Wireless Pers Commun (2017) 95:3185–3208
DOI 10.1007/s11277-017-3990-0

An Improved and Secure Chaotic-Map Based Multi-server Authentication Protocol Based on Lu et al. and Tsai and Lo's Scheme

Azeem Irshad[1] · Muhammad Sher[1] · Muhammad Usman Ashraf[1,6] · Bander A. Alzahrani[3] · Fan Wu[2] · Qi Xie[4] · Saru Kumari[5]

Published online: 7 February 2017

Abstract The simple password based authentication techniques have been evolving into more secure and advanced protocols, capable of countering the advanced breed of threats. Following this development, the multi-server authentication (MSA), lets subscribers the provision of services from various service providers out of a single registration performed initially. The user seeks to register from registration centre first, and could avail a range of services onwards. The research efforts on MSA based framework, for making it light-weight and security resilient, has been going on a reasonable pace. However, yet we have not come up with a framework that can be relied upon for deployment in an access network bearing nodes that demand low computational cost. Recently, in this regard, Tsai and Lo

✉ Azeem Irshad
irshadazeem2@gmail.com

Muhammad Sher
m.sher@iiu.edu.pk

Muhammad Usman Ashraf
usmanashraf@iiu.edu.pk

Bander A. Alzahrani
baalzahrani@kau.edu.sa

Fan Wu
conjurer1981@gmail.com

Qi Xie
qixie68@126.com

Saru Kumari
saryusiirohi@gmail.com

[1] Computer Science Department, International Islamic University, Islamabad, Pakistan

[2] Department of Computer Science and Engineering, Xiamen Institute of Technology, Xiamen 361021, China

[3] College of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

presented a chaotic map-based multi-server authentication protocol. However, the Tsai and Lo scheme is found vulnerable to key-compromise impersonation attack, Bergamo et al. and password guessing attack by Lu et al. In return, Lu et al. presented a model countering the flaws of Tsai and Lo scheme. We review both schemes and found that Tsai et al. is still vulnerable to more threats, and at the same time, we demonstrate that Lu et al. is also vulnerable to RC-spoofing attack, replay attack, anonymity failure and bears some technical flaws. In this paper, we propose a secure and efficient scheme improved upon Tsai et al. protocol. Besides, this study work presents the formal security analysis using BAN logic and performance efficiency has also been evaluated against contemporary protocols.

Keywords Multi-server authentication · Remote authentication · Attacks · Chebyshev chaotic map

1 Introduction

The multi-server authentication framework relieves the subscriber of multiple registrations, every time it requires a service from a service provider, yielding overhead efficiency. The objective for multi-server paradigm was to ease the user of the maintenance of as many password verifiers as the number of servers. This, positively, reduces the cost when a user needs to access multiple services from different servers in a network. The remote authentication often covers such kind of multi-server authentications, which further stresses the efficiency and strength of these protocols. All of the servers in a network rely on a single registration of a central Registration Centre for verifying the authenticity of a user. Those servers consult this online Registration Centre, on the receiving of authorization request of any user, in most of the techniques. This is not only beneficial for the user who has to maintain the same password and parameters for all servers, but also for servers who would forego the need to register and maintain the identities of different subscribers individually.

The authentication can be roughly categorized into password based protocols [1–8] and smart card based protocols [9–27]. The low entropy password based protocols, lets the subscribers log into servers conveniently. However, this convenience comes at a cost of attacks on the protocol. The smart card was introduced afterwards as a two-factor security to add a further security dimension to the authentication protocols, so that it may remember high entropy secrets to strengthen security. Despite the obvious shortcoming that a user cannot avail the services of a server without this smart card, the use of smart card has now become a crucial component of the authentication protocol framework. In the last decade several multi-server authentication techniques has been presented. However there is still a need of more efficient and robust techniques. In this regard the first chaotic map based single server protocol was presented by Xiao et al. [3] in 2007, in which Han [2] found few

[4] Hangzhou Key Laboratory of Cryptography and Network Security, Hangzhou Normal University, Hangzhou, China

[5] Department of Mathematics, Chaudhary Charan Singh University, Meerut, Uttar Pradesh 250004, India

[6] IBMS, University of Agriculture, Faisalabad, Faisalabad, Pakistan

weaknesses afterwards. Then, many single-server based protocols, in this connection, had been presented to counter the discovered day-to-day threats [1–8]. However, these single-server schemes might lead to added outlay in the cost, if applied in a scenario where a subscriber needs multiplicity of services. To meet the objectives, in this connection, Lin et al. [29] contributed about the concept of multi-server authentication in 2001. Although, that preliminary effort was supposed to be complex and computation intensive, that work turned out to be the basis of further development in the MSA domain. Afterwards, several MSA based protocols [28–35, 37, 38] have been presented for the cause of promoting efficiency and improving defense. Recently, Tsai and Lo has [36] has presented chaotic-map based multiserver authentication protocol. However, the scheme has been found to be vulnerable to password guessing attack and key-compromise information attack [39]. In this study we demonstrate that the Tsai et al. scheme is also vulnerable to server spoofing attack and stolen smart card attacks besides other attacks as demonstrated in [39]. Lu et al. [39] also presented an improved scheme, however the Lu et al.'s protocol seems to have a technical flaw for its practical implementation. The Lu et al. is also vulnerable to RC-spoofing attack, server spoofing attack, replay attack and lacks anonymity. In this study work we present a review on Tsai and Lo scheme, and Lu et al. schemes and then propose an efficient secure protocol improved upon the Tsai and Lo's scheme. Besides, this paper also demonstrates performance evaluation and formal security analysis.

The Sect. 2 relates to some preliminaries, describing chebyshev map and hash function. The Sects. 3 and 4 describe the working and review analysis of Tsai and Lo's, and Lu et al.'s scheme respectively. Section 5 presents our proposed model. Sections 6 and 7 demonstrate security analysis and performance evaluation. The last section presents the conclusion.

2 Preliminaries

This section presents some preliminary properties for Chebyshev chaotic maps and hash function.

2.1 Chebyshev Chaotic Maps

Some properties of the Chebyshev chaotic maps [22] and Chebyshev polynomial are illustrated as under:

1. We can assume α as a number and variable ρ ranging from $[-1, 1]$. Next, we express the Chebyshev polynomial i.e., $T_\alpha(\rho)$: $[-1, 1] \rightarrow [-1, 1]$ as $T_\alpha(\rho) = \cos(\alpha \arccos(\rho))$. A recurrent relation could be used to demonstrate Chebyshev polynomial map T_n: $R \rightarrow R$ of degree α, as given below:

$$T_\alpha(\rho) = 2\rho T_{\alpha-1}(\rho) - T_{\alpha-2}(\rho), \tag{1}$$

while $n \geq 2$, we have $T_0(\rho) = 1$ and $T_1(\rho) = \rho$
We can identify the first few Chebyshev polynomials as following:

$$T_2(\rho) = 2\rho^2 - 1 \tag{2}$$

$$T_3(\rho) = 4\rho^3 - 3\rho \tag{3}$$

$$T_4(\rho) = 8\,\rho^4 - 8\,\rho^2 + 1 \tag{4}$$

2. Chebyshev polynomial beholds the following features:
   The chaotic feature: For $\alpha \geq 1$, the Chebyshev polynomial could map $T_\alpha(\rho)$: $[-1, 1] \rightarrow [-1, 1]$ with degree α specifies a chaotic map having a constant density i.e., $f \times (\rho) = 1/\left(\pi\sqrt{1-\rho^2}\right)$ for all positive Lyapunov exponent ln α.
   The semigroup feature [24]: The semigroup feature of these Chebyshev polynomials can be represented on an interval $[-\infty, +\infty]$ as shown below:

$$T_\alpha(\rho) = (2\,\rho\,T_{\alpha-1}(\rho) - T_{\alpha-2}(\rho)) \bmod p \tag{5}$$

   Given that $\alpha \geq 2$, $\rho \in [-\infty, +\infty]$, and p shows a large prime number. Besides,

$$T_a(T_b(\rho))T_{ab}(\rho)T_b(T_a(\rho)) \bmod p \tag{6}$$

3. Chaotic maps based discrete logarithm problem (CMDLP): It is an intractable problem to output a, while given $T_a(\rho) = y$.
4. Chaotic maps based Diffie–Hellman problem (CMDHP): it is an intractable problem to output $T_{ab}(\rho)$, while given the parameters, $T_a(\rho)$ or $T_b(\rho)$.

We assume that there is not any polynomial time algorithm that could solve the above intractable problems with non-negligible probability.

2.2 One-Way Hash Function

We can define one-way secure hash operation $h: \mu \rightarrow \eta$ contains four features:

1. The hash function h inputs a message of randomly taken length and generates a fixed-length message digest.
2. Given μ, would be hard to find μ', such that $\mu' \neq \mu$, but $h(\mu') = h(\mu)$;
3. Given $h(\mu) = \eta$, it would be a hard problem to compute $h^{-1}(\eta) = \mu$;
4. It is also computationally intractable to locate a pair μ, μ' such that $\mu' \neq \mu$, but $h(\mu') = h(\mu)$.

3 Working and Inefficiencies in Tsai and Lo Scheme

This section describes the working and cryptanalysis of Tsai and Lo scheme as following:

3.1 Working of Tsai and Lo's scheme

The Tsai and Lo's protocol [36] comprises two phases: (1) Registration phase (2) Login and Authentication phase, as shown in Fig. 1. Some notations that have been used in the paper are given as under.

Some notations that have been used in the paper are given in Table 1 as under.

3.1.1 Server Registration Phase

The Tsai and Lo's protocol consists of a trusted registration centre (RC), and n number of trusted service providers Sj, where $(1 \leq j \leq n)$. The server Sj is registered through RC by

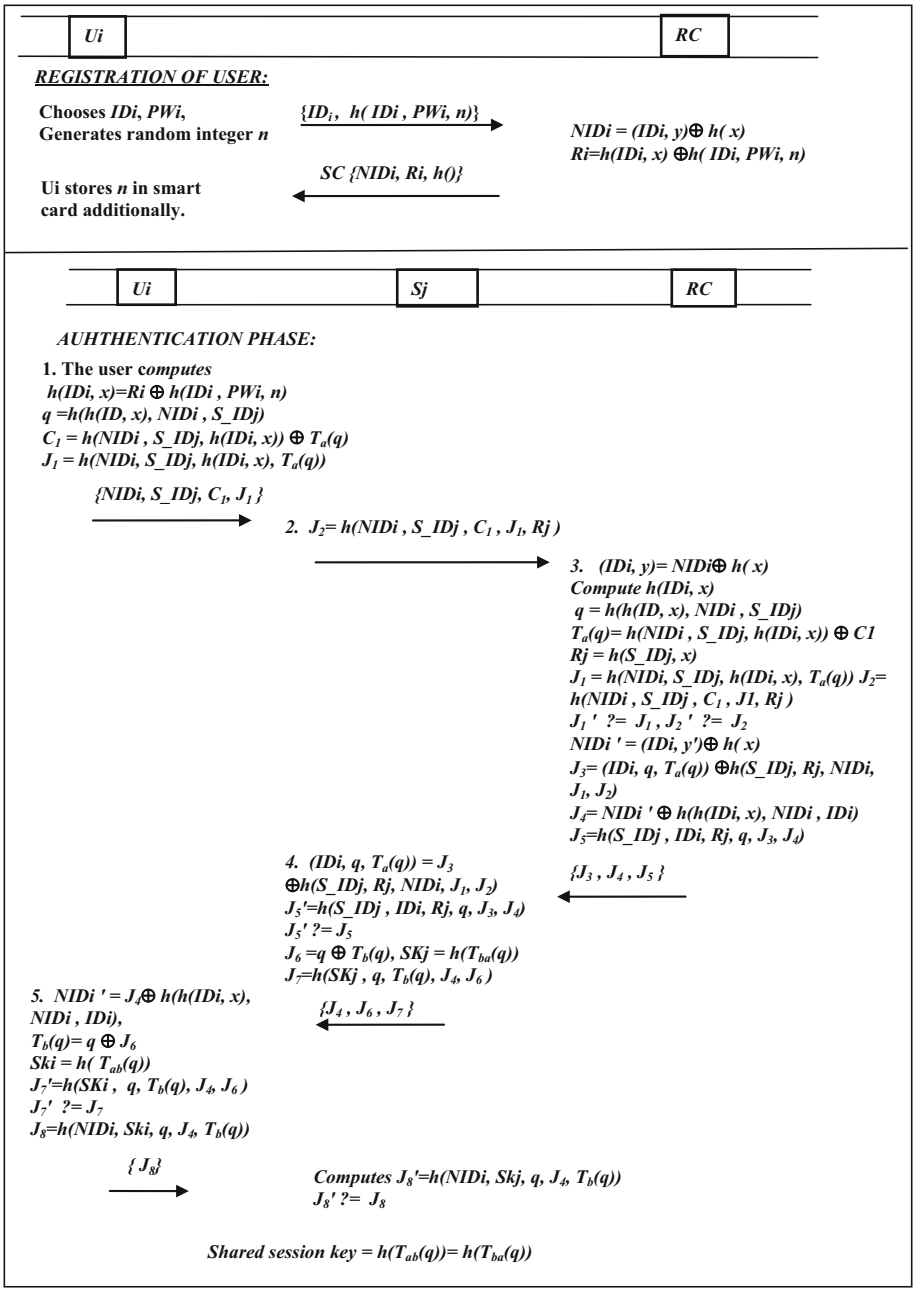

Fig. 1 Tsai and Lo model registration, login and authentication phase

having shared a secret Rj using a secure channel. First, the service provider Sj sends its identity ***S_IDj*** towards RC. Then, RC computes ***Rj = h(x, S_IDj)***, and sends towards Sj using a secure channel.

Table 1 Symbol notations

Notations	Description
Ui, Sj, RC	*ith* user, *jth* server, registration centre
IDi, PWi	Ui's Identity, Ui's password
NID_j	A shared parameter between *Sj* and *RC*
x, y	Master key and random secret of RC
S_IDj, Rj	Identity of *Sj*, Shared secret between *Sj* and *RC*
n, a, b, c	Randomly generated numbers: *(n, a)* by *Ui*, *b* by *Sj*, *c* by *Adversary*
$T_n(.), T_a(.), T_b(.)$	Chebyshev polynomial of degree *(n/a/b)*
q	Temporary secret mutually computed by *Ui* and *RC*
h(.)	A secure hash digest function
$\|\|, \oplus$	Concatenation function, XOR function

3.1.2 The User Registration Phase

In this phase, the Ui gets registered through RC. After registration process, Ui can obtain the services of all service providers, registered through the same registration centre. The steps for the user's registration are illustrated below:

1. First, Ui selects its identity ***IDi*** and password ***PWi***. Next, it generates a random number ***n*** and sends the message $\{ID_i, h(IDi, PWi, n)\}$ to registration centre.
2. Registration centre computes $NIDi = (IDi, y) \oplus h(x), Ri = h(IDi, x) \oplus h(IDi, PWi, n)$ and then stores $\{NIDi, Ri, h()\}$ in smart card. Finally, it sends the smart card to Ui.
3. The user receives the message, and adds the parameter *n* in smart card.

3.1.3 The Login and Authentication Phase

1. In this phase, user computes $h(IDi, x) = Ri \oplus h(IDi, PWi, n), q = h(h(ID, x), NIDi, S\_IDj), C_1 = h(NIDi, S\_IDj, h(IDi, x)) \oplus T_a(q),$ and $J_1 = h(NIDi, S\_IDj, h(IDi, x), T_a(q)).$ Then, it submits the message $\{NIDi, S\_IDj, C_1, J_1\}$ to Sj.
2. The Sj receives $\{NIDi, S\_IDj, C_1, J_1\}$ and compute $J_2 = h(NIDi, S\_IDj, C_1, J_1, Rj),$ and sends the message $\{NIDi, S\_IDj, C_1, J_1, J_2\}$ to registration centre for verification.
3. The registration centre receives $\{NIDi, S\_IDj, C_1, J_1, J_2\}$, computes $(IDi, y) = NIDi \oplus h(x), h(IDi, x), q = h(h(ID, x), NIDi, S\_IDj), T_a(q) = h(NIDi, S\_IDj, h(IDi, x)) \oplus C_1, Rj = h(S\_IDj, x), J_1 = h(NIDi, S\_IDj, h(IDi, x), T_a(q)),$ and $J_2 = h(NIDi, S\_IDj, C_1, J_1, Rj).$ Then, it will compare the equality for
4. $J_1' \stackrel{?}{=} J_1, J_2' \stackrel{?}{=} J_2$. If it holds true, it again computes $NIDi' = (IDi, y') \oplus h(x), J_3 = (IDi, q, T_a(q)) \oplus h(S\_IDj, Rj, NIDi, J_1, J_2), J_4 = NIDi' \oplus h(h(ID, x), NIDi, IDi), J_5 = h(S\_IDj, IDi, Rj, q, J_3, J_4).$ Then, ultimately it forwards $\{J_3, J_4, J_5\}$ to ***Sj*** for the purpose of verification.
5. Next, Sj calculates $(IDi, q, T_a(q)) = J_3 \oplus h(S\_IDj, Rj, NIDi, J_1, J_2), J_5' = h(S\_IDj, IDi, Rj, q, J_3, J_4),$ and checks the equality for $J_5' \stackrel{?}{=} J_5$. If it holds true, then further generates $J_6 = q \oplus T_b(q), Skj = h(T_{ba}(q)), J_7 = h(Skj, q, T_b(q), J_4, J_6),$ and forwards $\{J_4, J_6, J_7\}$ to Ui for further verification.

6. Ui, upon receiving the message $\{J_4, J_6, J_7\}$,computes $NIDi' = J_4 \oplus h(h(ID, x), NIDi, IDi)$, $T_b(q) = q \oplus J_6$, $Ski = h(T_{ab}(q))$ and $J_7' = h(Ski, q, T_b(q), J_4, J_6)$. It further compares $J_7' ? = J_7$. If it holds true, then calculates $J_8 = h(NIDi, Ski, q, J_4, T_b(q))$. Then it sends the message $\{J_8\}$ to Sj for verification.
7. Next, the server Sj calculates $J_8' = h(NIDi, Skj, q, J_4, T_b(q))$, and compares the equation $J_8' ? = J_8$. If the equation matching is positive, then it generates the session key as $Skj = Ski = h(T_{ba}(q)) = h(T_{ab}(q))$.

3.2 Weaknesses in Tsai and Lo. [36]

The Tsai and Lo [36] is a multi-server authentication protocol which is based on Chebyshev Chaotic Map (CCM). Despite, Lu et al.'s [39] indicated password guessing attack on [36], the scheme of Tsai and Lo is also vulnerable to *server-spoofing attack* provided that the smart card contents are leaked to adversary. Those smart card contents may be exposed to adversary out of differential power analysis attack. Onwards, if the same adversary comes to know about the Ui's identity IDi by any means, it could initiate a server-spoofing attack positively. We also assume that the attacker intercepts the publicly available messages i.e., $NIDi$, $NIDi$' and $J_4 = NIDi' \oplus h(h(IDi, x), NIDi, IDi)$ of two successive sessions. In this context, the adversary may compute the parameter $h(h(IDi, x), NIDi, IDi)$ by calculating $NIDi' \oplus J_4$. Further, by the use of smart card information including n, it tries all possible combinations of low entropy password PWi as shown below in Eqs. (7) and (8).

$$h(IDi, x)^* = Ri \oplus h(IDi, h(PWi^* || n)) \tag{7}$$

$$h(h(IDi, x)^* || NIDi || IDi) ? = h(h(IDi, x), NIDi, IDi) \tag{8}$$

If a single password combination PWi^* from many attempts, hits for example, the attacker comes to know the legitimate password and the valid $h(IDi, x)$ factor. Then, onwards, it could initiate a server-spoofing attack by designing a message $\{J_4, J_6, J_7\}$ using the steps as stated below:

1. It constructs the message J_4 by taking $NIDi$ from the authentication request and constructing $J_4 = NIDi^{old} \oplus h(h(IDi, x), NIDi, IDi)$. Since, the adversary, not capable of generating a novel $NIDi$, utilizes the old value of $NIDi^{old}$ for generating J_4. A user, normally does not store and maintain any record of $NIDi$, so it will not be able to trace the replay of $NIDi$.
2. Then, adversary computes $J_6 = q \oplus T_c(q)$ by computing $q = h(h(IDi, x), NIDi, S\_IDj)$ and $T_c(q)$, after assuming a random integer c.
3. Next, the adversary computes $J_7 = h(Skj, q, T_c(q), J_4, J_6)$ after computing $Skj = T_{ca}(q)$.
4. Following that, it could send the message $\{J_4, J_6, J_7\}$ towards Ui, which may be deceived positively, with the construction of a session key with the adversary as $Skj = T_{ac}(q)$.

4 Working and Inefficiencies in Lu et al. Scheme

This section describes the working and cryptanalysis of Lu et al.'s scheme as following:

4.1 Working of Lu et al.'s Scheme

The Lu et al.'s protocol [39] comprises two phases: (1) Registration phase (2) Login and Authentication phase, as shown in Fig. 2. Some notations that have been used in the paper are given as under.

Some notations that have been used in the paper are given in Table 2 as under.

4.1.1 Server Registration Phase

The Lu et al.'s protocol consists of a trustworthy registration centre (RC), and *n* number of authorized service providers Sj, where ($1 \leq j \leq n$). For registration, the server Sj sends its identity *S_IDj* to RC. Then, RC computes $\boldsymbol{R_j = h(S\_IDj, x)}$ and sends $\boldsymbol{R_j}$ towards server. The server then computes $\boldsymbol{Q_j = R_j \oplus r_j}$ and stores $\boldsymbol{Q_j}$ at a protected place to finalize the registration process.

4.1.2 The User Registration Phase

In this phase, the Ui adopts the following steps to complete the user registration process.

1. First, Ui selects its identity ***IDi***, password ***PWi*** imprints biometric $\boldsymbol{\beta_i}$ and computes $\boldsymbol{Gen(\beta_i) \rightarrow (\alpha_i, \beta_i)}$.
2. Next, it sends the message $\boldsymbol{\{ID_i, h(PWi, \alpha_i)\}}$ to RC. RC computes $\boldsymbol{R_i = h(ID_i, x) \oplus h(PWi, \alpha_i)}$, $\boldsymbol{Xi = h(ID_i, x) \oplus r}$, $\boldsymbol{Yi = h(x) \oplus h(PWi, \alpha_i)}$, and sends the message $\boldsymbol{\{X_i, R_i, Y_i\}}$ toward user.
3. The user receives the parameters and generate a random nonce $\boldsymbol{r_i}$ and further computes $\boldsymbol{L_i = Y_i \oplus r_i}$. Finally, it stores ***{Ri, Li, Xi, β_i, h(.)}*** in smart card as shown in Fig. 2.

4.1.3 The Login and Authentication Phase

1. In this phase, user first logs into smart card to get authenticated with server. For this purpose, Ui enters ***PWi***, $\boldsymbol{\beta_i'}$. Then, SC computes $\boldsymbol{Rep\ (\beta_i', \beta_i)}$ to generate $\boldsymbol{\alpha_i}$. Then, it computes $\boldsymbol{h(IDi, x) = R_i \oplus h(PWi, \alpha_i)}$, $\boldsymbol{r = X_i \oplus h(IDi, x)}$, $\boldsymbol{h(x) = L_i \oplus h(PWi, \alpha_i) - r_i}$, $\boldsymbol{C_i = T_{ri}(q) \oplus T_r(h(x))}$, $\boldsymbol{CIdi = IDi \oplus T_{ri}\ (T_r(h(x)))}$, $\boldsymbol{q = h(IDi, h(IDi, x))}$ and $\boldsymbol{A_i = h(q, T_{ri}(q), IDi, T_r(h(x)))}$. Finally, it sends the message ***{Ci, CIdi, Ai}*** to ***Sj*** for verification.

Table 2 Symbol Notations for Lu et al. [39]

Notations	Description
Gen(), Rep()	Generator and reproduction procedures used in fuzzy extractor [40]
x, r	Master key and random secret of RC
IDi, S_IDj	identity of *Ui* and *Sj*
r_i, r_j, r_j.	Randomly generated numbers: by *Ui*, *Sj*, or by Adversary
q	Temporary secret mutually computed by *Ui* and *RC*
$T_q(.), T_{ri}(.), T_{rj}(.)$	Chebyshev polynomials of degree ($q/r_i/r_j$)
β_i	Biometric parameter

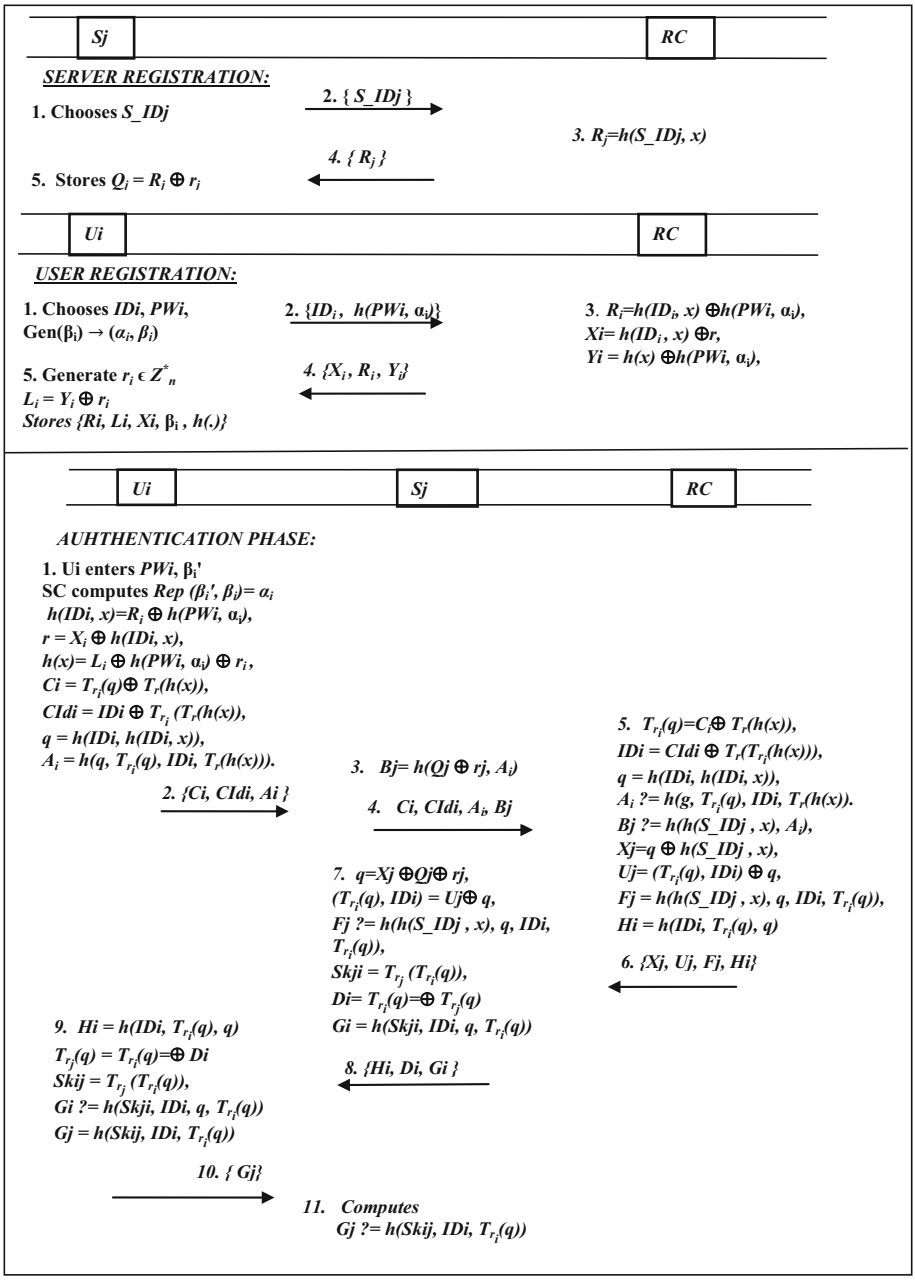

Fig. 2 Lu et al.'s model registration, login and authentication phase

2. Then, Sj computes $Bj = h(Qj \oplus rj, Ai)$ and sends the message $\{Ci, CIdi, Ai, Bj\}$ to RC.
3. After receiving the parameters $\{Ci, CIdi, Ai, Bj\}$ from Sj, RC computes $T_{ri}(q) = C_i \oplus T_r(h(x))$, $IDi = CIdi \oplus T_r(T_{ri}(h(x)))$ and $q = h(IDi, h(IDi, x))$. Next, RC checks the equality for $A_i ? = h(g, T_{ri}(q), IDi, T_r(h(x))$ and $Bj ? = h(h(S\_IDj, x),$

Ai). If it is not true, it aborts the session, otherwise validates Ui and Sj, and further computes $Xj = q \oplus h(S\_IDj, x)$, $Uj = (T_{ri}(q), IDi) \oplus q$, $Fj = h(h(S\_IDj, x), q, IDi, T_{ri}(q))$ and $Hi = h(IDi, T_{ri}(q), q)$. Finally, it sends the computed message ***{Xj, Uj, Fj, Hi}*** to ***Sj***. ***Sj***, then computes $q = Xj \oplus Qj \oplus rj$, $(T_{ri}(q), IDi) = Uj \oplus q$ and verifies the equality for $Fj ? = h(h(S\_IDj, x), q, IDi, T_{ri}(q))$. If true, then further computes the session key $Skji = T_{rj}(T_{ri}(q))$, $Gi = h(Skji, IDi, q, T_{ri}(q))$ and $Di = T_{ri}(-q) = \oplus T_{rj}(q)$. Then it sends the message ***{Hi, Di, Gi}*** towards Ui for verification.

4. Then, after receiving the message, Ui computes $Hi = h(IDi, T_{ri}(q), q)$, $T_{rj}(q) = T_{ri}(-q) = \oplus Di$ and $Skij = T_{rj}(T_{ri}(q))$. Then, it verifies the equality for $Gi ? = h(Skji, IDi, q, T_{ri}(q))$. On positive check, it validates the ***Sj*** as a valid entity, and computes $Gj = h(Skij, IDi, T_{ri}(q))$. Finally, it sends the message ***{Gj}*** towards ***Sj*** to enable the later verify the user ultimately.
5. The server receives ***Gj*** and computes $h(Skij, IDi, T_{ri}(q))$. Then, it verifies the equation $Gj ? = h(Skij, IDi, T_{ri}(q))$. If it is true, it verifies the user, otherwise aborts the session.

4.2 Weaknesses in Lu et al

The Lu et al. [39] scheme is found vulnerable to replay attack, RC-spoofing attack and server spoofing attack. Besides, the Lu et al.'s scheme fails to provide anonymity to user. There is also a technical flaw in Lu et al's scheme, which renders the scheme unfeasible to be applied practically, until the flaws are resolved. The weaknesses of Lu et al.'s scheme are illustrated below:

4.2.1 Technical Flaw in Lu et al. Scheme

In registration phase of Lu et al's scheme, the authors do not specify any mechanism for storing the random number r_i so that it may be accessed during user login procedure. The user cannot proceed with the protocol to compute the authentication request, until the parameter r_i is approached. Since, being a high entropy integer the users are not supposed to remember this number, it is advisable to get the number stored in smart card in encrypted form. Besides, the plain storage of r_i might lead to stolen smart card attacks, for the adversary could compute the Ui's parameter $h(PWi, \alpha_i)$ easily. Therefore, until the parameter r_i storage and access procedure is specified in the smart card, the scheme cannot be implemented practically.

4.2.2 Replay Attack

In Lu et al.'s scheme, the participants do not seem to generate fresh random numbers or nonce, which always result in the construction of identical session key as in previous session. The adversary may exploit this weakness to initiate replay attacks without getting noticed.

4.2.3 RC-Spoofing Attack and Server Spoofing Attack

A malicious user, acting as an adversary may intercept the message ***{Xj, Uj, Fj, Hi}*** sent from RC towards server, out of its own generated authentication request towards server. Upon intercepting the message ***{Xj, Uj, Fj, Hi}***, the insider adversary, having ***q*** may derive the server's secret parameter $h(S\_IDj, x)$ from $Xj = q \oplus h(S\_IDj, x)$. The adversary may

use this secret to launch RC-spoofing attack against the same server interacting with another user, by adopting the following steps:

1. Having intercepted ***{Xj, Uj, Fj, Hi}*** from previous sessions of some participants (Ui and Sj), the adversary may recover ***q*** and ***IDi*** by computing $q = Xj \oplus h(S\_IDj, x)$ and $(T_{ri}(q), IDi) = Uj \oplus q$.
2. Later, the adversary may intercept the authentication request ***{Ci, CIdi, Ai, Bj}*** for the same Ui intended to acquire services from the same Sj, and replay the same parameters {***Xj, Uj, Fj, Hi}*** towards Sj.
3. Sj, then computes $q = Xj \oplus Qj \oplus rj$, $(T_{ri}(q), IDi) = Uj \oplus q$ and checks the equality for $Fj ? = h(h(S\_IDj, x), q, IDi, T_{ri}(q))$, which will be matched in the same manner as it did previously. Hence the Lu et al. scheme is vulnerable to RC-spoofing attack.
4. On the other hand, the same adversary may also launch server-spoofing attack by directly constructing a message ***{Hi, Di, Gi}*** after computing $Di = T_{ri}(q) = \oplus T_{ri'}(q)$, $Hi = h(IDi, T_{ri}(q), q)$, $Skji = T_{rj'}(T_{ri}(q))$ and $Gi = h(Skji, IDi, q, T_{ri}(q))$, where ***rj'*** is a random number generated by adversary. The message is then forwarded to Ui, which is duly verified by the user, however fake.

4.2.4 No Anonymity

As we see earlier, the adversary having come to extract the parameter ***h(S_IDj, x)*** may further compute the identity by calculating $q = Xj \oplus h(S\_IDj, x)$ and $(T_{ri}(q), IDi) = Uj \oplus q$, subject to the intercepted parameter ***Xj***. Hence, the Lu et al.'s scheme does not provide anonymity to the user.

5 Proposed Model

The proposed model addresses the identified loopholes in Tsai and Lo, and Lu et al. by contributing its improved scheme, with the modifications highlighted as shown in Fig. 3. In this regard, this section covers the registration, login and authentication phase, and password modification phase, as following:

5.1 Registration Phase (Server)

We assume the similar system setup in proposed scheme, as Tsai and Lo scheme. i.e. the proposed scheme setup contains a trusted registration centre with n number of trusted service providers Sj, where ($1 \leq j \leq n$). The server, Sj gets registered with registration centre by sharing a secret Rj on a secure channel. The service provider, termed as server (Sj), submits its identity ***S_IDj*** towards RC with the objective of getting registration. RC, then, calculates $Rj = h(x, S\_IDj)$, and sends it towards Sj using a secure channel. All of the service providers Sj, in the system, get registered through RC, in this manner.

5.2 Registration Phase (User)

For registration, Ui seeks to register from RC, to avail the services offered by different service providers Sj, by performing the following steps:

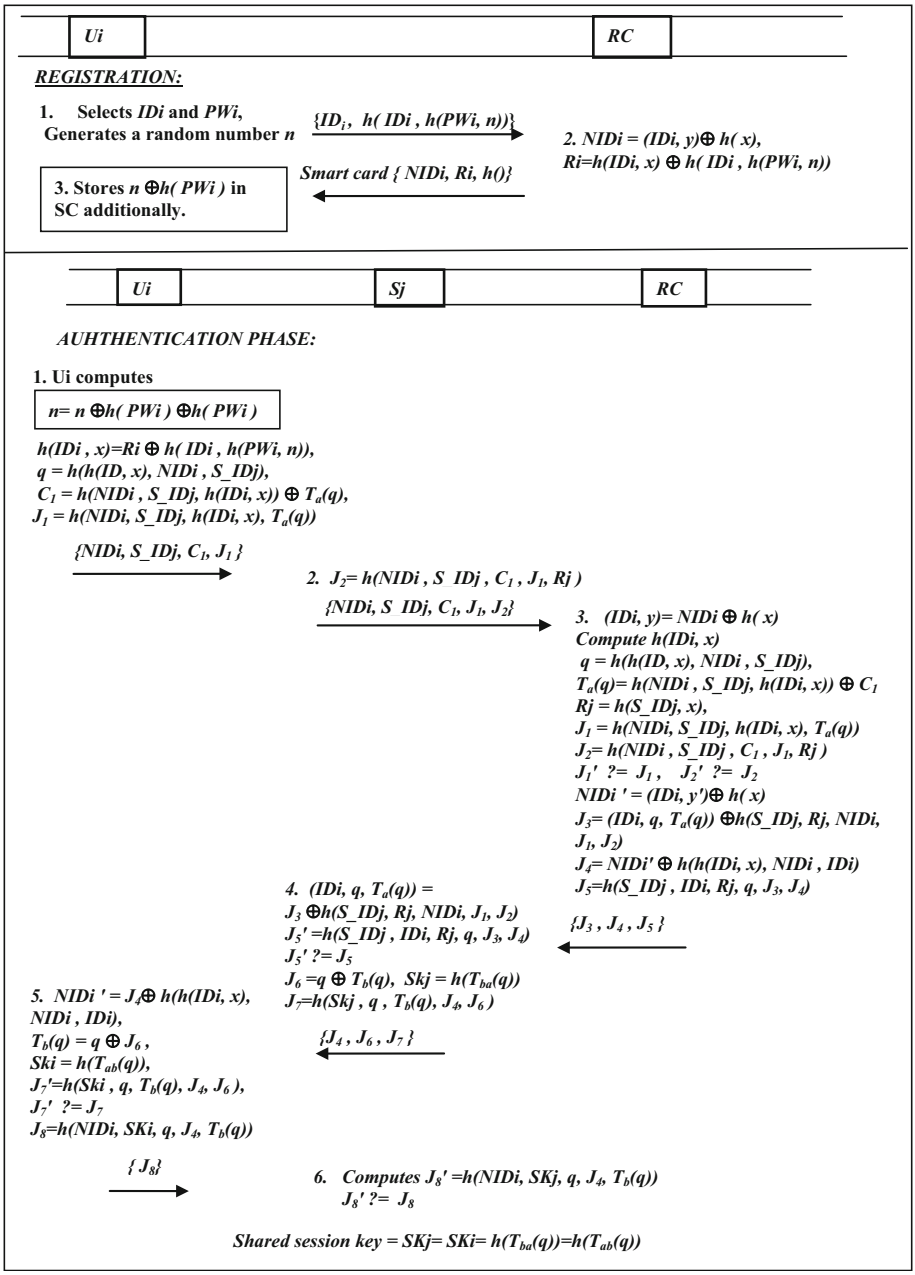

Fig. 3 Proposed model registration, login and authentication phase

1. The user chooses its identity ***IDi*** and password ***PWi***. Then, it randomly generates a number ***n*** and submits the message **{*IDi, h(IDi, h(PWi, n))*}** towards registration centre.

2. Then, registration centre calculates $NIDi = (IDi, y) \oplus h(x)$, $Ri = h(IDi, x) \oplus h(IDi, h(PWi, n))$ and stores the parameters $\{NIDi, Ri, h()\}$ in its smart card. Thereafter, it sends smart card to user.
3. Ui receives, computes $n \oplus h(PWi)$ and stores in smart card, additionally.

5.3 Authentication Phase

1. For authentication, Ui calculates $n = n \oplus h(PWi) \oplus h(PWi)$, $h(IDi, x) = Ri \oplus h(IDi, h(PWi, n))$, $q = h(h(ID, x), NIDi, S\_IDj)$, $C_1 = h(NIDi, S\_IDj, h(IDi, x)) \oplus T_a(q)$, and $J_1 = h(NIDi, S\_IDj, h(IDi, x), T_a(q))$. Next, the user sends $\{NIDi, S\_IDj, C_1, J_1\}$ towards Sj for the purpose of verification.
2. Then, server receives message $\{NIDi, S\_IDj, C_1, J_1\}$ and computes $J_2 = h(NIDi, S\_IDj, C_1, J_1, Rj)$. Then, it sends the constructed message $\{NIDi, S\_IDj, C_1, J_1, J_2\}$ to registration centre for verification.
3. RC receives the message $\{NIDi, S\_IDj, C_1, J_1, J_2\}$, computes $(IDi, y) = NIDi \oplus h(x)$, $h(IDi, x)$, $q = h(h(ID, x), NIDi, S\_IDj)$, $T_a(q) = h(NIDi, S\_IDj, h(IDi, x)) \oplus C_1$, $Rj = h(S\_IDj, x)$, $J_1 = h(NIDi, S\_IDj, h(IDi, x), T_a(q))$, and $J_2 = h(NIDi, S\_IDj, C_1, J_1, Rj)$. Next, it verifies the equations $J_1' \stackrel{?}{=} J_1$ and $J_2' \stackrel{?}{=} J_2$. If positive, then it further calculates $NIDi' = (IDi, y') \oplus h(x)$, $J_3 = (IDi, q, T_a(q)) \oplus h(S\_IDj, Rj, NIDi, J_1, J_2)$, $J_4 = NIDi' \oplus h(h(ID, x), NIDi, IDi)$, $J_5 = h(S\_IDj, IDi, Rj, q, J_3, J_4)$. It finally forwards $\{J_3, J_4, J_5\}$ towards Sj for further verification.
4. Next, Sj calculates $(IDi, q, T_a(q)) = J_3 \oplus h(S\_IDj, Rj, NIDi, J_1, J_2)$, $J_5' = h(S\_IDj, IDi, Rj, q, J_3, J_4)$, and also verifies the check for $J_5' \stackrel{?}{=} J_5$. If positive, then further computes the values $J_6 = q \oplus T_b(q)$, $Skj = h(T_{ba}(q))$, $J_7 = h(Skj, q, T_b(q), J_4, J_6)$. Then, it sends the message $\{J_4, J_6, J_7\}$ to user for further proceedings.
5. After receiving the message $\{J_4, J_6, J_7\}$, Ui computes $NIDi' = J_4 \oplus h(h(ID, x), NIDi, IDi)$, $T_b(q) = q \oplus J_6$, $Ski = h(T_{ab}(q))$, $J_7' = h(Ski, q, T_b(q), J_4, J_6)$. It further compares the equation $J_7' \stackrel{?}{=} J_7$. If it holds true, then computes $J_8 = h(NIDi, Ski, q, J_4, T_b(q))$, and sends the message $\{J_8\}$ to Sj.
6. The server Sj receives and computes $J_8' = h(NIDi, Skj, q, J_4, T_b(q))$, and checks the equality for $J_8' \stackrel{?}{=} J_8$. If it holds true, then constructs the session key ultimately as $Skj = SKi = h(T_{ba}(q)) = h(T_{ab}(q))$.

5.4 Password Modification Phase

Ui changes its old password PWi^{old} into a new password PWi^{new}, without any interaction with RC, by adopting the following procedure:

1. The user inserts its SC into card reader and also inputs its identity IDi and password PWi^{old}.
2. Next, the SC computes $n = n \oplus h(PWi^{old}) \oplus h(PWi^{old})$. Then, it further computes $h(IDi, x) = Ri \oplus h(IDi, h(PWi^{old}, n))$ and $Ri' = h(IDi, x) \oplus h(IDi, h(PWi^{new}, n))$ after selecting a new password.
3. Next, it computes $n \oplus h(PWi^{new})$ and replaces Ri and $n \oplus h(PWi^{old})$ with Ri' and $n \oplus h(PWi^{new})$ respectively, in the SC.

In this manner, the Ui changes the password, without any RC involvement.

6 Security Analysis

An improved MSA technique has been devised following the identified attacks in Tsai and Lo as described in the above section. The current section encompasses the informal and formal security analysis of the proposed protocol.

6.1 Informal Security Analysis

The informal security analysis has been elaborated as following.

6.1.1 Password Guessing Attack

An attacker may attempt to extract a Ui's password with the help of open available message parameters i.e. ***{NIDi, S_IDj, C_1, $J_1 - J_8$}***. An attacker may gather the message parameters i.e., *NIDi*, *NIDi'* and J_4 i.e., $J_4 = NIDi' \oplus h(h(ID, x), NIDi, IDi)$ from observing two successive sessions, and derive the parameter $h(h(ID, x), NIDi, IDi)$ by computing $NIDi' \oplus J_4$, for the calculation of ***PWi***. Unlike, Tsai and Lo, will be unable to guess ***PWi*** by initiating a brute force attack, since, guessing the password ***PWi*** using (3) and (4), requires easy access to parameter ***n,*** as was in Tsai and Lo scheme. However, the proposed scheme stores ***n*** in the form of $n \oplus h(PWi)$, which makes it hard for an attacker to guess the ***PWi*** in polynomial time.

6.1.2 Resistance to Server Spoofing Attacks

The server spoofing attacks can be initiated by an attacker by impersonating as a server towards some legal user participant.

In proposed scheme, given that, an adversary cannot guess ***PWi*** by initiating a brute force attack as remarked in Sect. 4.1, the former may not be able to launch a server spoofing attack, despite having the knowledge of SC contents and user's identity.

6.1.3 Replay Attacks

The replay attacks can be initiated by an adversary who could repeat the intercepted message or parameter to forge any of the legitimate participants.

An adversary, having the openly available message parameters ***{NIDi, S_IDj, C_1, $J_1 - J_8$}***, may maliciously attempt to replay the same to any of the three legitimate participants i.e., user, server, and RC. However, in proposed scheme, the attacker may not be able to launch an attack. For instance, when an attacker sends the replayed message ***{NIDi, S_IDj, C_1, J_1}*** towards Sj, the latter removes any possibility of replay after verifying the equality for $J_8' \; ? = J_8$. Likewise, the Ui removes any possibility of replay attack by verifying the equality check for $J_7' \; ? = J_7$. The registration centre, however, will always respond with the of construction of message ***{J_3, J_4, J_5}***, without discerning any kind of replay attack. Since, the generation of message ***{J_3, J_4, J_5}*** does not employ heavy operations in terms of cost, so it may not be termed as a DOS (Denial of Service) attack and attacker may not gain any advantage of replaying the messages towards RC.

6.1.4 Man in the Middle Attack (MiTM)

This attack may be launched by an attacker who acts as silent intermediary among the participants involved in a session. In this attack, is able enough to make the other participants believe it as legal entity; however the former will not be a valid participant, though.

An adversary could not launch a MiTM attack, since no intermediary can access the J_4 and J_5 message parameters, where J_4 equals to $NIDi' \oplus h(h(IDi, x), NIDi, IDi)$ and J_5 equals to $h(S\_IDj, IDi, Rj, q, J_3, J_4)$. Their construction requires access to $h(IDi, x)$ and Rj respectively. The user may thwart any MiTM attack by verifying the equality check for $J_7' \; ? = J_7$, where, J_7 equals to $h(SKj, q, T_b(q), J_4, J_6)$. Likewise, the Sj removes any possibility of MiTM attack after verifying the equality check for $J_8' \; ? = J_8$. Where J_8 equals to $h(NIDi, SKi, q, J_4, T_b(q))$.

6.1.5 Stolen Verifier Attacks

An attacker may steal password-based verifiers stored on the server's end; if the latter maintains the database of those password verifiers, and attempt to masquerade the legal participants which are known as stolen verifier attack.

The proposed scheme is immune to stolen verifier attacks as it does not maintain any such database on either Sj or RC's end, which is a necessary condition for an adversary to initiate such kind of attack.

6.1.6 Stolen Smart Card

An attacker may steal a smart card and attempt to use the extracted information for guessing passwords by inputting all of the possible combinations using brute force approach.

Using SC, an adversary might try to manipulate with its contents. However, those available contents $\{NIDi, Ri, n \oplus h(PWi)\}$ are useless, since PWi extraction from $Ri = h(IDi, x) \oplus h(IDi, h(PWi, n))$ is not possible until n is recovered from $n \oplus h(PWi)$ function as remarked in Sect. 4.1. At the same time, these parameters does not contribute in guessing the session key $Skj = Ski = h(T_{ba}(q)) = h(T_{ab}(q))$. Hence, the stolen card may not be beneficial to attacker in suchlike manner.

6.1.7 Session Key Security

The session key security signifies towards the privacy of the session key between the participants, establishing it. In proposed scheme, the session key is constructed as $h(T_{ab}(q)) = h(T_{ba}(q))$. For generating a valid session key an adversary needs to approach either a or b, besides accessing $T_a(q)$ or $T_b(q)$, which is inaccessible to adversary, and a hard problem to derive from $T_a(q)$ or $T_b(q)$.

6.1.8 Known-Key Security

This security feature signifies towards guessing the other session keys of the participants, provided that the current session key is revealed.

If we assume, the current session key $Skj = Ski = h(T_{ba}(q)) = h(T_{ab}(q))$ gets exposed to adversary, still it will not be able to guess other session keys for the same participants. Since, it requires assuming new random numbers each time, a session is created. Hence, the disclosure of any session key does not harm the integrity of other session keys.

6.1.9 Perfect Forward Secrecy

The perfect forward secrecy describes the property of protection for session keys, if a long-term master key or secret of a user or any server gets revealed.

The proposed scheme ensures perfect forward secrecy, despite of the fact that the long term secrets of the entities get revealed. The reason being, our scheme relies on Chaotic maps based discrete logarithm problem (CMDLP) for its robustness that leads to perfect forward secrecy. If an adversary manages to steal the $T_a(x)$ or $T_b(x)$, however, it can never compute a or b for computing the session key as $T_{ab}(x)$, due the hard problem of computing a or b from $T_a(x)$ or $T_b(x)$.

6.1.10 Mutual Authentication

This security feature ensures that the involved entities authenticate one another after the execution of the same protocol. Our proposed model ensures mutual authentication for both legal entities, Ui and Sj. Here, Ui verifies the server Sj by comparing the equation $J_7' \; ? = J_7$, while $J_7 = h(Skj, q, T_b(q), J_4, J_6)$. Likewise, the server Sj verifies Ui by checking the equality $J_8' \; ? = J_8$, while $J_8 = h(NIDi, Ski, q, J_4, T_b(q))$. If these equations do not match for the two entities on either side, the mutual authentication will not take place, and the session will be aborted by the entity performing verification.

6.1.11 Anonymous Authentication

The anonymous authentication provides anonymity to a user during mutual authentication with Sj. The adversary cannot make out the identity of a user by intercepting the publicly available messages.

In proposed model, Ui sends its dynamic identity $h(IDi, x)$, or IDi in the form of message

$NIDi = (IDi, y) \oplus h(x)$ as developed by the RC. The induction of such method nullifies any chances of Ui's leakage of identity. Hence, our scheme provide anonymous authentication for user Ui.

6.1.12 Resistance Against Bergamo et al.'s threat

The Bergamo et al.'s attack in the form of computing a valid session key, could be launched on our proposed chaotic map-based authentication protocol provided that:

1. The related parameters, i.e. q, $T_a(q)$, $T_b(q)$ are recovered by adversary.
2. The several Chebyshev-based polynomials might pass through a single point for the reason of periodicity of the cosine function.

In proposed scheme, the adversary may not be able to extract or compute $q = h(h(ID, x), NIDi, S\_IDj)$ either without accessing the user's password PWi or compromising the RC's entity. Likewise, the later may not be able to access $T_a(q)$ from $C_1 = h(NIDi, S\_IDj, h(IDi, x)) \oplus T_a(q)$ without knowledge of $h(NIDi, S\_IDj, h(IDi, x))$. Similarly, it may not be able to recover $T_b(q)$ from the equation $J_6 = q \oplus T_b(q)$ without the knowledge of q. At the same time, a and b are temporary random numbers generated by the participants, which are difficult to approach by adversary. The later may not be able to construct a valid session key $T_{ab}(q)$ until the factors q, $T_a(q)$, $T_b(q)$, a or b are accessed.

6.2 Formal Security Analysis

By employing random oracle model, we have performed a formal security analysis that might prove the security of the proposed protocol [31]. To meet the purpose, we employ two oracles as given under:

Reveal1: The Reveal1 oracle gives the parameter 'α' from its related hash value $\beta = h(\alpha)$, unconditionally.
Reveal2: The Reveal2 oracle outputs 'a' from the corresponding chaotic map parameter $T_a(q)$, unconditionally.

Theorem 1 *Given that, a one-way or unidirectional hash function behaves nearly as a random oracle, the proposed scheme stands secure, in case an attacker tries to derive the shared session key SK between the participants (Ui and Sj), by taking CMDLP as assumption.*

Proof For proving the above statement, we assume an adversary , who is in capacity of extracting the shared session key SK between the participants (Ui and Sj), employing the oracle *Reveal1* to implement the said algorithm $EXP1_{ISCMST}^{HASH,SC}$. The success probability for $EXP_{ISCMST}^{HASH,SC}$ is Succ $= \Pr.2[EXP_{ISCMST}^{HASH,SC} = 1] - 1$, where the $\Pr[\mathcal{E}]$ represents probability for an event $\mathcal{E}$. We can notice here that the advantage function for the current experiment will become $Adv_{ISCMST}^{HASH,SC}(t_1, q_{R1}, q_{R2}) = \max [Succ_{ISCMST}^{HASH,SC}]$, having execution time t_1 and the corresponding Reveal queries q_{R1} and q_{R2} maximized on . We describe the proposed protocol as provably secure against an attacker about extracting the shared session key SK between the same participants (Ui and Sj), if $Adv_{ISCMST}^{HASH,SC}(t_1, q_{R1}, q_{R2}) \leq \varepsilon$ for any sufficiently small $\varepsilon > 0$. The experiment demonstrates that if is capable of inverting the one-way hash function, and to solve the intractable problem CMDLP, then it may extract the original session key SK as used between the legitimate participants, and finally wins the game. Nonetheless, according to the definition in Sect. 2.1 (3–4) and 2.2, this is practically not possible to invert that hash function and solve CMDLP, since Adv_{ISCMST}^{HASH} and $Adv_{ISCMST}^{HASH,SC}(t) \leq \varepsilon$ for any sufficiently small $\varepsilon > 0$. Hence, the proposed scheme can be regarded as immune as the security properties for hash operation and CMDLP are pretty robust and hard to break.

Algorithm 1. EXP^{HASH}_{ISCMST}

1. Eavesdrop the Login request message $\{ NIDi, S\_IDj, C_1, J_1 \}$ in the login phase, where $C_1 = h(NIDi, S\_IDj, h(IDi, x)) \oplus T_a(q)$, $J_1 = h(NIDi, S\_IDj, h(IDi, x), T_a(q))$ and $NIDi = (IDi, y) \oplus h(x)$.
2. Call Reveal1 oracle on input J_1 to retrieve $NIDi'$, S_IDj, $h(IDi, x)$, $T_a(q)$ as $(NIDi, S\_IDj, h(IDi, x), T_a(q)) \leftarrow reveal1\ (J_1)$
3. Call Reveal1 oracle for input $h(IDi, x)$ that gives IDi' and x as $(IDi', x) \leftarrow reveal1\ (h(IDi, x))$
4. Eavesdrop the message $\{ J_5 \}$ in the Authentication phase, where $J_5 = h(S\_IDj, IDi, Rj, q, J_3, J_4)$.
5. Call Reveal1 oracle on input J_5 to retrieve S_IDj, IDi'', Rj, q', J_3, J_4 as $(S\_IDj, IDi, Rj, q, J_3, J_4) \leftarrow reveal1\ (J_5)$
6. Eavesdrop the message $\{ J_7 \}$ in the Authentication phase, where $J_7 = h(SKj, q, T_b(q), J_4, J_6)$.
7. Call Reveal1 oracle on input J_7 to retrieve SKj, q'', $T_b(q)$, J_4, J_6 as $(SKj, q, T_b(q), J_4, J_6) \leftarrow reveal1\ (J_7)$
8. **If** $(IDi' = IDi'')$ **AND** $(q' = q'')$ **Then**
9. Next, eavesdrop another authentication message $\{J_8\}$ in authentication phase, where $J_8 = h(NIDi, Ski, q, J_4, T_b(q))$.
10. Call Reveal1 oracle on input J_8 to retrieve $NIDi''$, Ski', q, J_4, $T_b(q)$ as $(NIDi, Ski, q, J_4, T_b(q)) \leftarrow reveal1\ (J_8)$
11. Call Reveal2 oracle on input $T_b(q)$ to retrieve b' as $(b) \leftarrow reveal2\ (T_b(q))$
12. Then, compute $SK'' = h(T_{ab}(q))$ using b and $T_a(q)$, by taking chaotic map and hash operation.
13. **If** $(NIDi' = NIDi'')$ **AND** $(Ski' = Ski'')$ **Then**
14. Accept IDi as the correct identity IDi of the user Ui, and $Skj = Ski) = h(T_{ba}(q)) = h(T_{ab}(q)$ as the valid session key between the participants (Ui and Sj).
15. **Return 1 (success)**
16. Otherwise
17. **Return 0 (failure)**
18. **End if**
19. **End if**

7 Comparison and Performance Analysis

The Lu et al.'s and Tsai and Lo's scheme employs Chaotic Chebyshev map for the mutual authentication among the communicating participants. As mentioned earlier, the computation for Chebyshev chaotic polynomial is nearly more than three times of ECC, and it is even more delay-efficient than RSA [2–10]. The Chebyshev chaotic polynomial is computationally efficient for its less key size, bandwidth requirements and memory consumption [38]. The following section illustrates the cost comparison for proposed scheme with Tsai and Lo scheme.

A few notations used in the comparison as T_H, T_M and T_{CCM} are defined as under:

T_H: The time taken for the 160-bit hash operation;
T_M: The time taken for a point multiplication operation;
T_{CCM}: The time to execute the map for Chebyshev Chaotic polynomial, i.e. $T_n(x)$ mod p keeping in view the algorithm [38].

To make the computational cost-based comparison keeping in view the running timings of different crypto-primitives, we based our results on the PBC library, with (Ubuntu 12.04.2) 32-bit operating system, with 3.6 GHz CPU, and 4.0 GB RAM. In this regard, the computational time for operations of one-way hash operation, scalar point multiplication and Chebyshev chaotic polynomial amounts to 0.0006 s, 0.0733 s, 0.02104 s, respectively. The total cost for Li et al. [34], Khan and He. [33], Lu et al. [39], Tsai and Lo [36], and proposed scheme amounts to 0.0174, 0.45, 0.18092, 0.05408 and 0.05468 s, respectively. The Li et al. being a simple hash based scheme is susceptible to replay and stolen verifier

attacks. The Khan and He, although free of attacks, employs costly operations like point multiplication operations, which renders the scheme inapplicable for low end devices having scarce resources. The Lu et al. scheme [39], improved upon Tsai and Lo's scheme, is also vulnerable to RC-spoofing attack, replay attack and lacks anonymity. The cost difference for Tsai and Lo, and proposed protocol is quite trivial, yet the proposed scheme fairly resists the identified threats that Tsai and Lo has been found unable to deal with. The XOR function cost is deemed to be negligible as compared to other operations of cryptography. Therefore, the cost may be ignored. The comparison for different security features for scheme [33, 34, 36, 39] and proposed scheme, is shown in Table 3.

The proposed scheme modifies the registration and user authentication phases of the Tsai and Lo protocol, in a way that the proposed scheme withstands server spoofing attack, password guessing attack, and stolen smart card attack, as contrary to Tsai and Lo scheme. The Table 4 compares the cost of five schemes and depicts that the proposed scheme is more secure than all the five schemes and is resistant to threats as posed to, particularly Tsai and Lo [36] and Lu et al. [39].

8 Conclusion

The multi-server authentication has been acknowledged as one of the required component of the current internet authentication paradigm. A lot of schemes have been proposed in the last decade by the research academia. This paper studies the Tsai and Lo scheme, and Lu et al. scheme which are based on multi-server remote authentication. The review of both schemes has been presented thoroughly. The Tsai and Lo scheme was found vulnerable to few more threats, besides Lu et al. indicates. Our cryptanalysis reveals the two ways, in which the scheme could be attacked, (1) server spoofing attack (2) and stolen smart card attack. At the same time, the Lu et al.'s scheme is also vulnerable to replay attack, RC and

Table 3 Comparison for security features

	Li et al. [34]	Khan and He [33]	Lu et al. [39]	Tsai and Lo [36]	Proposed protocol
Anonymity	†Yes	Yes	‡No	Yes	Yes
Mutual authentication	Yes	Yes	Yes	Yes	Yes
Resist stolen smart card attack	Yes	Yes	Yes	No	Yes
Resist offline password guessing attack	Yes	Yes	Yes	No	Yes
Resist server spoofing attack	Yes	Yes	No	No	Yes
Resist replay attack	No	Yes	No	Yes	Yes
Session key agreement	Yes	Yes	Yes	Yes	Yes
Man-in-the-middle attack	Yes	Yes	Yes	Yes	Yes
Resists against Bergamo et al. attack	–	–	Yes	No	Yes
Perfect forward secrecy	Yes	Yes	Yes	Yes	Yes
Known key secrecy	Yes	Yes	Yes	Yes	Yes
Resist stolen verifier attack	No	Yes	Yes	Yes	Yes

† Yes: Protocol is resistant to attacks

‡ No: Protocol is not resistant to threats

Table 4 Computational cost comparison

	Li et al. [34]	Khan and He [33]	Lu et al. [39]	Tsai and Lo [36]	Proposed protocol
C1	$11T_H$	$5T_H + 3T_M$	$8T_H + 4T_{CCM}$	$6T_H + 1T_{CCM}$	$8T_H + 1T_{CCM}$
C2	$6T_H$	$6T_H + 2T_M$	$4T_H + 2T_{CCM}$	$6T_H + 1T_{CCM}$	$6T_H + 1T_{CCM}$
C3	$12T_H$	$6T_H + 1T_M$	$9T_H + 2T_{CCM}$	$8T_H$	$8T_H$
Total	0.0174 s	0.45 s	0.18092 s	0.05408 s	0.05528 s

C1, Total communication cost for the user; C2, Total communication cost for the server; C3, Total communication cost for the registration centre

server spoofing attack, lack anonymity. The proposed study counters these attacks with the contribution of an improved version. Besides, this research work presents the security analysis formally and the performance efficiency analysis with other notable schemes.

Acknowledgements The work of Qi XIE was supported by Natural Science Foundations of Zhejiang Province (No. LZ12F02005), and the Major State Basic Research Development (973) Program of China (No.2013CB834205).

References

1. Xiao, D., Liao, X., & Deng, S. (2008). Using time-stamp to improve the security of a chaotic maps-based key agreement protocol. *Information Sciences, 178,* 1598–11602.
2. Han, S. (2008). Security of a key agreement protocol based on chaotic maps. *Chaos, Solitons & Fractals, 38,* 764–768.
3. Xiao, D., Liao, X., & Deng, S. (2007). A novel key agreement protocol based on chaotic maps. *Information Sciences, 177,* 1136–1142.
4. Xiang, T., Wong, K., & Liao, X. (2009). On the security of a novel key agreement protocol based on chaotic maps. *Chaos, Solitons & Fractals, 40*(2), 672–675.
5. Han, S., & Chang, E. (2009). Chaotic map based key agreement with/out clock synchronization. *Chaos, Solitons & Fractals, 39,* 1283–1289.
6. Yoon, E. J., & Yoo, K. Y. (2008). A new key agreement protocol based on chaotic maps. In N. T. Nguyen, G. S. Jo, R. J. Howlett, & L. C. Jain (Eds.), *Agent and multi-agent systems: Technologies and applications* (pp. 897–906). Springer: Heidelberg.
7. Gong, P., Li, P., & Shi, W. B. (2012). A secure chaotic maps-based key agreement protocol without using smart cards. *Nonlinear Dynamics, 70*(4), 2401–2406.
8. Guo, X., & Zhang, J. (2010). Secure group key agreement protocol based on chaotic hash. *Information Sciences, 180,* 4069–4074.
9. Niu, Y., & Wang, X. (2011). An anonymous key agreement protocol based on chaotic maps. *Communications in Nonlinear Science and Numerical Simulation, 16*(4), 1986–1992.
10. Wang, X., & Zhao, J. (2010). An improved key agreement protocol based on chaos. *Communications in Nonlinear Science and Numerical Simulation, 15*(12), 4052–4057.
11. Tseng, H., Jan, R., & Yang, W. (2009). A chaotic maps-based key agreement protocol that preserves user anonymity. In *IEEE international conference on communications (ICC09)* (pp. 1–6).
12. He, D., Chen, Y., & Chen, J. H. (2012). Cryptanalysis and improvement of an extended chaotic maps-based key agreement protocol. *Nonlinear Dynamics, 69*(3), 1149–1157.
13. Chaudhry, S. A., Naqvi, H., Mahmood, K., Ahmad, H. F., & Khan, M. K. (2016). An improved remote user authentication scheme using elliptic curve cryptography. *Wireless Personal Communication*. doi:10.1007/s11277-016-3745-3.
14. Khan, I., Chaudhry, S. A., Sher, M., Khan, J. I., & Khan, M. K. (2016). An anonymous and provably secure biometric based authentication scheme using chaotic maps for accessing medical drop box data. *Journal of Supercomputing*. doi:10.1007/s11227-016-1886-5.
15. Guo, C., & Chang, C. C. (2013). Chaotic maps-based password-authenticated key agreement using smart cards. *Communications in Nonlinear Science and Numerical Simulation, 18*(6), 1433–1440.

16. Yoon, E. J. (2012). Efficiency and security problems of anonymous key agreement protocol based on chaotic maps. *Communications in Nonlinear Science and Numerical Simulation, 17*(7), 2735–2740.
17. Chaudhry, S. A. (2016). A secure biometric based multi-server authentication scheme for social multimedia networks. *Multimedia Tools and Applications*. doi:10.1007/s11042-015-3194-0.
18. Lee, C. C., Li, C. T., & Hsu, C. W. (2013). A three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps. *Nonlinear Dynamics, 73*(1–2), 125–132.
19. Chaudhry, S. A., Naqvi, H., Sher, M., Farash, M. S., & Hassan, M. U. (2015). An improved and provably secure privacy preserving authentication protocol for SIP. *Peer to Peer Networking and Applications*. doi:10.1007/s12083-015-0400-9.
20. Chaudhry, S. A., Naqvi, H., Shon, T., Sher, M., & Farash, M. S. (2015). Cryptanalysis and improvement of an improved two factor authentication protocol for telecare medical information systems. *Journal of Medical Systems*. doi:10.1007/s10916-015-0244-0.
21. Chaudhry, S. A., Farash, M. S., Naqvi, H., Kumari, S., & Khan, M. K. (2015). An enhanced privacy preserving remote user authentication scheme with provable security. *Security and Communication Networks*. doi:10.1002/sec.1299.
22. Kocarev, L. (2001). Chaos-based cryptography: a brief overview. *IEEE Circuits and Systems Magazine, 1*(3), 6–21.
23. Baptista, M. S. (1998). Cryptography with chaos. *Physics Letters A, 240*(1–2), 50–54.
24. Xiao, D., Liao, X., & Deng, S. (2005). One-way hash function construction based on the chaotic map with changeable parameter. *Chaos, Solitons & Fractals, 24,* 65–71.
25. Wang, Y., Wong, K., Liao, X., & Xiang, T. (2009). A block cipher with dynamic s-boxes based on tent map. *Communications in Nonlinear Science and Numerical Simulation, 14*(7), 3089–3099.
26. Chen, G., Chen, Y., & Liao, X. (2007). An extended method for obtaining s-boxes based on three-dimensional chaotic Baker maps. *Chaos, Solitons & Fractals, 31,* 571–579.
27. Juang, W. S. (2004). Efficient multi-server password authenticated key agreement using smart cards. *IEEE Transactions on Consumer Electronics, 50*(1), 251–255.
28. Chang, C. C., & Lee, J. S. (2004). An efficient and secure multi-server password authentication scheme using smart card. In *Proceedings of the international conference on cyberworlds* (pp. 417–422).
29. Li, L. H., Lin, I. C., & Hwang, M. S. (2001). A remote password authentication scheme for multi-server architecture using neural networks. *IEEE Transactions on Neural Networks, 12*(6), 1498–1504.
30. Yeh, K. H., & Lo, N. W. (2010). A novel remote user authentication scheme for multi-server environment without using smart cards. *International Journal of Innovative Computing Information and Control, 6*(8), 3467–3478.
31. Lee, J. S., Chang, Y. F., & Chang, C. C. (2008). A novel authentication protocol for multi-server architecture without smart cards. *International Journal of Innovative Computing Information and Control, 4*(6), 1357–1364.
32. Tsai, J. L. (2008). Efficient multi-server authentication scheme based on one-way hash function without verification table. *Computers & Security, 27*(3–4), 115–121.
33. Khan, M. K., & He, D. (2012). A new dynamic identity-based authentication protocol for multi-server environment using elliptic curve cryptography. *Security and Communication Networks, 5*(11), 1260–1266.
34. Li, X., Xiong, Y., Ma, J., & Wang, W. (2012). An enhanced and security dynamic identity based authentication protocol for multiserver architecture using smart cards. *Journal of Network and Computer Applications, 35*(2), 763–769.
35. Yeh, K. H., Lo, N. W., & Li, Y. (2011). Cryptanalysis of Hsiang-Shih's authentication scheme for multi-server architecture. *International Journal of Communication Systems, 24*(7), 829–836.
36. Tsai, J. L., & Lo, N. W. (2015). A chaotic map-based anonymous multi-server authenticated key agreement protocol using smart card. *International Journal of Communication Systems, 28*(13), 1955–1963.
37. Han, W. (2012). Weaknesses of a dynamic identity based authentication protocol for multi-server architecture. arXiv:1201.0883v1, 2012. http://arxiv.org/abs/1201.0883.
38. Tsai, J. L., Lo, N. W., & Wu, T. C. (2013). A new password-based multi-server authentication scheme robust to password guessing attacks. *Wireless Personal Communications*. doi:10.1007/s11277-012-0918-6.
39. Lu, Y., Li, L., Peng, H., & Yang, Y. (2016). Cryptanalysis and improvement of a chaotic maps-based anonymous authenticated key agreement protocol for multiserver architecture. *Security and Communication Networks, 9*, 1321–1330.
40. Dodis, Y., Reyzin, L., & Smith, A. (2004). Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *Advances in Cryptology—EUROCRYPT, 3027,* 523–540. doi:10.1007/978-3-540-24676-3_31.

Azeem Irshad received Master′s degree from Arid Agriculture University, Rawalpindi, Pakistan. Currently, he is persuing his PhD in security for multi-server architectures, from International Islamic University, Islamabad, Pakistan. He authored more than 27 international journal and conference publications, including 12 SCI-E journal publications. 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.

Dr. Muhammad Sher is a Professor having more than 120 scientific publications. He is chairman of the Department of Computer Science and Software Engineering, International Islamic University. He is also Dean of the Faculty of Basic and Applied Sciences. He did his Ph.D. Computer Science from TU Berlin, Germany and M. Sc. from Quaid-e-Azam University, Islamabad. His research interests include Next Generation Networks and Network Security.

Muhammad Usman Ashraf is a Ph.D. scholar at the department of Computer Science & Software Engineering, International Islamic University Islamabad, Pakistan. He is also serving as a Lecturer at the IBMS, University of Agriculture Faisalabad, Pakistan. His multidisciplinary interests include HRI, Distributed Healthcare Systems and Network Security.

Bander A. Alzahrani is an assistance professor at King Abdulaziz University, Saudi Arabia. He completed his M.Sc. in Computer Security (2010), and his Ph.D. in Computer Science (2015), both from Essex University, United Kingdom. His research interests include Network security, Information centric networks, Bloom filter data structure and its applications, secure content routing, Big data privacy (IoT). Bander has published more than 17 research papers in International Journals and conferences.

Fan Wu received Master degree in Computer Software and Theory from Xiamen University, Xiamen, China in 2008. Now he works in Xiamen Institute of Technology. His current research interests include information security, internet protocols, and security of wireless communication.

Qi Xie is a professor in Hangzhou Key Laboratory of Cryptography and Network Security, Hangzhou Normal University, China. He received his PhD degree in applied mathematics from Zhejiang University, China, in 2005. He was a visiting scholar between 2009 and 2010 at Department of Computer Science, University of Birmingham in UK, and a visiting scholar to the Department of Computer Science at City University of Hong Kong in 2012. His research area is applied cryptography, including digital signatures, authentication and key agreement protocols etc. He has published over 60 research papers in international journals and conferences, and served as general co-chair of ISPEC2012 and ASIACCS2013, and a reviewer for over 20 international journals.

Dr. Saru Kumari is currently an Assistant Professor with the Department of Mathematics, Ch. Charan Singh University, Meerut, Uttar Pradesh, India. She received her Ph.D. degree in Mathematics in 2012 from CCS University, Meerut, UP, India. She has published more than 82 research papers in reputed International journals and conferences, including 65 publications in SCI-Indexed Journals. She is a reviewer of more than a dozen of reputed Journals including SCI-Indexed such as IEEE-Transactions on Dependable and Secure computing, IEEE Security and Privacy, Computer Networks, Journal of Network and Computer Applications, Computer and Electrical Engineering, Wireless Personal Communications, Cryptologia, Security and Communication Networks, International Journal of Distributed Sensor Networks, International Journal of Ad Hoc and Ubiquitous Computing, Nonlinear Dynamics and Journal of Medical Systems. She is an Associate Editor of well-reputed journal "KSII Transactions on Internet and Information Systems". Her research interest includes Cryptography and Information Security.