(BIO_{ux})$ and $R'_{ux} = h(ID_{ux}\|h(PW_{ux}\|N_{ux}))$.
Step L1A2: SC_{ux} verifies $R_{ux} \overset{?}{=} h(ID_{ux}\|h(PW_{ux}\|N_{ux}))$, if not true, SC_{ux} aborts the session.
Step L1A3: SC_{ux} generates a random number n_{sy} and computes $M_1 = E_{Pub_{sy}}(ID_{ux}, n_{ux}, h(PW_{ux}\|N_{ux}))$ and $M_2 = h((X_{ux} \oplus x_{ux})\|n_{ux}\|h(PW_{ux}\|N_{ux}))$
Step L1A4: Further, SC_{ux} sends login message $\{M_1, M_2\}$ to $\mathcal{S}_y$.
Step L1A5: For the received login message, $\mathcal{S}_y$ using his private key decrypts M_1 to get $(ID_{ux}, n_{ux}, h(PW_{ux}\|N_{ux}))$.
Step L1A6: $\mathcal{S}_y$ checks whether $M_2 \overset{?}{=} h(h(PSK_{rs})\|n_{ux}\|h(PW_{ux}\|N_{ux}))$, if not true $\mathcal{S}_y$ aborts the session. Otherwise, $\mathcal{S}_y$ selects a random number n_{sy} and computes $M_3 = n_{sy} \oplus h(n_{ux}\|ID_{ux}\|h(PW_{ux}\|N_{ux}))$, the session key $SK_{yx} = h(n_{ux}\|n_{sy}\|h(PW_{ux}\|N_{ux}))$ and $M_4 = h(ID_{ux}\|n_{ux}\|SK_{yx}\|h(PW_{ux}\|N_{ux}))$. Further $\mathcal{S}_y$ sends $\{M_3, M_4\}$ to $\mathcal{U}_x$.
Step L1A7: For the received login message, $\mathcal{U}_x$ computes $n_{sy} = M_3 \oplus h(n_{ux}\|ID_{ux}\| h(PW_{ux}\|N_{ux}))$ and session key $SK_{xy} = h(n_{ux}\|n_{sy}\|h(PW_{ux}\|N_{ux}))$. $\mathcal{U}_x$

then checks $M_4 \overset{?}{=} h(ID_{ux} \| n_{ux} \| SK_{xy} \| h(PW_{ux} \| N_{ux}))$. If it holds, $\mathcal{U}_x$ ponders $\mathcal{S}_y$ as authenticated.

Step L1A8: Finally, $\mathcal{U}_x$ computes and sends $M_5 = h(SK_{xy} \| ID_{ux} \| n_{sy} \| h(PW_{ux} \| N_{ux}))$ to $\mathcal{S}_y$.

Step L1A9: $\mathcal{S}_y$ checks $M_5 \overset{?}{=} h(h(SK_{yx} \| ID_{ux} \| n_{sy} \| h(PW_{ux} \| N_{ux}))$ if it holds, $\mathcal{S}_y$ ponders $\mathcal{U}_x$ as authenticated.

The computed shared key between $\mathcal{U}_x$ and $\mathcal{S}_y$ is:

$$SK_{xy} = h(n_{ux} \| n_{sy} \| h(PW_{ux} \| N_{ux})) = SK_{yx} \tag{2}$$

3.2.3 Password change phase

$\mathcal{U}_x$ inserts his smart card (SC_{ux}) in specialized reader. $\mathcal{U}_x$ then inputs ID_{ux}, PW_{ux} and BIO_{ux}. SC_{ux} computes $N_{ux} = B_{ux} \oplus H(BIO_{ux})$ and checks $R_{ux} = h(ID_{ux} \| h(PW_{ux} \| N_{ux}))$, if it holds SC_{ux} asks for new password. $\mathcal{U}_x$ inputs new password PW_{ux}^{new}. SC_{ux} computes $R_{ux}^{new} = h(ID_{ux} \| h(PW_{ux}^{new} \| N_{ux}))$. Finally SC_{ux} replaces R_{ux} by R_{ux}^{new}.

4 Cryptanalysis of Lu et al.'s schemes

This section performs cryptanalysis of Lu et al.'s schemes. We show that Lu et al's scheme-1 is vulnerable to: (1) user anonymity violation attack and (2) user impersonation attack. Likewise, we show that Lu et al.'s scheme-2 is vulnerable to user impersonation attack.

4.1 Weaknesses of Lu et al.'s scheme-1

4.1.1 User anonymity violation attack

To mount a successful user impersonation attack, initially an attacker $\mathcal{A}$ selects his identity ID_{ua}, password PW_{ua}, biometrics BIO_{ua} and his own secret key x_{ua}. Then $\mathcal{A}$ registers to the system and obtains a smart card containing $X_{ua} = h(ID_{ua} \| y_{rs})$, $V_{ua} = h(ID_{ua} \| h(PW_{ua} \| H(BIO_{ua})))$ and $Y_{ua} = h(PSK_{rs}) \oplus x_{ua}$. $\mathcal{A}$ performs following steps for the successful anonymity violation attack:

Step L1A1: $\mathcal{A}$ extracts $h(PSK_{rs})$ as follows:

$$h(PSK_{rs}) = x_{ua} \oplus Y_{ua} \tag{3}$$

Step L1A2: When $\mathcal{U}_x$ initiates the authentication requests by sending $Z_{ux}, M_1, M_2, M_3, T_1$ to $\mathcal{S}_y$. $\mathcal{A}$ intercepts the message and computes:

$$K = h(h(PSK_{rs} \| SID_{sy})) \tag{4}$$

$$n_{ux} = M_2 \oplus K \tag{5}$$

$$ID_{ux} = K \oplus M_1 \tag{6}$$

In (6) ID_{ux} is the real identity of user $\mathcal{U}_x$. Hence, $\mathcal{A}$ has successfully break the anonymity of $\mathcal{U}_x$.

4.1.2 User impersonation attack

Here, we prove that Lu et al.'s scheme-1 is vulnerable to impersonation attack. We show that an adversary $\mathcal{A}$ can impersonate any other registered user of the system if he becomes able to steal his smart card. Initially $\mathcal{A}$ extracts $X_{ux} = h(ID_{ux}\|y_{rs})$ out of a stolen smart card. Then he performs following steps to impersonate himself as $\mathcal{U}_x$:

Step L1A1: $\mathcal{A}$ computes:

$$K = h(h(PSK_{rs}\|SID_{sy})) \tag{7}$$

$$M_1 = K \oplus ID_{ux} \tag{8}$$

Step L1A2: $\mathcal{A}$ generates two random numbers n_{ua} and P_{ua}. Then generates time stamp T_1 and computes:

$$M_2 = n_{ua} \oplus K \tag{9}$$

$$M_3 = K \oplus P_{ua} \tag{10}$$

$$Z_{ua} = h(X_{ux}\|n_{ua}\|P_{ua}\|T_1) \tag{11}$$

Step L1A3: $\mathcal{A}$ sends $\{Z_{ua}, M_1, M_2, M_3, T_1\}$ to $\mathcal{S}_y$.
Step L1A4: $\mathcal{S}_y$ upon receiving login message, checks the freshness of T_1, as T_1 is freshly generated so $\mathcal{S}_y$ computes:

$$K = h(h(PSK_{rs})\|SID_{sy}) \tag{12}$$

$$n_{ua} = M_2 \oplus K \tag{13}$$

$$ID_{ux} = K \oplus M_1 \tag{14}$$

$$X_{ux} = h(ID_{ux}\|y_{rs}) \tag{15}$$

$$P_{ua} = M_3 \oplus K \tag{16}$$

Step L1A5: $\mathcal{S}_y$ verifies $Z_{ux}\underline{?}h(n_{ua}\|X_{ux}\|P_{ua})$ and finds it true. $\mathcal{S}_y$ then selects a random number n_{sy} and computes:

$$M_4 = n_{sy} \oplus h(n_{ua}\|X_{ux}\|P_{ua})) \tag{17}$$

$$M_5 = h(ID_{ux}\|n_{ua}\|n_{sy}\|K\|T_2) \tag{18}$$

$$SK_{yx} = h(n_{ua}\|n_{sy}\|P_{ua})) \tag{19}$$

Step L1A6: Further $\mathcal{S}_y$ sends $\{M_4, M_5, T_2\}$ to $\mathcal{U}_x$, where T_2 is current time stamp.
Step L1A7: Upon reception, $\mathcal{A}$ computes:

$$n_{sy} = M_4 \oplus h(n_{ua}\|X_{ux}\|Pua) \tag{20}$$

$$SK_{xy} = h(ID_{ux}\|n_{ua}\|n_{sy}\|K) \tag{21}$$

$$M_6 = h(SK_{xy}\|ID_{ux}\|n_{sy}\|T_3) \tag{22}$$

Step L1A8: Finally $\mathcal{A}$ sends M_6, T_3 to $\mathcal{S}_y$, where T_3 is current time stamp. $\mathcal{S}_y$ upon receiving the message checks $M_6\underline{?}h(SK_{yx}\|ID_{ux}\|n_{sy}\|T_3)$ and finds it true.

Hence, $\mathcal{A}$ has successfully deceived $\mathcal{S}_y$ by impersonating himself as $\mathcal{U}_x$. The session key shared among both is:

$$SK_{xy} = h(ID_{ux}\|n_{ua}\|n_{sy}\|K) \tag{23}$$

4.2 Weaknesses of Lu et al.'s scheme-2

This section elaborates the weaknesses of Lu et al.'s scheme-2 against user impersonation attack. We show that a dishonest legal user $\mathcal{A}$ can easily masquerade himself as an other honest user $\mathcal{U}_x$ considering the common adversarial model as mentioned in Subsection 2.5.

4.2.1 User impersonation attack

Let $\mathcal{A}$ be a legal user having smart card SC_{ua} and wants to impersonate himself as another user $\mathcal{U}_x$. Following steps will be performed by $\mathcal{A}$ for a successful forgery attack to $\mathcal{S}_y$:

Step L1A1: $\mathcal{A}$ extracts the information stored in SC_{ua} and computes:

$$h(PSK_{rs}) = X_{ua} \oplus x_{ua} \tag{24}$$

Step L1A2: $\mathcal{A}$ generates two random number n_{ua} and P_{ua} and computes:

$$M_{\bar{1}} = E_{Pub_{sy}}(ID_{ux}, n_{ua}, P_{ua}) \tag{25}$$

$$M_{\bar{2}} = h((X_{ua} \oplus x_{ua})\|n_{ua}\|P_{ua}) \tag{26}$$

Step L1A3: $\mathcal{A}$ sends $M_{\bar{1}}$ and $M_{\bar{2}}$ as login message to $\mathcal{S}_j$.
Step L1A4: For the received login message, $\mathcal{S}_y$ decrypts $M_{\bar{2}}$ to obtain:

$$(ID_{ux}, n_{ua}, P_{ua}) = D_{Pri_{sy}}(M_{\bar{1}}) \tag{27}$$

Step L1A5: $\mathcal{S}_y$ further verifies $M_{\bar{2}} \stackrel{?}{=} h(h(PSK_{rs})\|n_{ua}\|P_{ua})$ and finds it to be true.
Step L1A6: $\mathcal{S}_y$ further selects n_{sy} and computes:

$$M_3 = n_{sy} \oplus h(n_{ua}\|ID_{ux}\|P_{ua}) \tag{28}$$

$$SK_{yx} = h(n_{ux}\|n_{sy}\|P_{ua}) \tag{29}$$

$$M_4 = h(ID_{ux}\|n_{ua}\|SK_{yx}\|P_{ua}) \tag{30}$$

Step L1A7: $\mathcal{S}_y$ sends M_3 and M_4 to $\mathcal{U}_x$ as response message.
Step L1A8: $\mathcal{A}$ intercepts the message and computes:

$$n_{sy} = M_3 \oplus h(n_{ua}\|ID_{ux}\|P_{ua}) \tag{31}$$

$$SK_{xy} = h(n_{ua}\|n_{sy}\|P_{ua}) \tag{32}$$

$$M_5 = h(SK_{xy}\|ID_{ux}\|n_{sy}\|P_{ua}) \tag{33}$$

Step L1A9: $\mathcal{A}$ sends M_5 to $\mathcal{S}_y$.
Step L1A10: $\mathcal{S}_y$ checks $M_5 \stackrel{?}{=} h(h(SK_{yx}\|ID_{ux}\|n_{sy}\|P_{ua})$ and finds it to be true.

Hence, $\mathcal{A}$ successfully deceived $\mathcal{S}_y$ by impersonating himself as $\mathcal{U}_x$. The shared key between $\mathcal{A}$ and $\mathcal{S}_y$ is:

$$SK_{xy} = h(n_{ua}\|n_{sy}\|P_{ua}) = SK_{yx} \tag{34}$$

5 Proposed scheme

In this section, we propose an improved and secure biometric based three factor authentication scheme for social multimedia networks to overcome the weaknesses of Lu et al.'s schemes. The proposed scheme is depicted in Fig. 3 and is explained in following four subsections:

5.1 Initialization

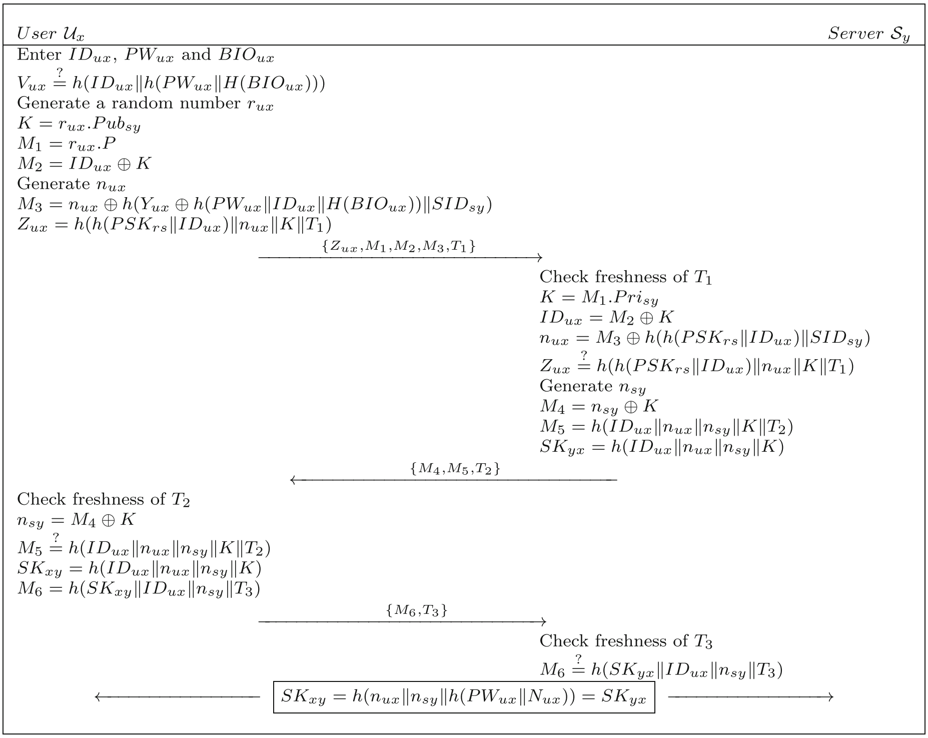
In this phase system parameters are selected by registration server. Initially registration server RC selects an elliptic curve $E_p(a, b) \mod p$, a base point P over $E_p(a, b)$, a one way hash function $h(.)$, BioHashing $H(.)$ and a shared key with all servers PSK_{rs}. Finally RC publishes system public parameters $E_p(a, b)$, $h(.)$, $H(.)$.

5.2 Registration phase

In this phase both the users and servers registers with the registration server. Following two subsections describes the process of registration:

5.2.1 Server registration

To register with the system, a server $\mathcal{S}_y$ selects his identity SID_{sy} and his private key Pri_{sy}. Then $\mathcal{S}_y$ computes his public key $Pub_{sy} = Pri_{sy}.P$ and sends his identity SID_{sy} and his



User $\mathcal{U}_x$ *Server* $\mathcal{S}_y$

Enter ID_{ux}, PW_{ux} and BIO_{ux}

$V_{ux} \overset{?}{=} h(ID_{ux}\|h(PW_{ux}\|H(BIO_{ux})))$

Generate a random number r_{ux}

$K = r_{ux}.Pub_{sy}$

$M_1 = r_{ux}.P$

$M_2 = ID_{ux} \oplus K$

Generate n_{ux}

$M_3 = n_{ux} \oplus h(Y_{ux} \oplus h(PW_{ux}\|ID_{ux}\|H(BIO_{ux}))\|SID_{sy})$

$Z_{ux} = h(h(PSK_{rs}\|ID_{ux})\|n_{ux}\|K\|T_1)$

$\xrightarrow{\{Z_{ux},M_1,M_2,M_3,T_1\}}$

Check freshness of T_1

$K = M_1.Pri_{sy}$

$ID_{ux} = M_2 \oplus K$

$n_{ux} = M_3 \oplus h(h(PSK_{rs}\|ID_{ux})\|SID_{sy})$

$Z_{ux} \overset{?}{=} h(h(PSK_{rs}\|ID_{ux})\|n_{ux}\|K\|T_1)$

Generate n_{sy}

$M_4 = n_{sy} \oplus K$

$M_5 = h(ID_{ux}\|n_{ux}\|n_{sy}\|K\|T_2)$

$SK_{yx} = h(ID_{ux}\|n_{ux}\|n_{sy}\|K)$

$\xleftarrow{\{M_4,M_5,T_2\}}$

Check freshness of T_2

$n_{sy} = M_4 \oplus K$

$M_5 \overset{?}{=} h(ID_{ux}\|n_{ux}\|n_{sy}\|K\|T_2)$

$SK_{xy} = h(ID_{ux}\|n_{ux}\|n_{sy}\|K)$

$M_6 = h(SK_{xy}\|ID_{ux}\|n_{sy}\|T_3)$

$\xrightarrow{\{M_6,T_3\}}$

Check freshness of T_3

$M_6 \overset{?}{=} h(SK_{yx}\|ID_{ux}\|n_{sy}\|T_3)$

$\longleftarrow \boxed{SK_{xy} = h(n_{ux}\|n_{sy}\|h(PW_{ux}\|N_{ux})) = SK_{yx}} \longrightarrow$

Fig. 3 Proposed scheme

public key Pub_{sy} to RC. Upon reception, RC shares the secret key PSK_{rs} with $\mathcal{S}_y$ and publishes $\mathcal{S}_y$'s public key Pub_{sy}.

5.2.2 User registration

User registration involves following three steps:

Step L1A1: $\mathcal{U}_x$ selects his identity ID_{ux}, password PW_{ux} and scans his biometrics BIO_{ux}. Further $\mathcal{U}_x$ sends $\{ID_{ux}, h(PW_{ux}\|H(BIO_{ux}))\}$ to RC on a private channel.

Step L1A2: Upon reception, RC computes $V_{ux} = h(ID_{ux}\|h(PW_{ux}\|H(BIO_{ux})))$ and $h(PSK_{rs}\|ID_{ux})$ and stores $h(PSK_{rs}\|ID_{ux})$ and V_{ux} in the smart card SC_{ux}. RC sends smart card (SC_{ux}) to $\mathcal{U}_x$.

Step L1A3: Upon reception of smart card, $\mathcal{U}_x$ computes $Y_{ux} = h(PSK_{rs}\|ID_{ux}) \oplus h(PW_{ux}\|ID_{ux}\|H(BIO_{ux}))$. Finally, smart card contains $\{Y_{ux}, V_{ux}, h(.)\}$.

5.3 Login and authentication phase

Login phase starts when a user $\mathcal{U}_x$ enters his SC_{ux} into card reader, embosses his biometrics (BIO_{ux}) and enters ID_{ux} and PW_{ux}. The subsequent steps accomplished by SC_{ux} and $\mathcal{S}_y$ are as under:

Step L1A1: SC_{ux} calculates $h(ID_{ux}\|h(PW_{ux}\|H(BIO_{ux})))$ and confirms $V_{ux}\overset{?}{=}h(ID_{ux}\|h(PW_{ux}\|H(BIO_{ux})))$, if condition does not hold, SC_{ux} terminates the session.

Step L1A2: SC_{ux} produces a random number r_{ux} and calculates $K = r_{ux}.Pub_{sy}$, $M_1 = r_{ux}.P$ and $M_2 = ID_{ux} \oplus K$.

Step L1A3: Moreover, SC_{ux} produces a random number n_{ux} and calculates $M_3 = n_{ux} \oplus h(Y_{ux} \oplus h(PW_{ux}\|ID_{ux}\|H(BIO_{ux}))\|SID_{sy})$ and $Z_{ux} = h(h(PSK_{rs}\|ID_{ux})\|n_{ux}\|K\|T_1)$.

Step L1A4: Thereafter, SC_{ux} transmits login message $\{Z_{ux}, M_1, M_2, M_3, T_1\}$ to $\mathcal{S}_y$.

Step L1A5: On getting login message, $\mathcal{S}_y$ verifies freshness of T_1.

Step L1A6: $\mathcal{S}_y$ calculates $K = M_1.Pri_{sy}$ with his private key and also calculates $ID_{ux} = M_2 \oplus K$ and $n_{ux} = M_3 \oplus h(h(PSK_{rs}\|ID_{ux})\|SID_{sy})$.

Step L1A7: $\mathcal{S}_y$ verifies $Z_{ux}\overset{?}{=}h(h(PSK_{rs}\|ID_{us})\|n_{ux}\|K\|T_1)$, if not holds, $\mathcal{S}_y$ terminates the session. Otherwise, $\mathcal{S}_y$ generates a random number n_{sy} and calculates $M_4 = n_{sy} \oplus K$, $M_5 = h(ID_{ux}\|n_{ux}\|n_{sy}\|K\|T_2)$ and the session key $SK_{yx} = h(ID_{ux}\|n_{ux}\|n_{sy}\|K)$. Further $\mathcal{S}_y$ sends $\{M_4, M_5, T_2\}$ to $\mathcal{U}_x$.

Step L1A8: On receiving login message, $\mathcal{U}_x$ verifies freshness of T_2. computes $n_{sy} = M_4 \oplus K$ and confirms $M_5\overset{?}{=}h(ID_{ux}\|n_{ux}\|n_{sy}\|K\|T_2)$, if holds, $\mathcal{U}_x$ cogitates $\mathcal{S}_y$ as authenticated. Then session key is computed as $SK_{xy} = h(ID_{ux}\|n_{ux}\|n_{sy}\|K)$.

Step L1A9: After that, $\mathcal{U}_x$ calculates $M_6 = h(SK_{xy}\|ID_{ux}\|n_{sy}\|T_3)$ and and transmits $\{M_6, T_3\}$ to $\mathcal{S}_y$.

Step L1A10: $\mathcal{S}_y$ checks the freshness of T_3 and verifies $M_6\overset{?}{=}h(SK_{yx}\|ID_{ux}\|n_{sy}\|T_3)$ if it holds, $\mathcal{S}_y$ cogitates $\mathcal{U}_x$ as authenticated.

The derived shared key between $\mathcal{U}_x$ and $\mathcal{S}_y$ is:

$$SK_{xy} = h(n_{ux}\|n_{sy}\|h(PW_{ux}\|N_{ux})) = SK_{yx} \tag{35}$$

5.4 Password change phase

$\mathcal{U}_x$ inserts his smart card (SC_{ux}) in specialized reader. $\mathcal{U}_x$ then inputs ID_{ux}, PW_{ux} and BIO_{ux}. SC_{ux} computes $N_{ux} = B_{ux} \oplus H(BIO_{ux})$ and checks $R_{ux} = h(ID_{ux}\|h(PW_{ux}\|N_{ux}))$, if it hold SC_{ux} asks for new password. $\mathcal{U}_x$ inputs new password PW_{ux}^{new}. SC_{ux} computes $R_{ux}^{new} = h(ID_{ux}\|h(PW_{ux}^{new}\|N_{ux}))$ and $X_{ux}^{new} = X_{ux} \oplus h(PW_{ux}\|ID_{ux}\|N_{ux}) \oplus h(PW_{ux}^{new}\|ID_{ux}\|N_{ux}^{new})$ Finally, SC_{ux} replaces R_{ux} and X_{ux} by R_{ux}^{new} and X_{ux}^{new}.

6 Security analysis

The formal security analysis followed by security discussion is performed in this section. Further, protocol verification thorough automated tool ProVerif is also substantiated here.

6.1 Formal security

To demonstrate formally, that proposed scheme is secure, we adopted the same analysis as mentioned in [46, 48]. Following oracles are defined for analysis purpose:

- ***Reveal:*** This oracle unconditionally outputs a string S from the one way hash function $R = h(S)$.
- ***Extract:*** This oracle unconditionally outputs the scalar multiplier k out of a given elliptic curve points $O = kP$ and P.

Theorem 1 *The proposed biometric based multi server authentication scheme is secure for an attacker $\mathcal{A}$ to stanch $\mathcal{U}_x$'s identity (ID_{ux}), the parameter K, the session key SK_{xy} and the shared key PSK_{rs} between RC and $\mathcal{S}_y$ considering one way hash function as random oracle and under the hardness assumption of $ECDLP$.*

Proof Let $\mathcal{A}$ be an adversary having capabilities to compute $\mathcal{U}_x$'s ID_{ux}, the secret session parameter K the session key SK_{xy} and the shared key PSK_{rs} between RC and $\mathcal{S}_y$. $\mathcal{A}$ simulates both oracles *Reveal* and *Extract* to run the algorithmic experiment $EXPE1_{\mathcal{A},TFBAMS}^{HASH,ECDLP}$ against our proposed three factor biometric based authentication scheme for multi server environments ($TFBAMS$). The success probability for the mentioned experiment is defined as $Succe_1 = |Prb[EXPE1_{\mathcal{A},TFBAMS}^{HASH,ECDLP} = 1] - 1|$. $\mathcal{A}$'s advantage is solicited as $Advt1_{\mathcal{A},TFBAMS}^{HASH,ECDLP}(t, q_{rev}, q_{ext}) = max_{\mathcal{A}}(Succe_1)$, where $\mathcal{A}$ is allowed to make at maximum q_{rev} *Reveal* and q_{ext} *Extract* queries. Referring to the experiment $\mathcal{A}$ can compute ID_{ux}, K, SK_{xy} and PSK_{rs}, if he can (i) invert the hash function and (ii) solve the ECDLP. However, referring to Definition 1 it is computationally infeasible to invert a secure one way hash function, similarly by Definition 2 it is computationally infeasible to solve ECDLP. Hence, we have $Advt1_{\mathcal{A},TFBAMS}^{HASH,ECDLP}(t, q_{rev}, q_{ext}) \leq \epsilon$. Therefore, proposed three factor biometric bases authentication scheme for multi server environments is secure against an adversary $\mathcal{A}$ to computes $\mathcal{U}_x$'s ID_{ux}, the secret session parameter K the session key SK_{xy} and the shared key PSK_{rs} between RC and $\mathcal{S}_y$. □

Theorem 2 *The proposed biometric based multi server authentication scheme is secure for an attacker $\mathcal{A}$ to stanch $\mathcal{U}_x$'s biometrics $H(BIO_{ux})$, identity (ID_{ux}), password PW_{ux} and the security parameter $h(PSK_{rs}\|ID_{ux})$ considering one way hash function as random oracle for the stolen smart card attack.*

Proof Let $\mathcal{A}$ be an adversary having capabilities to stanch $\mathcal{U}_x$'s biometrics $H(BIO_{ux})$, identity (ID_{ux}), password PW_{ux} and the security parameter $h(PSK_{rs}\|ID_{ux})$ out of a stolen smart card. $\mathcal{A}$ simulates *Reveal* oracle to run the algorithmic experiment $EXPE2^{HASH}_{\mathcal{A},TFBAMS}$ against our proposed three factor biometric bases authentication scheme for multi server environments ($TFBAMS$). The success probability for the mentioned experiment is defined as $Succe_2 = |Prb[EXPE2^{HASH}_{\mathcal{A},TFBAMS} = 1] - 1|$. $\mathcal{A}$'s advantage is solicited as $Advt2^{HASH}_{\mathcal{A},TFBAMS}(t, q_{rev} = max_{\mathcal{A}}(Succe_2)$, where $\mathcal{A}$ is allowed to make at maximum q_{rev} *Reveal* queries. Referring to the experiment, $\mathcal{A}$ can compute $H(BIO_{ux})$, ID_{ux}, PW_{ux} and PSK_{rs}, if he can invert the hash function. However, referring to Definition 1 it is computationally infeasible to invert a secure one way hash function. Hence, we have $Advt2^{HASH}_{\mathcal{A},TFBAMS}(t, q_{rev}) \leq \epsilon$. Therefore, proposed three factor biometric bases authentication scheme for multi server environments is secure against an adversary $\mathcal{A}$ to computes $\mathcal{U}_x$'s biometrics $H(BIO_{ux})$, identity (ID_{ux}), password PW_{ux} and the security parameter $h(PSK_{rs}\|ID_{ux})$ out of a stolen smart card. □

6.2 Further security discussion

In this subsection, we informally describes the security functionalities provided by proposed scheme.Table 2 illustrates a security comparison of proposed scheme with related existing schemes [8, 41, 42, 46].

Algorithm 1 $EXPE1^{HASH,ECDLP}_{\mathcal{A},TFBAMS}$

```
Eavesdrop the login message $Z_{ux}, M_1, M_2, M_3, T_1$, Where $M_1 = r_{ux}.P$, $M_2 = ID_{ux} \oplus K$, $M_3 = n_{ux} \oplus h(h(PSK_{rs}\|ID_u x)\|SID_{sy})$ and $Z_{ux} = h(h(PSK_{rs}\|ID_{ux})\|n_{ux}\|K\|T_1)$
Call Extract oracle on $M_1$ and $P$ to obtain $r'_{ux} \leftarrow Extract(M_1, P)$
Compute $K' = r_{ux} \oplus Pub_{sy}$ and $ID'_{ux} = K' \oplus M_2$
Call Reveal on $Z_{ux}$ to get $h(PSK_{rs}\|ID_{ux})'\|n'_{ux}\|K''\|T'_1) \leftarrow Reveal(Z_{ux})$
if $(K'' = K')$ then
    Call Reveal on $h(PSK_{rs}\|ID_{ux})'$ and get $(PSK'_{rs}\|ID''_{ux}) \leftarrow Reveal(h(PSK_{rs}\|ID_{ux})')$
    if $(ID'_{ux} = ID''_{ux})$ then
        Accept $ID'_{ux}$ and $PSK'_{rs}$ along with session specific parameters $n'_{ux}$ and $K'$
        Eavesdrop challenge message $M_4, M_5, T_2$, where $M_4 = n_{sy} \oplus K$ and $M_5 = h(ID_{ux}\|n_{ux}\|n_{sy}\|K\|T_2)$
        Compute $n'_{sy} = M'_4 oplus K'$ and $SK'_{xy} = h(ID'_{ux}\|n'_{ux}\|n'_{sy}\|K)$
        Eavesdrop response message $M_6, T_3$
        Compute $M'_6 = h(SK'_{xy}\|ID'_{ux}\|n'_{sy}\|T_3)$
        if $(M'_6 = M_6)$ then
            Accept $SK'_{xy}$
        else
            return Fail
        end if
    else
        return Fail
    end if
else
    return Fail
end if
```

Algorithm 2 $EXPE2^{HASH}_{\mathcal{A},TFBAMS}$

```
Extract the parameters $Y_{ux}, V_{ux}$ from stolen smart card using the methods mentioned in [33, 47] Where $Y_{ux} = h(PSK_{rs}\|ID_{ux}) \oplus h(PW_{ux}\|ID_{ux}\|H(BIO_{ux}))$ and $V_{ux} = h(ID_{ux}\|h(PW_{ux}\|H(BIO_{ux})))$
Call Reveal oracle on $V_{ux}$ and obtain $(ID'_{ux}\|h(PW_{ux}\|H(BIO_{ux}))') \leftarrow Reveal(V_{ux})$
Call Reveal on $h(PW_{ux}\|H(BIO_{ux}))'$ to get $(PW'_{ux}\|H(BIO_{ux})') \leftarrow Reveal(h(PW_{ux}\|H(BIO_{ux}))')$
Compute $W = h(PW'_{ux}\|ID'_{ux}\|H(BIO_{ux})')$ and $T = Y_{ux} \oplus W = \overline{h(PSK_{rs}\|ID_{ux})}$
Call Reveal on $T$ and obtain $(PSK'_{rs}\|ID''_{ux}) \leftarrow Reveal(T)$
if $(ID''_{ux} = ID'_{ux})$ then
    Accept $PSK_{rs}$, $PW'_{ux}$ and $H(BIO_{ux})'$
else
    return Fail
end if
```

6.2.1 *Anonymity and privacy*

In our proposed biometric scheme the user $\mathcal{U}_x$'s identity ID_{ux} is not sent over public network rather M_1 and M_2 are sent to $\mathcal{S}_y$. These two parameters are freshly generated for each session. The anonymity can only be broken if an adversary can compute K, but it can be seen that K can be computed only be the use of $\mathcal{S}_y$'s private key. Hence, proposed scheme preserves anonymity and untraceability.

6.2.2 *Mutual authentication*

$\mathcal{S}_y$ authenticates $\mathcal{U}_x$ by checking $Z_{ux} \overset{?}{=} h(h(PSK_{rs}\|ID_{ux})\|n_{ux}\|K\|T_1)$. Computation of Z_{ux} involves $h(PSK_{rs}\|ID_{ux})$ which requires the smart card as well as password PW_{ux} and the biometrics BIO_{ux} of $\mathcal{U}_x$. Therefore to deceive $\mathcal{S}_y$, the adversary needs $\mathcal{U}_x$'s password, biometric as well as his smart card. Likewise, $\mathcal{U}_x$ authenticates $\mathcal{S}_y$ by checking $M_5 \overset{?}{=} h(ID_{ux}\|n_{ux}\|n_{sy}\|K\|T_2)$ which requires the computation of $\mathcal{U}_x$'s identity ID_{ux}, the session parameter n_{ux} and K. ID_{ux} and K can be computed only by using $\mathcal{S}_y$'s private key as mentioned in Subsection 6.2.1, while n_{ux} can be computed by using

Table 2 Comparison of security parameters

Scheme:	Proposed	[42]	[41]	[46]	[8]
Anonymity and privacy	Yes	Yes	No	Yes	Yes
Mutual Authentication and key agreement	Yes	Yes	Yes	Yes	Yes
Resists Impersonation attack	Yes	No	No	No	No
Resists Smart card theft attack	Yes	Yes	Yes	Yes	No
Resists Replay attack	Yes	Yes	Yes	Yes	Yes
Provides Forward secrecy	Yes	Yes	Yes	No	Yes
Resists Insider and Stolen verifier attacks	Yes	Yes	Yes	Yes	Yes
Resists password guessing attack	Yes	Yes	Yes	Yes	Yes
Provided No clock synchronization	Yes	Yes	Yes	Yes	Yes

$h(h(PSK_{rs}\|ID_{ux})\|SID_{sy})$ which requires the shared secret key between $\mathcal{S}_y$ and RC. So in order to deceive $\mathcal{U}_x$, the adversary needs $\mathcal{S}_y$'s private key Pri_{sy} as well as the shared key $h(PSK_{rs})$ between $\mathcal{S}_y$ and RC. Hence only legal user can pass authentication test from server and vice versa. Therefore, proposed scheme provides proper mutual authentication.

6.2.3 User and server impersonation attacks

Only legal user can generate legal authentication request message $\{Z_{ux}, M_1, M_2, M_3, T_1\}$ and response message $\{M_6, T_3\}$, similarly only legal server can respond with challenge message $\{M_4, M_5, T_2\}$ as proved in Subsection 6.2.2. Hence, user and server impersonation attacks are not feasible on proposed scheme.

6.2.4 Smart card theft/stolen attack

Let us assume, the adversary by using some means becomes able to acquire $\mathcal{U}_x$'s smart card. The adversary further extracts the parameters $V_{ux} = h(ID_{ux}\|h(PW_{ux}\|H(BIO_{ux})))$, $Y_{ux} = h(PSK_{rs}\|ID_{ux}) \oplus h(PW_{ux}\|ID_{ux}\|H(BIO_{ux}))$ and $h(.)$. Then to compute the secret parameter $h(PSK_{rs}\|ID_{ux})$, the adversary needs PW_{ux} and BIO_{ux}. Hence, the stolen smart card will not benefit the adversary for forgery.

6.2.5 Replay attack

If some adversary after intercepting the login request message $\{Z_{ux}, M_1, M_2, M_3, T_1\}$, replays it later on. The server $\mathcal{S}_y$ after receiving the message will check the freshness of time stamp T_1, as the time stamp is old dated, $\mathcal{S}_y$ will simply discard the message. Therefore, replay attack is not viable on proposed scheme.

6.2.6 Perfect forward secrecy

The computed session key between $\mathcal{S}_y$ and $\mathcal{U}_x$ contains share (n_{sy}, n_{ux}) from both the participants respectively. So even if the long term private key of $\mathcal{S}_y$ or $\mathcal{U}_x$'s password is revealed to the attacker it will not benefit to compute previous session keys. Therefore, proposed scheme possesses perfect forward secrecy.

6.2.7 Insider and stolen verifier and attacks

For the proposed scheme, $\mathcal{S}_y$ does not store any parameter related to $\mathcal{U}_x$'s password (PW_{ux}) or his biometrics (BIO_{ux}), as there is no verifier table so no stolen verifier attack is possible. Likewise, $\mathcal{U}_x$ does not send his password (PW_{ux}) or his biometrics BIO_{ux} in plain text. Hence, no insider will have any advantage to expose his password or biometrics.

6.2.8 Password guessing attack

For the proposed scheme, the information relating to $\mathcal{U}_x$'s password is protected by his identity ID_{ux}, BioHashed biometrics $H(BIO_{ux})$ further it is XORed with $h(PSK_{rs}\|ID_{ux})$. Moreover, there is no parameter stored in smart card to check the validity of guessed password by adversary. Hence no offline password guessing attack is feasible on proposed scheme. Likewise, the system incorporates built in maximum number of login requests, which ensures no online password guessing attack.

7 Verification through ProVerif

The purpose of verification tools for cryptographic protocols is to confirm the robustness of the protocols against active and passive adversaries having some knowledge of the cryptographic parameters. ProVerif is an applied π calculus based automated verification tool to validate the security of cryptographic protocols against knowledgeable attackers. ProVerif can prove a number of security properties like: reachability, secrecy, authentication and so on. [4, 10, 51]. We have implemented the login and authentication steps of the proposed protocol as illustrated in Fig. 3 and explained in Subsection 5.3. The formal verification model of ProVerif consists of following three parts. (1) Declaration is used for defining names, constants, variables and cryptographic operations. We have shown declaration part in Fig. 4a. (2) Process part is reserved for defining processes involved in protocol execution. As illustrated in Fig. 4b we have defined two processes: server process *(ServerSy)* and user process *(UserUx)*. (3) Main part simulates the protocol execution, as shown in Fig. 4c, we simulate parallel execution of two processes along with definition of two events to verify reach-ability property. Finally, we simulate three queries. The results are as follows:

1. RESULT inj-event(end_Serversy(id)) ==> inj-event(begin_Serversy(id)) is true.
2. RESULT inj-event(end_Userux($\text{id}_1$114)) ==> inj-event(begin_Userux($\text{id}_1$114)) is true.
3. RESULT not attacker(SKxy[]) is true.

The results (1) and (2) validates that both user and server processes started and terminated normally, which confirms the correctness and reach-ability properties. While (3) verifies that the session key *(SKxy[])* is not exposed to adversary. Hence Proposed protocol possesses reach-ability as well as secrecy and authentication properties.

8 Performance comparisons

This section presents performance assessment of the proposed scheme against two Lu et al.'s pertinent schemes. Recently, Lu et al. presented two schemes based on biometrics for multi-server environments and professed that their schemes provide security against the known threats. This paper suggests that Lu et al.'s schemes do not provide invincibility against few known attacks. The first scheme of Lu et al fails retaliate against user anonymity violation and impersonation attacks, whereas their second scheme is vulnerable against impersonation attack. The the proposed scheme's performance is equated with both the schemes of Lu et al. in Table 3. Following Notations are used for computation cost analysis:

- T_{Oh} refers to accumulated execution time of one-way hash operation.
- T_{Re} refers to accumulated execution time of RSA encryption.
- T_{Rd} refers to accumulated execution time of RSA decryption.
- T_{Epm} refers to elliptic curve point multiplication.

As per Kilinc and Yanik [34] experiment on a personal computer involving a processor with Dual CPU E2200 2.20 GHz along with RAM size of 2048MB, the computation cost for T_{Oh} is approximately $0.0023ms$, T_{Re} is $3.8500ms$, T_{Rd} is $0.1925ms$ and T_{Epm} is $2.229ms$. Kilinc and Yanik [34] experiment was performed on the Ubuntu Operating system and using PBC Library.

The comparison presented in Table 3 reveals that the proposed scheme is computationally inexpensive than scheme in [42]. While the proposed scheme is quite expensive than rest of

```
(*************** Channels **************)
free Ch_Pub:channel.
(********** Names & Variables **********)
free IDux:bitstring.
free PWux:bitstring.
free Yux:bitstring.
free BIOux:bitstring [private].
free Vux:bitstring [private].
const P: bitstring.
free Pubsy:bitstring.
free PSKrs:bitstring [private].
free SIDsy:bitstring [private].
(**Constructors*destructors*Equations **)
fun h(bitstring):bitstring.
fun H(bitstring):bitstring.
fun mult(bitstring,bitstring):bitstring.
fun concat(bitstring,bitstring):bitstring.
fun xor(bitstring,bitstring):bitstring.
equation forall a:bitstring,b:bitstring;
 xor(xor(a,b),b)=a.
```

(a) Declarations

```
(***************processes **************)
(****************User ux****************)
let UserUx =
(*Login and Authentication Phase*)
let Vux'=h(concat(IDux,h(concat(PWux,H(
    BIOux))))) in
new rux:bitstring;
let K=mult(rux,Pubsy) in
let M1=mult(rux,P) in
let M2=xor(IDux,K) in
new nux:bitstring;
new T1:bitstring;
let M3=xor(nux,h(xor(Yux , concat(h(concat(
    PWux,(IDux,H(BIOux)))),SIDsy)))) in
let Zux=h(concat(h(concat(PSKrs,IDux)),(nux
    ,K,T1))) in
out(Ch_Pub,(Zux,M1,M2,M3,T1));
in(Ch_Pub,(xM4:bitstring,xM5:bitstring,xT2:
    bitstring));
new T2:bitstring;
let nsy=xor(xM4,K) in
let M5=h(concat(IDux,(nux,nsy,K,T2))) in
if(M5=xM5) then
let SKxy=h(concat(IDux,(nux,nsy,K))) in
new T3:bitstring;
let M6=h(concat(SKxy,(IDux,nsy,T3))) in
out(Ch_Pub,(M6,T3))
else 0.
(************** Server Sy***************)
let ServerSy=
new Prisy:bitstring;
let Pubsy=mult(Prisy,P) in
in(Ch_Pub,(xZux:bitstring,xM1:bitstring,xM2
    :bitstring,xM3:bitstring,xT1:bitstring)
    );
new T1:bitstring;
let K=mult(xM1,Prisy) in
let IDux'=xor(xM2,K) in
let nux=xor(xM3,h(concat(h(concat(PSKrs,
    IDux')),SIDsy))) in
let Zux=h(concat(h(concat(PSKrs,IDux')),(
    nux,K,T1))) in
if (Zux=xZux) then
new nsy:bitstring;
new T2:bitstring;
let M4=xor(nsy,K) in
let M5=h(concat(IDux',(nux,nsy,K,T2))) in
let SKxy=h(concat(IDux',(nux,nsy,K))) in
out(Ch_Pub,(M4,M5,T2));
in(Ch_Pub,(xM6:bitstring,xT3:bitstring));
new T3:bitstring;
let M6=h(concat(SKxy,(IDux',nsy,T3))) in
if(M6=xM6) then 0.
```

(b) Processes

```
(****************Events ****************)
event begin_Userux(bitstring).
event end_Userux(bitstring).
event begin_Serversy(bitstring).
event end_Serversy(bitstring).
(********Process Replication************)
process (  (!ServerSy) |  (!UserUx) )
(************** *queries* **************)
free SKxy:bitstring [private].
query attacker(SKxy).
query id:bitstring; inj event(end_Userux(id
    )) ==> inj event(begin_Userux(id)) .
query id:bitstring; inj event(end_Serversy(
    id)) ==> inj event(begin_Serversy(id))
    .
```

(c) Main

Fig. 4 ProVerif code

Table 3 Computation cost comparison

Scheme	User side	server Side	Total execution time
Chuang et al. [8]	$8T_{Oh}$	$8T_{Oh}$	$16T_{Oh} \approx 0.0368$
Mishra et al. [46]	$10T_{Oh}$	$7T_{Oh}$	$17T_{Oh} \approx 0.0391$
Lu et al. [41]	$9T_{Oh}$	$8T_{Oh}$	$17T_{Oh} \approx 0.0391ms$
Lu et al. [42]	$8T_{Oh} + 3T_{Re}$	$8T_{Oh} + 3T_{Rd}$	$16T_{Oh} + 3T_{Re} + 3T_{Rd} \approx 12.1643ms$
Proposed scheme	$9T_{Oh} + 2T_{Epm}$	$7T_{Oh} + 1T_{Epm}$	$16T_{Oh} + 3T_{Epm} \approx 6.7148ms$

the schemes [8, 41, 46]. Moreover proposed scheme provide invincibility against the known threats. It is further declared that only the proposed scheme resists the known attacks, while rest of the competing schemes [8, 41, 42, 46] are vulnerable to impersonation and/or other related attacks.

9 Conclusion

In this paper, we have cryptanalyzed two most recent biometric based multi factor authentication schemes proposed by Lu et al. We have proved both of their schemes to be vulnerable to impersonation attacks, additionally we have also showed that one of their scheme is also vulnerable to anonymity violation attack. Then we proposed an improved biometric based multi factor authentication scheme. The proposed scheme is proved to be robust against all known attacks. We have substantiated the security of proposed scheme using famous automated security validation tool ProVerif.

Acknowledgments Author would like to thank Prof. Muhammad Arshad Zia, Mr, Shahzad Saddique Chaudhry, the anonymous reviewers and the editor for their valuable suggestions to improve the quality, correctness, presentation and readability of the manuscript.

References

1. Awasthi AK, Srivastava K (2013) A biometric authentication scheme for telecare medicine information systems with nonce. J Med Syst 37(5):1–4
2. Belguechi R, Rosenberger C, Ait-Aoudia S (2010) Biohashing for securing minutiae template. In: 20th International Conference on Pattern Recognition (ICPR), 2010. IEEE, pp 1168–1171
3. Chaudhry S, Naqvi H, Shon T, Sher M, Farash M (2015) Cryptanalysis and improvement of an improved two factor authentication protocol for telecare medical information systems. J Med Syst 39(6):66. doi:10.1007/s10916-015-0244-0
4. Chaudhry SA, Farash MS, Naqvi H, Kumari S, Khan MK (2015) An enhanced privacy preserving remote user authentication scheme with provable security. Secur Commun Netw 1–13. doi:10.1002/sec.1299
5. Chaudhry SA, Mahmood K, Naqvi H, Sher M (2015) A secure authentication scheme for session initiation protocol based on elliptic curve cryptography. In: The 13th IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC 2015). IEEE, pp 1–5
6. Chaudhry SA, Mahmood K, Naqvi H, Khan MK (2015) An improved and secure biometric authentication scheme for telecare medicine information systems based on elliptic curve cryptography. Journal of Medical Systems 66. doi:10.1007/s10916-015-0335-y
7. Chaudhry SA, Naqvi H, Sher M, Farash MS, Hassan M (2015) An improved and provably secure privacy preserving authentication protocol for SIP. Peer-to-Peer Netw Appl. doi:10.1007/s12083-015-0400-9
8. Chuang MC, Chen MC (2014) An anonymous multi-server authenticated key agreement scheme based on trust computing using smart cards and biometrics. Expert Syst Appl 41(4):1411–1418
9. Cao X, Zhong S (2006) Breaking a remote user authentication scheme for multi-server architecture. IEEE Commun Lett 10(8):580–581. doi:10.1109/LCOMM.2006.1665116
10. Chaudhry S, Farash M, Naqvi H, Sher M (2015) A secure and efficient authenticated encryption for electronic payment systems using elliptic curve cryptography. Electron Commer Res:1–27. doi:10.1007/s10660-015-9192-5
11. Das AK (2015) A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor. Int J Commun Syst. doi:10.1002/dac.2933
12. Dolev D, Yao AC (1983) On the security of public key protocols. IEEE Trans Inf Theory 29(2):198–208. doi:10.1109/TIT.1983.1056650
13. 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, DOI doi:10.1007/978-3-540-85174-5

14. Farash MS, Attari MA (2014) A secure and efficient identity-based authenticated key exchange protocol for mobile client–server networks. J Supercomput 69(1):395–411
15. Farash MS, Attari MA (2014) An anonymous and untraceable password-based authentication scheme for session initiation protocol using smart cards. Int J Commun Syst. doi:10.1002/dac.2848
16. Farash MS, Attari MA (2014) Cryptanalysis and improvement of a chaotic map-based key agreement protocol using chebyshev sequence membership testing. Nonlinear Dyn 76(2):1203–1213
17. He D, Zeadally S (2015) Authentication protocol for an ambient assisted living system. IEEE Commun Mag 53(1):71–77
18. He D, Kumar N, Chen J, Lee CC, Chilamkurti N, Yeo SS (2013) Robust anonymous authentication protocol for health-care applications using wireless medical sensor networks. Multimed Syst 21(1):49–60
19. He D (2012) An efficient remote user authentication and key agreement protocol for mobile client–server environment from pairings. Ad Hoc Netw 10(6):1009–1016
20. He D, Kumar N, Chilamkurti N (2015) A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks. Inf Sci 321:263–277. doi:10.1016/j.ins.2015.02.010
21. He D, Wang D (2015) Robust biometrics-based authentication scheme for multi server environment. IEEE Syst J 9(3):816–823
22. Heydari M, Sadough SMS, Farash MS, Chaudhry SA, Mahmood K (2015) A secure and efficient authenti-cated encryption for electronic payment systems using elliptic curve cryptography. Wirel Person Comm 2015. doi:10.1007/s11277-015-3123-6
23. He D, Kumar N, Lee JH, Sherratt R (2014) Enhanced three-factor security protocol for consumer usb mass storage devices. IEEE Trans Consum Electron 60(1):30–37. doi:10.1109/TCE.2014.6780922
24. Irshad A, Sher M, Faisal MS, Ghani A, Hassan M, Ch SA (2013) A secure authentication scheme for session initiation protocol by using ecc on the basis of the Tang and Liu scheme. Secur Commun Netw 7(8):1210–1218. doi:10.1002/sec.834
25. Irshad A, Sher M, Rehman E, Ch SA, Hassan M, Ghani A (2014) A single round-trip sip authentication scheme for voice over internet protocol using smart card. Multimed Tools Appl 74(11):3967–3984. doi:10.1007/s11042-013-1807-z
26. Islam S, Khan M (2014) Cryptanalysis and improvement of authentication and key agreement protocols for telecare medicine information systems. J Med Syst 38(10):135. doi:10.1007/s10916-014-0135-9
27. Islam SH (2015) Design and analysis of a three party password-based authenticated key exchange protocol using extended chaotic maps. Inf Sci 312:104–130
28. Islam SH (2014) Provably secure dynamic identity-based three-factor password authentication scheme using extended chaotic maps. Nonlinear Dyn 78(3):2261–2276
29. Islam SH (2014) A provably secure id-based mutual authentication and key agreement scheme for mobile multi-server environment without esl attack. Wirel Person Commun 79(3):1975–1991
30. Islam S, Khan MK (2014) Provably secure and pairing-free identity-based handover authentication protocol for wireless mobile networks. Int J Commun Syst. doi:10.1002/dac.2847
31. Jiang Q, Ma J, Tian Y (2014) Cryptanalysis of smart-card-based password authenticated key agreement protocol for session initiation protocol of zhang et al. Int J Commun Syst. doi:10.1002/dac.2767
32. Jin ATB, Ling DNC, Goh Ax (2004) Biohashing: two factor authentication featuring fingerprint data and tokenised random number. Pattern Recogn 37(11):2245–2255
33. Kocher P, Jaffe J, Jun B (1999) Differential power analysis. Advances in Cryptology CRYPTO 99, Springer, pp 388–397
34. Kilinc HH, Yanik T (2014) A survey of sip authentication and key agreement schemes. IEEE Commun Surv Tutorials 16(2):1005–1023
35. Lamport L (1981) Password authentication with insecure communication. Commun ACM 24(11):770–772
36. Lu R, Lin X, Liang X, Shen X. (2012) A dynamic privacy-preserving key management scheme for location-based services in vanets. IEEE Trans Intell Trans Syst 13(1):127–139
37. Lu Y, Li L, Yang Y (2015) Robust and efficient authentication scheme for session initiation protocol. Math Probl Eng. doi:10.1155/2015/894549
38. Lu Y, Li L, Peng H, Yang Y (2015) An enhanced biometric-based authentication scheme for telecare medicine information systems using elliptic curve cryptosystem. J Med Syst 39(3):1–8
39. Li X, Niu J, Khan MK, Liao J, Zhao X (2014) Robust three-factor remote user authentication scheme with key agreement for multimedia systems. Secur Comm Netw. doi:10.1002/sec.961
40. Li X, Khan M, Kumari S, Liao J, Liang W (2014) Cryptanalysis of a robust smart card authentication scheme for multi-server architecture. In: International Symposium on Biometrics and Security Technologies (ISBAST), 2014, pp 120–123. doi:10.1109/ISBAST.2014.7013106

41. Lu Y, Li L, Yang X, Yang Y (2015) Robust biometrics based authentication and key agreement scheme for multi-server environments using smart cards. PloS ONE 10(5). doi:10.1371/journal.pone.0126323
42. Lu Y, Li L, Peng H, Yang Y (2015) A biometrics and smart cards-based authentication scheme for multi-server environments. Secur Commun Netw 1–10. doi:10.1002/sec.1246
43. Lumini A, Nanni L (2007) An improved biohashing for human authentication. Pattern Recogn 40(3):1057–1065
44. Mehmood Z, uddin N, Ch SA, Nasar W, Ghani A (2012) An efficient key agreement with rekeying for secured body sensor networks. In: Second International Conference on Digital Information Processing and Communications (ICDIPC), 2012. IEEE, pp 164–167
45. Mishra D, Kumari S, Khan MK, Mukhopadhyay S (2015) An anonymous biometric-based remote user authenticated key agreement scheme for multimedia systems. Int J Commun Syst. doi:10.1002/dac.2946
46. Mishra D, Das AK, Mukhopadhyay S (2014) A secure user anonymity-preserving biometric-based multi-server authenticated key agreement scheme using smart cards. Expert Syst Appl 41(18):8129–8143
47. Messerges TS, Dabbish EA, Sloan RH (2002) Examining smart-card security under the threat of power analysis attacks. IEEE Trans Comput 51(5):541–552
48. Mir O, Nikooghadam M (2015) A secure biometrics based authentication with key agreement scheme in telemedicine networks for e-health services. Wirel Person Comm 83(4):2439–2461
49. Sun DZ, Huai JP, Sun JZ, Li JX, Zhang JW, Feng ZY (2009) Improvements of juang's password-authenticated key agreement scheme using smart cards. IEEE Trans Indust Electron 56(6):2284–2291
50. Ul Amin N, Asad M, Din N, Ch SA (2012) An authenticated key agreement with rekeying for secured body sensor networks based on hybrid cryptosystem. In: 9th IEEE International Conference on Networking, Sensing and Control (ICNSC), 2012. IEEE, pp 118–121
51. Xie Q, Dong N, Wong DS, Hu B (2014) Cryptanalysis and security enhancement of a robust two-factor authentication and key agreement protocol. Int J Commun Syst. doi:10.1002/dac.2858
52. Zhao D, Peng H, Li L, Yang Y (2014) A secure and effective anonymous authentication scheme for roaming service in global mobility networks. Wirel Person Commun 78(1):247–269
53. Zhang L, Tang S, Cai Z (2014) Robust and efficient password authenticated key agreement with user anonymity for session initiation protocol-based communications. IET Commun 8(1):83–91
54. Zhang M, Zhang J, Zhang Y (2015) Remote three-factor authentication scheme based on fuzzy extractors. Secur Commun Netw 8(4):682–693. doi:10.1002/sec.1016

Shehzad Ashraf Chaudhry received his MS Computer Science with distinction, from International Islamic University Islamabad, Pakistan in 2009 and was awarded Gold Medal. Currently he is working as Lecturer at the Department of Computer Science & Software Engineering, International Islamic University, Islamabad. He authored more than 25 scientific publications published in different international journals and proceedings including 17 publication in SCI/E journals. His research interests include Lightweight Cryptography, Elliptic/Hyper Elliptic Curve Cryptography, Multimedia Security, E- Payment systems, MANETs, SIP authentication, IP Multimedia sub-system and Next Generation Networks.