J Supercomput (2016) 72:1623–1644
DOI 10.1007/s11227-016-1688-9

An efficient and anonymous multi-server authenticated key agreement based on chaotic map without engaging Registration Centre

Azeem Irshad[1] · Muhammad Sher[1] · Shehzad Ashraf Chaudhary[1] · Husnain Naqvi[1] · Mohammad Sabzinejad Farash[2]

Published online: 8 March 2016

Abstract Multi-server authentication (MSA) enables the user to avail multiple services permitted from various servers out of a single registration through registration centre. Earlier, through single-server authentication, a user had to register all servers individually for availing the respective services. In the last few years, many MSA-based schemes have been presented; however, most of these suffer communication overhead cost due to the Registration Centre (RC) involvement in every mutual authentication session. In voice communication this round-trip latency becomes even more noticeable. Hence, the focus of the protocols design has been shifted towards light-weight cryptographic techniques such as Chebyshev chaotic map technique (CCM). We have reviewed few latest MSA-related schemes based on CCM and elliptic curve cryptography (ECC) as well. Based on these limitations and considerations, we have proposed a single-round trip MSA protocol based on CCM technique that foregoes the RC involvement during mutual authentication. Our study work is cost efficient in terms

✉ Azeem Irshad
irshadazeem2@gmail.com

Muhammad Sher
m.sher@iiu.edu.pk

Shehzad Ashraf Chaudhary
shahzad@iiu.edu.pk

Husnain Naqvi
husnain.naqvi@iiu.edu.pk

Mohammad Sabzinejad Farash
sabzinejad@khu.ac.ir

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

[2] Faculty of Mathematical Sciences and Computer, Kharazmi University, Tehran, Iran

of communication delay and computation, and provides enhanced security by the use of public key cryptosystem. The proposed scheme is duly backed by formal security analysis and performance evaluation.

Keywords Multi-server authentication · Chebyshev chaotic map · Registration centre · Authentication key agreement

1 Introduction

Single-server user authentication has become unable to meet increasing application demand, as the service demands are expanding with the time. Multi-server authentication, in this regard, can never be underestimated, because a single registration from a Registration Centre (RC) enables the user to avail multiple services from a range of servers in a network. Earlier, through single-server authentication, a user had to seek and register all servers individually for availing the respective services. Majority of the single-server authentication schemes put a restriction on the number of services offered by a network. The multi-server concept relieves the user of more than one authentication with its corresponding server, as the subscriber needs to re-login with its related server using the same password and parameters. The remote internet authentication often entails such type of multi-server authentications, which further underscores the performance and robustness of these protocols. The multi-server scenario consists of three entities, i.e., user, server and Registration Centre. The user registers with RC and avails the services of available servers by getting authenticated from RC.

Generally, a user communicates over a public network, where an adversary finds an open field to intercept publicly available messages and can easily modify, delete or replay the message to launch an attack. This vulnerability of an insecure channel requires the authentication protocols to be technically robust in every security aspect, but still light-weight to be able to run on low-end devices. The authentication schemes are seen to be evolved from low computational techniques (hash, XOR, etc.) to high computational techniques (modular exponentiation, scalar multiplication, chaotic map, and symmetric cryptosystem, etc.) encompassing complex cryptography. The researchers have continuously focused to come up with light-weight cryptographic techniques catering low end devices as well. Beside these light-weight and robust cryptographic tools, the academia also needs to focus on minimizing the communication latency and round-trip delay, in view of the fact that, the messages destined for some destination, have to traverse various nodes in some physical network infrastructure, which adds to the transmission and propagation delay. Hence, we need to bring down the communication cost as well as making the authentication protocols computation-efficient.

Multi-server authentication protocols seek to register at the registration centre and ease out the requirement for recurring authentication [1–5]. We can also sort these protocols out into three sections described as under.

Creative phase This phase covers the early contribution as put forward by Li et al. [6]. Thereafter, Lin et al. [7] commented that Li et al. scheme is inefficient for taking long time for training neural networks. Lin et al. then presented its scheme based on ElGamal digital signature.

Development phase The research, being a continuous activity, makes its way through various developments. In this regard, Tsai [8] proposed a one-way hash function-based multi-server authentication scheme without a stored verification table. Although it was a low-cost scheme for its low-cost operations in the distributed network architecture, it was found susceptible to privileged insider, server-spoofing attacks, and the compromise of perfect forward secrecy.

Diversification phase Now, the focus of research, in almost every authentication domain including MSA, has been shifted to functionality-based techniques. Hence we can see identity-based MSA techniques, dynamic identity-based MSA protocols, bilinear pairing or elliptic curve cryptography (ECC) based MSA schemes [9,10], chaotic map-based MSA schemes [11,12], along with other protocols as well [13–15].

Lately, we can see many MSA schemes [16–21] based on smart card, biometrics and anonymity. In this context, Liao and Wang [16] proposed a dynamic-ID-based authentication protocol and was challenged by Hsiang and Shih [18] for being prone to insider attack, masquerading attacks, and also lacking mutual authentication. Hsiang and Shih then proposed an improved model. Following this, few more schemes were presented for MSA [22–24]. To overcome the weaknesses of these schemes, further schemes were presented based on biometric two-factor authentication [25–27]. However, these protocols also suffer weaknesses like lacking efficiency and anonymity. Thereafter, Chuang and Chen [28] presented a multi-server authentication protocol focusing on privacy. Then, Hao et al. [29] launched spoofing and impersonation attacks on [28], and the scheme could not maintain the perfect forward secrecy. In return, Hao et al. presented an improved model in the wake of above-mentioned flaws. However, Hao et al.'s scheme suffers replay attack, and also lack mutual authentication. All of these MSA schemes suffer various kinds of attacks in one form or another.

Recently, we have scrutinized few state-of-the-art MSA-based schemes [9–12], and to our observation, these schemes are designed in a manner that engages RC in each mutual authentication of a session, hence, increasing the number of round-trips, and communication delay ultimately. We propose a cost-efficient MSA protocol based on the Chebyshev chaotic map that enables the reduction of communication delay from 3–5 round-trips to 2, and also restrains the revoked users by maintaining a Certificate Revocation List (CRL) on the RC's end.

As for the division of this paper, the Sect. 2 describes the preliminaries related to cryptographic techniques. The Sect. 3 provides the review of schemes incurring drawbacks. The Sect. 4 presents our proposed model. The Sect. 5 exhibits the security analysis and performance analysis. Lastly, Sect. 6 concludes the findings.

2 Preliminaries

This section covers the overview of Chebyshev chaotic map and elliptic curve cryptography that are utilized by most of the current schemes.

2.1 Chebyshev chaotic maps

The chaotic map-based authentication protocols can be seen in the research literature and these Chaotic-encryption-based techniques are still being adopted as a tradeoff between security and computational cost. We can see few chaotic map variants, i.e., symmetric, asymmetric, and one-way hash functions, as being used in cryptography; however, most of the chaotic map-based techniques are following symmetric cryptosystems [30]. For better understanding, some of the properties of Chebyshev polynomial and chaotic maps [31] are defined as under:

Definition 1 To describe the first property of Chebyshev polynomial, we assume n as an integer, and a variable x of the interval [−1, 1]. While, we define the Chebyshev polynomial T_n(x): [−1, 1] → [−1, 1] as T_n(x) = cos(n arccos(x)).

The recurrent relation in the above definition can be used to define Chebyshev polynomial map T_n: R → R of degree n, and the Chebyshev polynomial meets the recursive relationship in Eq (1), provided $n \geq 2$, $T_0(x) = 1$, and $T_1(x) = x$.

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \tag{1}$$

The first few Chebyshev polynomials are listed as below:

$$\begin{aligned} T_2(x) &= 2x^2 - 1 \\ T_3(x) &= 4x^3 - 3x \\ T_4(x) &= 8x^4 - 8x^2 + 1 \end{aligned} \tag{2}$$

Definition 2 (*The chaotic feature*) For the second property of Chebyshev polynomial, let say $n \geq 1$, the Chebyshev polynomial map $T_n(x) : [-1, 1] \to [-1, 1]$ of degree n indicates a chaotic map with an invariant density f*(x) = $1/(\pi\sqrt{1 - x^2})$ for all positive Lyapunov exponent ln n.

Definition 3 (*The semi-group feature* [24]) For the third property, i.e., semi-group feature of Chebyshev polynomial can be defined on an interval [−∞, +∞] as defined below:

$$T_n(x) = (2xT_{\mathrm{n}-1}(x) - T_{n-2}(x)) \bmod \mathrm{p} \tag{3}$$

Given that $n \geq 2$, x ϵ [−∞, +∞], and p be a large range prime number. In addition,

$$T_{\mathrm{a}}(T_{\mathrm{b}}(x)) \equiv T_{\mathrm{ab}}(x) \equiv T_{\mathrm{ba}}(x) \equiv T_{\mathrm{b}}(T_{\mathrm{a}}(x))\, modp \tag{4}$$

Definition 4 [*Chaotic map-based discrete logarithm problem* (*CMDLP*)] It is a hard problem to locate s, such that T_{s} (a) = b.

Definition 5 [*Chaotic map-based Diffie–Hellman problem* (*CMDHP*)] It is hard problem to compute$T_{\mathrm{ab}}(x)$, given that $T_{\mathrm{a}}(T_{\mathrm{b}}(x)) = T_{\mathrm{ab}}(x)\, or T_{\mathrm{b}}(T_{\mathrm{a}}(x)) = T_{\mathrm{ba}}(x)$.

2.2 Elliptic curve cryptography (ECC)

The ECC-based security, as introduced by Koblitz [32], provides an efficient cryptographic tool as compared to earlier conventional techniques like (Rivest–Shamir–Adleman) RSA, (Diffie–Hellman) DH and (Digital Signature Algorithm) DSA. This technique provides an equivalent level of security with far less key sizes, i.e., a key size of 160-bit provides an equivalent level of security in ECC as 1024-bit key size does in RSA-based cryptography. This light-weight cryptographic tool is one of the important candidates for the use in state-of-the-art authentication protocols, as it employs point multiplication and addition operations instead of using expensive exponentiation operations as employed in RSA.

Some mathematical operations are drawn over an elliptic curve equation as $E_{\mathrm{p}}(a, b) : y^2 = x^3 + ax + b(modp)$ and $4a^3 + 27b^3 \neq 0(modp)$, Where $a, b \epsilon F_p$ and p is a large prime number. The values a, b defines the elliptic curve, and the points (x, y) that satisfies the former equation embracing a point at infinity lies on this elliptic curve. The scalar multiplication is implemented using repeated additions as $vP = P + P + \ldots P_v$, given a point P and an integer $v \epsilon F_p^*$. All other domain parameters like (p, a, b, G, h and n) belong to finite field, F_p^*. E is an abelian group and the point at infinity serves as an identity element for this group. Here, we define some of the security terms needed to grasp this research work.

1. *Term1* A computational Diffie–Hellman Problem (CDHP) is defined as: given three points P, aP, bP where $a, b \in F_p^*$, it is intractable to compute abP.
2. *Term2* The elliptic curve discrete logarithm problem (ECDLP) is defined as: given a point $Q = aP$ on Elliptic Curve, it is intractable to compute $a \in F_p^*$, assuming two points Q and P over $E(a, b)$.
3. *Term3* The elliptic curve factorization problem (ECFP) is defined as: it is hard to find either of these values, i.e., aP or bP, assuming two points P and $Q = aP + bP$ over $E(a, b)$, where $a, b \in F_p^*$.

3 Review of MSA protocols

This section covers the reviews, working, and drawbacks of recently presented MSA techniques.

3.1 Review of Shen et al. protocol

The review of Shen et al. protocol elaborates the working and drawbacks for the scheme as defined under:

3.1.1 Working of Shen et al. protocol [9]

The Shen et al. scheme [9] consists of three phases: registration, login, and authentication phase. These phases are described as under:

(a) *The server registration phase*

The Shen et al. scheme assumes one trusted RC and n number of trusted servers Sj, where $j = 1 \ldots n$. The Sj is already registered with RC by sharing a secret *Xj* between both of the entities (RC and Sj) using secure channel. Initially, the server Sj sends its identity *SIDj* to RC. RC, then, computes $Xj = h(SIDj||y)$, and sends it to Sj using a secure channel. Here, y acts as the RC master secret.

(b) *The user registration phase*

In this phase, Ui registers with RC, while Sj has already been registered with RC. Thereafter, Ui can access all Sj servers. RC performs with Ui the following steps:

1. The Ui selects identity *IDi* and password *PWi*. Next, it generates a random number *ni*, imprints *Bi* biometric impression, and sends $\{ID_i, Bi, h(PWi||Bi||ni)\}$ to RC.
2. RC computes $Xi = h(IDi||x)$ ×P, Ui=Xi $\oplus$ $h(PWi||Bi||ni)$, and stores {Ui, Bi, d(),v, h()} in smart card. Next, it sends the SC carrying $\{Ui, Bi, d(), v, h\}$. While, P acts as the generator, v is the threshold, $d()$ is the symmetric parametric function, and the symbol '×' represents the point multiplication.
3. Ui receives, and inserts *ni* additionally in smart card.

(c) *The login and authentication phase*

1. In this phase the Ui uses the SC for getting authenticated access to Sj. For this purpose the Ui inputs its *Bi'*, and verifies $d(Bi, Bi') < v$. If the outcome of this function does not exceed the threshold, the SC authenticates affirmative, the biometric*Bi'* as input by*Ui.* Next, it inputs *IDi, PWi*, and generates a random number a. Further, it computes $Xi = Ui \oplus h(PWi||Bi||ni)$, $A = a \times P$, $A' = a \times Xi$, $C_1 = h(IDi||A||A')$. Finally, it sends the message $m_1 = \{IDi, A, C_1\}$ towards Sj.
2. The Sj receives $m_1 = \{IDi, A, C_1\}$ and generates a random number b and compute $Z = b \times P, C_2 = h(IDi||A||C_1||Xj||Z)$. Sj sends the message $m_2 = \{IDi, A, C_1, SIDj, Z, C_2\}$ to RC for further verification.
3. The RC receives the message $m_2 = \{IDi, A, C_1, SIDj, Z, C_2\}$ and computes $C_1' = h(IDi||A||h(IDi||x) \times A)$ and $C_2' = h(IDi||A||C_1||h(SIDj||y)||Z)$. It compares the equations $C_1'? = C_1$ and $C_2'? = C_2$ and checks the authentication of both user and Sj. If found positive, then further computes $V = h(h(SIDj||y)||Z||A)$, $W = h(SIDj||Z||A||h(IDi||x) \times A)$, $C_3 = W \oplus V$, and $C_4 = h(W||V||IDi)$. Finally, it sends the message $m_3 = \{C_3, C_4\}$ to Sj for verification.
4. The Sj receives the message $m_3 = \{C_3, C_4\}$ and computes $V' = h(Xj||Z||A)$, $W' = C_3 \oplus V'$, $C_4' = h(W'||V'||IDi)$, and compares the equality $C_4'? = C_4$. On equality match, it further computes $SKj = b \times A$, $V = h(h(SIDj||y)$ $||Z||A)$, and $C_5 = h(IDi||SIDj||SKj||W')$. It then sends the message $m_4 = \{Z, C_5\}$ to user.
5. The user Ui, receives the message $m_4 = \{Z, C_5\}$, and computes $W'' = h(SIDj||Z||A||A')$, $Ski = a \times Z$, $C_5 = h(IDi||SIDj||SKi||W'')$. Next,

it checks the equality $C_5'? = C_5$ and on positive verification, it computes $C_6 = h(W''||Ski||Z)$, and sends the message $m_5 = \{C_6\}$ to Sj finally.

6. The Sj receives $m_5 = \{C_6\}$ and computes $C_6' = h(W''||Ski||Z)$, and matches the equality $C_6'? = C_6$. If this comes true, then it establishes the final session key as $SKi = SKj = a \times Z = b \times A = ab \times P$.

3.1.2 Inefficiencies and flaws of the Shen scheme

The Shen et al. protocol presents a multi-server authentication scheme based on ECC technique. However, this scheme does not provide anonymity to the user. Secondly, the scheme involves RC involvement in each mutual authentication of a session that renders the scheme too expensive for the extra round-trips it adds into the protocol. The computational resources have been becoming even more powerful with time, in comparison with the infrastructure responsible for the transportation of message; thus, there is a need to design such a scheme that ensures the RC's involvement only up to the registration phase, and not for the later login and authentication procedures, this would significantly reduce the round-trip latency of authentication messages on insecure channel.

3.2 Review of Tsai et al. protocol

The review of Tsai et al. protocol elaborates the working and drawbacks for the scheme as defined under:

3.2.1 Working of Tsai et al. protocol [10]

The Tsai et al. scheme [10] consists of three phases: registration, login, and authentication phase. These phases are described as under:

(a) *The server registration phase*

The Tsai et al. scheme consists of one trusted RC and n number of trusted servers Sj, where $j = 1...n$. The Sj is already registered with RC by sharing a secret Rj between both of the entities (RC and Sj) using secure channel. Initially, the server Sj sends its identity *SIDj* to RC. RC, then, computes *Rj = h(s, SIDj)*, and sends it to Sj using a secure channel. Here, *s* acts as the master key of RC.

(b) *The user registration phase*

In this phase Ui registers with RC, while Sj has already been registered with RC. Consequently, Ui can access all Sj servers. RC performs with Ui the following steps:

1. The Ui selects identity *IDi* and password *PWi*. Next, it generates a random number n and sends $\{ID_i, h(IDi, PWi, n)\}$ to RC.
2. RC computes $CIDi = (IDi, r) \oplus h(s)$, $Ri = h(IDi, s) \oplus h(IDi, PWi, n)$ and stores $\{CIDi, Ri, h()\}$ in smart card. Next, it sends the SC to Ui. Here, r acts as a random number generated by RC.

3. Ui receives, and stores n additionally in smart card.

(c) *Login and authentication phase*

1. In this phase the Ui computes $h(IDi||s) = Ri \oplus h(IDi, PWi, n), q = h(h(ID, s), CIDi, SIDj), C_1 = h(CIDi, SIDj, h(IDi, s)) \oplus T_a(q)$, and $V_1 = h(CIDi, SIDj, h(IDi, s), T_a(q))$. Next, it sends the message $\{CIDi, SIDj, C_1, V_1\}$ to Sj.
2. The Sj receives $\{CIDi, SIDj, C_1, V_1\}$ and compute $V_2 = h(CIDi, SIDj, C_1, V_1, Rj)$, and sends the message $\{CIDi, SIDj, C_1, V_1, V_2\}$ to RC for further verification.
3. The RC receives the message $\{CIDi, SIDj, C_1, V_1, V_2\}$ and computes $(IDi, r) = CIDi \oplus h(s), h(IDi, s), q = h(h(ID, s), CIDi, SIDj), T_a(q) = h(CIDi, SIDj, h(IDi, s)) \oplus C_1, Rj = h(SIDj, s), V_1 = h(CIDi, SIDj, h(IDi, s), T_a(q))$, and $V_2 = h(CIDi, SIDj, C_1, V_1, Rj)$. Next, it compares the equation equality $V_1'? = V_1, V_2'? = V_2$. If true, then further computes $CIDi' = (IDi, r') \oplus h(s), V_3 = (IDi, q, T_a(q)) \oplus h(SIDj, Rj, CIDi, V_1, V_2), V_4 = CIDi' \oplus h(h(ID, s), CIDi, IDi), V_5 = h(SIDj, IDi, Rj, q, V_3, V_4)$, and finally sends the message $\{V_3, V_4, V_5\}$ to Sj for verification.
4. The Sj computes $(IDi, q, T_a(q)) = V_3 \oplus h(SIDj, Rj, CIDi, V_1, V_2), V_5' = h(SIDj, IDi, Rj, q, V_3, V_4)$, and compares the values $V_5'? = V_5$. If successful, then compute $V_6 = q \oplus T_b(q), SKj = h(T_{ba}(q)), V_7 = h(SKj, q, T_b(q), V_4, V_6)$, and sends the message $\{V_4, V_6, V_7\}$ to Ui for verification.
5. The user Ui, receives the message $\{V_4, V_6, V_7\}$, and computes $CIDi' = V_4 \oplus h(h(ID, s), CIDi, IDi), T_b(q) = q \oplus V_6, SKi = h(T_{ab}(q)), V_7' = h(SKi, q, T_b(q), V_4, V_6)$. It then compares $V_7'? = V_7$. If found true, computes $V_8 = h(CIDi, Ski, q, V_4, T_b(q))$, and sends $\{V_8\}$ to Sj for final verification.
6. The Sj computes $V_8' = h(CIDi, Skj, q, V_4, T_b(q))$, and matches the equality $V_8'? = V_8$. If this comes true, then it establishes the final session key as $Ski = SKj = h(T_{ab}(q)) = h(T_{ba}(q))$.

3.2.2 Inefficiencies and flaws of the Tsai et al. scheme

The Tsai et al. protocol presents a multi-server authentication scheme based on chaotic map technique. The Tsai scheme also engages RC in each mutual authentication of a session that adds communication delay for the extra round-trips. The scheme's communication delay can be minimized if we eliminate the RC entity for the login and authentication phases in the protocol, however, with a bit extra computational cost, as shown in Table 3.

3.3 Review of Jiang et al. protocol

The review of Jiang et al. protocol elaborates the working and drawbacks for the scheme as defined under:

3.3.1 Working of Jiang et al. protocol [11]

The Jiang et al. scheme [11] consists of three phases: registration, login, and authentication phase. These phases are described as under:

(a) *Server registration phase*

The Jiang et al. scheme assumes one trusted RC and n number of trusted servers Sj, where $j = 1\ldots n$ in a network. The Sj gets registered with RC by sharing two secret values between both of the entities (RC and Sj) using a secure channel. Initially, RC generates a master key s and random secret t. Next, the server Sj sends its identity *SIDj* to RC. RC, then, computes $h(SIDj||t)$ and $h(s||t)$. Now, RC sends both of these computed parameters to Sj using a secure channel.

(b) The user registration phase

In this phase Ui registers with RC, while Sj has already been registered with RC. Thus, Ui can access all Sj servers. RC performs with Ui the following steps:

1. The Ui selects identity *IDi* and password *PWi*. Next, it generates a random number r, and *Bi* biometric impression, and compute $TPWi = h(PWi||h(IDi)||r)$. Next, it sends $\{ID_i, TPWi, Bi\}$ to RC.
2. RC computes $A = h(IDi||s), B = A \oplus TPWi, C = B \oplus h(s), Gen(Bi) = (R, Q), D = h(IDi||TPWi||R), E = B \oplus h(t), M = A \oplus h(s), Temp = Enc_{H(IDi)}(template)$ and stores $\{C, D, E, h(), Temp\}$ in smart card and sends it to Ui. Where, *Gen, Rep* functions are fuzzy extractors, and the *template* function is used for biometric verification.
3. Ui receives the SC and stores random value r additionally.

(c) Login and authentication phase

1. In this phase the Ui uses the SC for getting authenticated access to *Sj*. For this purpose the Ui inputs IDi, PWi, Bi and obtains the template. Then it computes $TPWi = h(PWi||h(IDi)||r)$, $R = Rep(Bi, Q)$, and $D^* = h(IDi||TPWi||R)$. Next, it checks the equality $D^*? = D$. If successful, then it generates a random number x and computes*xP* using point multiplication (ECC). Further, it computes $M = TPWi \oplus C, N = TPWi \oplus E, P_1 = M \oplus SIDj, P_2 = N \oplus xP, P_3 = h(P_1||P_2), T = h(M||SIDj), CIDi = TPWi \oplus h(M||xP)$ and $Wij = h(TPWi||SIDj)$. Finally, it sends the message $\{CIDi, Wij, xP, P_3, T\}$ to *Sj* for verification.
2. The Sj receives $\{CIDi, Wij, xP, P_3, T\}$ and compute $V_1 = h(CIDi||xP)$, and stores $(CIDi, V_1)$ to resist replay attack and Man-in-the-Middle attack, in future. Next, it generates a random number y and *yP.* Then it computes $P_4 = h(h(SIDj||t)||h(s||t)) \oplus yP$, and sends the message $\{T, xP, P_3, yP, P_4\}$ to *RC* for Ui's verification.
3. The RC receives the message $\{T, xP, P_3, yP, P_4\}$ and computes $M' = A \oplus h(s)$, $P_1' = M' \oplus SIDj, P_2' = A \oplus h(t) \oplus xP, P_3' = h(P_1'||P_2'), P_4' = h(h(SIDj||t) ||h(s||t)) \oplus yP$. Next, it checks the equality $P_3'? = P_3$, and $P_4'? = P_4$ to authenticate user and server. If successful, then compute $P_5 = h(h(SIDj||t)||yP \oplus$

$xP, P_6 = h(M'||xP) \oplus yP, P_7 = P_5 \oplus P_6$ and $P_8 = h(P_5||P_6)$. Finally, it sends the message $\{P_7, P_8\}$ to Sj.

4. The Sj receives the message $\{P_7, P_8\}$ and computes $P_5' = h(h(SIDj||t)||yP \oplus xP, P_6' = h(M'||xP) \oplus yP, P_8' = h(P_5'||P_6')$, and checks the equality $P_8'? = P_8$ for authenticating RC. If successful, then further compute $RPWi' = P_6' \oplus CIDi \oplus yP, Wij' = h(TPWi||SIDj)$, and checks again the equation $Wij'? = Wij$. Then it computes $SK = y(xP), P_9 = h(CIDi||SIDj||SK||P_6')$, and sends the message $\{P_9, yP\}$ to Ui for verification.
5. The user Ui, receives the message $\{P_9, yP\}$ and computes $P_6'' = CIDi \oplus TPWi \oplus yP, SK = x(yP)$ and $P9 = h(CIDi||SIDj||SK||P6'')$. Finally, it compares the equation $P_9'? = P_9$. If matches the equality, then it establishes the session key as $SK = x(yP)$.

3.3.2 Inefficiencies and flaws of the Jiang et al. scheme

The Jiang et al. protocol presents a multi-server authentication scheme based on chaotic map technique. The limitations of the scheme are given below.

(a) The Jiang scheme does not provide resistance to location traceability, as the two parameters *T* and *Wij* in login request $\{CIDi, Wij, xP, P_3, T\}$, remains the same for all sessions.

(b) Secondly, it engages RC in each mutual authentication of a session that adds communication delay for the extra round-trips. The scheme's communication latency can be minimized if we eliminate the RC entity for the login and authentication phase of mutual authentication.

(c) Lastly, the Jiang et al. scheme stores the verifiers in RC's database. Assuming the attacker, being a malicious legal insider, could launch stolen-verifier attack on both ends, i.e., it could impersonate the user as well as server.

   1. On the user side, if it steals the $\{A, T\}$ verifiers from RC's database, it could construct this login request message $\{CIDi, Wij, xP, P_3, T\}$ successfully. For being a legal user, this adversary could construct *M* by using the stolen *A*. Then, it derives *TPWi* from previous *CIDi* and further computes $P_1 = M \oplus SIDj, N = TPWi \oplus E$. Next, it assumes random number *x*, computes $P_2 = N \oplus xP, P_3 = h(P_1||P_2), T = h(M||SIDj), CIDi = TPWi \oplus h(M||xP)$ and $Wij = h(TPWi||SIDj)$. Finally sends the $\{CIDi, Wij, xP, P_3, T\}$ successfully.
   2. On the server side, it could impersonate the user through sending the manufactured *{P9, yP}* message by constructing $P_9 = h(CIDi||SIDj||SK||P_6')$ after having calculated $P_6' = h(M'||xP) \oplus yP$ by assuming *yP*.

3.4 Review of Zhu protocol

The review of Zhu et al. protocol elaborates the working and drawbacks for the scheme as defined under:

3.4.1 Working of Zhu et al. protocol [12]

Zhu et al.'s scheme [12] consists of two phases: the server registration, and user login and authentication phase. These phases are described as under:

(a) *Server registration phase*

In server registration phase, each server $\mathrm{S_x}$ gets registered with RC by verifying its identity. For this, all of the servers meant to provide services, must have their identities verified by the RC. In this regard, the server $\mathrm{S_x}$ sends its identity ID_{S_x} to RC. RC computes $R = H(ID_{S_\mathrm{x}}||K_y)$ and sends R to S_x using a secure channel. Here, K_y acts as the server's master key.

One thing must be noted here, that there is only server registration in this scheme and no user registration, as this scheme is meant for one-way authentication of server $\mathrm{S_x}$, and the $\mathrm{S_x}$ does not authenticate the user but just provides the services anonymously to the user without getting registered and authenticated.

(b) *Login and authentication phase*

1. In the login and authentication phase, a random integer a as a user secret, is generated. Then user develops its public key as $T_a(x)$, and shared key $K_{U-RC} = T_aT_{Ky}(x)$, using Chebyshev chaotic map, and further computes $H_U = H(SID_U||ID_{S_x}||T_a(x)), C_1 = E_{KU-RC}(SID_U||ID_{S_x}||H_U)$. The user sends message $m_1 = \{SID_U, T_a(x), C_1\}$ to S_x, finally. Here, ID_{S_x} and ID_{RC} act as the server $\mathrm{S_x}$ and RC's identity, respectively.
2. Next, $\mathrm{S_x}$ receives $m_1 = \{SID_U, T_a(x), C_1\}$ and generates a random integer b, and compute $T_b(x)$. Then computes $C_2 = H(m_1||ID_{Sx}||T_b(x)||R)$ using hash function $H(.)$. Finally, S_x sends $m_2 = \{m_1, T_b(x), ID_{S_\mathrm{x}}, C_2\}$ to RC for verification.
3. After receiving $m_2 = \{m_1, T_b(x), ID_{S_x}, C_2\}$ from S_x, RC computes $R' = H(ID_{Sx}||K_y)$, $C_2' = H(m_1||ID_{S_x}||T_b(x)||R')$, and checks the equation $C_2'?C_2$. If true, then it further computes $K_{RC-U} = T_{Ky}T_a(x)$, and decrypts using K_{RC-U} as $D_{KRC-U}(C_1) = (SID_U||ID_{S_x}||H_U)$. Next, it computes $H_U' = H(SID_U||ID_{S_x}||T_a(x))$ and confirms the equality $H_U'? = H_U$, and check if ID_{S_x} in C_1 equates the ID_{S_x} in plaintext. If true, then further computes $C_3 = H(ID_{RC}||ID_{S_x}||m_1||R'||T_b(x))$, $H_{RC} = H(SID_U||ID_{RC}||ID_{S_x}||T_b(x))$ and $C_4 = E_{KRC-U}(ID_{RC}||ID_{S_x}||m_1||T_b(x)||H_{RC})$. Finally, it sends the message $\{ID_{S_x}, C_4\}$ to Ui, and $\{ID_{S_x}, C_3\}$ to S_x. Here, $E_\mathrm{k}(.)$ acts as an encryption function.
4. $\mathrm{S_x}$, on the receipt of message $\{ID_{S_x}, C_3\}$, computes $C_3' = H(ID_{RC}||ID_{Sx}||m_1||R'||T_b(x))$ and compares the equality $C_3'? = C_3$. After positive verification, it establishes the shared session key as $SK = T_bT_a(x)$,
5. The user receives the message $\{ID_{S_x}, C_4\}$ simultaneously in the same round trip as S_x receives $\{ID_{S_x}, C_3\}$. Next, it uses K_{U-RC} to decrypt C_4 and compute $H_{RC}' = H(SID_U||ID_{RC}||ID_{S_x}||T_b(x))$. Then it compares the equation $H_{RC}'? = H_{RC}$. On positive verification check, it establishes the session key as $SK = T_aT_b(x)$ shared with the S_x.

Table 1 Notations description

Notations	Description
$ID_{U,} PWi, Bi$	User's identity, password, and biometric
K	RC's master key
u, PK_{Ui}	User's private key, public key
$SIDj, s, PK_{Sj}$	Server's identity, private key, and public key
X	The seed generating Chebyshev chaotic map
$T_K(.)$	Chebyshev chaotic map
a, b	The random numbers as degree for Chebyshev chaotic map
T_1, T_2, T_3, T_4	Timestamps
$h(.)$	hash function
? =	Equality comparison
$\oplus$, ||	XOR, concatenation

3.4.2 Inefficiencies and flaws of the Zhu et al. scheme

The Zhu et al. protocol presents a multi-server authentication scheme based on chaotic map technique. The Zhu scheme, like earlier schemes also engages RC in each mutual authentication of a session that adds communication delay for the extra round-trips, similarly. The scheme's communication delay can be minimized on the same lines, at a bit extra computational cost, if we eliminate the RC entity for the login and authentication phase of mutual authentication as shown in Table 3.

4 Proposed model

The proposed model has been presented with a motivation to come up with a novel protocol that may counter the identified threats and limitations of the schemes, as reviewed above [9–12]. The scheme makes a use of a few notations as mentioned in Table 1.

The proposed model consists of four phases, the server registration phase, user registration phase, login and authentication phase and password update phase.

4.1 Server registration phase

In this phase, the server Sj gets registered with RC by sending its identity SIDj. RC computes $s = SK_{S_j} = h(SIDj||k)$ using its master key k. This s serves as the private key of Sj. Then RC, further computes the public key PK_{S_j} by computing $T_s(x)$. Next, it sends the message $\{s, PK_{S_j}\}$ securely to Sj as referred in Fig. 1. The Sj receives its private and public key, while publishes two parameters in public directory as $SIDj$ and PK_{S_j}, i.e., $T_s(x)$.

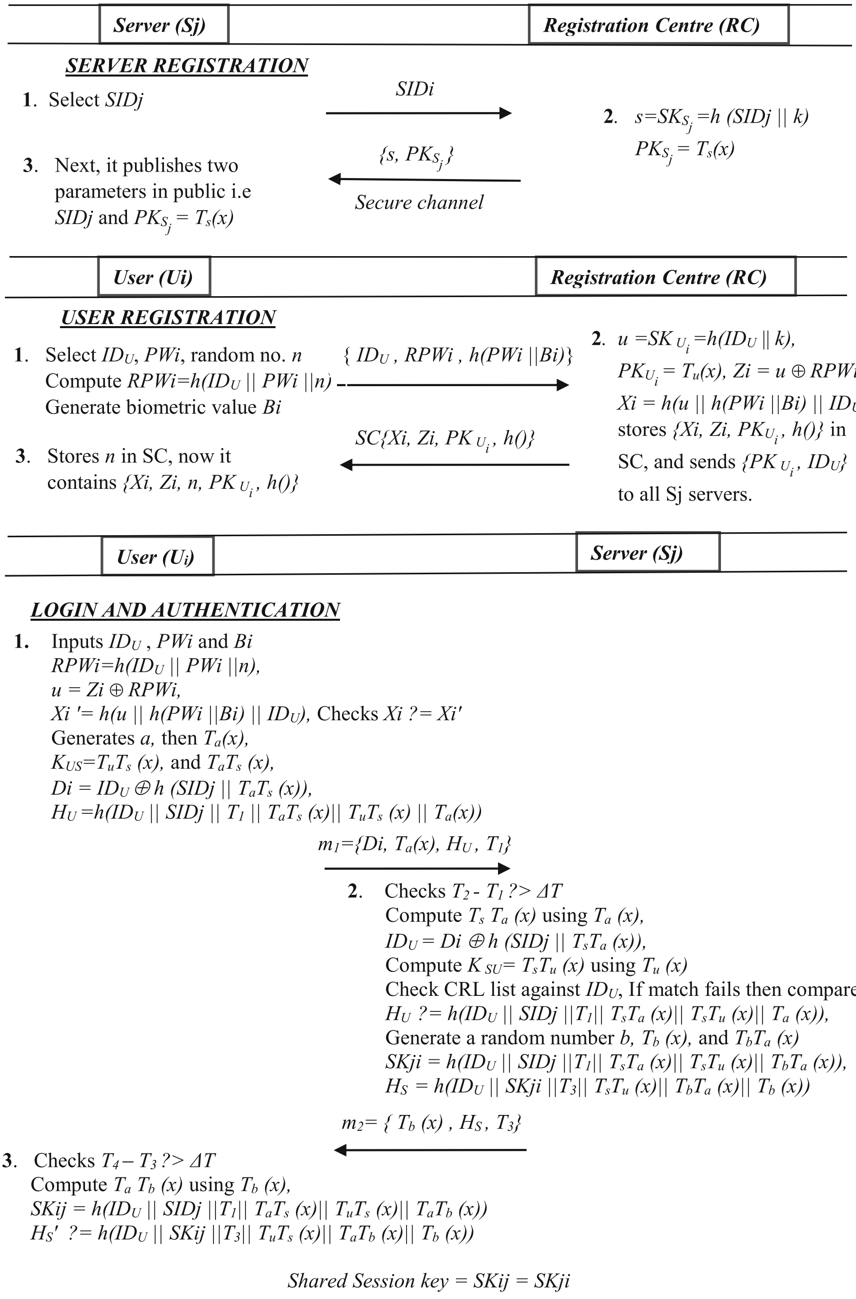
4.2 User registration phase

1. In this phase, the user also gets registered with RC using a secure channel, and employing Chebyshev chaotic map-based architecture. Initially, the user selects ID_U, PWi, random number n and *c*omputes $RPWi = h(ID_{\text{U}}||PWi||n)$. Then it generates biometric value *Bi,* and sends $\{ID_U, RPWi, h(PWi||Bi)\}$ to RC for registration.
2. RC, then computes $u = SK_{\text{Ui}} = h(ID_U||k)$ and $PK_{\text{Ui}} = T_u(x)$. Next, it computes $Zi = u \oplus RPWi$, $Xi = h(u||h(PWi||Bi)||ID_U)$. It finally stores and sends SC $\{Xi, Zi, PK_{\text{Ui}}, h()\}$ to Ui. Then, it sends the $\{PK_{U_i}, ID_U\}$ to all servers Sjs.
3. The Ui receives the SC and stores n in SC to conclude the user registration phase.

Although transmitting those parameters to all servers might be a drawback for a large network domain; however, once transmitted safely, onwards, it will provide authenticated access of services to the intended users.

4.3 Login and authentication phase

1. In the login and authentication phase the user inputs its identity, PWi and Bi. Then it computes $RPWi = h(ID_{\text{U}}||PWi||n), u = Zi \oplus RPWi, Xi' = h(u|| h(PWi||Bi)||ID_U)$, and checks the equality for $Xi? = Xi'$. If matches positively, then it generates a random number a,ad computes $T_a(x)$, $T_aT_s(x)$, and $K_{US} = T_uT_s(x)$. Next, it computes $Di = ID_U \oplus h(SIDj||T_aT_s(x))$, $H_U = h(ID_U||SIDj||T_1||T_aT_s(x)||T_uT_s(x)||T_a(x))$ and sends the message $m_1 = \{Di, T_a(x), H_U, T_1\}$ to Sj for authentication as shown in Fig. 1.
2. The Sj receives the message $m_1 = \{Di, T_a(x), H_U, T_1\}$ and checks the timestamp against the threshold ΔT, i.e., $T_2 - T_1? > \Delta T$. If the difference is more than ΔT, then it aborts the session. Otherwise, it computes $T_sT_a(x)$ using $T_a(x)$, $ID_U = Di \oplus h(SIDj||T_sT_a(x))$ and $K_{SU} = T_sT_u(x)$ using $T_u(x)$. Then, it matches the recovered ID_U with the CRL list as published by the RC. If the equality check for this comparison fails, then it maintains the fact that the registered user (ID_U) still has a valid certificate and not yet revoked. (If the CRL check hits, it sends the negative acknowledgement to the user for an expired certificated). Next, it locates the public key $T_u(x)$ against ID_U in the repository, maintained for registered users. Then, it computes and compares the equation $H_U? = h(ID_U||SIDj||T_1||T_sT_a(x)||T_sT_u(x)||T_a(x))$. If it is true, then it generates a random number b, and computes $T_b(x)$, and $T_bT_a(x)$. Next, it computes $SKji = h(ID_U||SIDj||T_1||T_sT_a(x)||T_sT_u(x)||T_bT_a(x))$ and $H_S = h(ID_U||SKji||T_3||T_sT_u(x)||T_bT_a(x)||T_b(x))$. After complete verification, it sends the message $m_2 = \{T_b(x), H_S, T_3\}$ to Ui, finally.
3. The Ui receives the message $m_2 = \{T_b(x), H_S, T_3\}$, and checks the timestamp difference*as* $T_4 - T_3? > \Delta T$. If the difference exceeds the threshold, it aborts the session. Otherwise, computes $T_aT_b(x)$ using $T_b(x)$, $SKij = h(ID_U||SIDj||T_1 ||T_aT_s(x)||T_uT_s(x)||T_aT_b(x))$. Then, it matches the equality for $H_S'? = h(ID_U|| SKij||T_3||T_uT_s(x)||T_aT_b(x)||T_b(x))$. If the equality holds true, then it authenti-



Server (Sj) **Registration Centre (RC)**

SERVER REGISTRATION

1. Select *SIDj*

$\xrightarrow{SIDi}$

2. $s=SK_{S_j}=h(SIDj \,||\, k)$
$PK_{S_j}=T_s(x)$

$\xleftarrow[\text{Secure channel}]{\{s, PK_{S_j}\}}$

3. Next, it publishes two parameters in public i.e *SIDj* and $PK_{S_j}=T_s(x)$

User (Ui) **Registration Centre (RC)**

USER REGISTRATION

1. Select ID_U, PWi, random no. n
Compute $RPWi=h(ID_U \,||\, PWi \,||n)$
Generate biometric value Bi

$\xrightarrow{\{ID_U, RPWi, h(PWi \,||Bi)\}}$

2. $u=SK_{U_i}=h(ID_U \,||\, k)$,
$PK_{U_i}=T_u(x)$, $Zi=u \oplus RPWi$,
$Xi=h(u \,||\, h(PWi \,||Bi) \,||\, ID_U)$,
stores $\{Xi, Zi, PK_{U_i}, h()\}$ in SC, and sends $\{PK_{U_i}, ID_U\}$ to all Sj servers.

$\xleftarrow{SC\{Xi, Zi, PK_{U_i}, h()\}}$

3. Stores n in SC, now it contains $\{Xi, Zi, n, PK_{U_i}, h()\}$

User (Ui) **Server (Sj)**

LOGIN AND AUTHENTICATION

1. Inputs ID_U, PWi and Bi
$RPWi=h(ID_U \,||\, PWi \,||n)$,
$u=Zi \oplus RPWi$,
$Xi'=h(u \,||\, h(PWi \,||Bi) \,||\, ID_U)$, Checks $Xi ?= Xi'$
Generates a, then $T_a(x)$,
$K_{US}=T_uT_s(x)$, and $T_aT_s(x)$,
$Di=ID_U \oplus h(SIDj \,||\, T_aT_s(x))$,
$H_U=h(ID_U \,||\, SIDj \,||\, T_1 \,||\, T_aT_s(x) \,||\, T_uT_s(x) \,||\, T_a(x))$

$\xrightarrow{m_1=\{Di, T_a(x), H_U, T_1\}}$

2. Checks $T_2 - T_1 ?> \Delta T$
Compute $T_s T_a(x)$ using $T_a(x)$,
$ID_U=Di \oplus h(SIDj \,||\, T_sT_a(x))$,
Compute $K_{SU}=T_sT_u(x)$ using $T_u(x)$
Check CRL list against ID_U, If match fails then compares
$H_U ?= h(ID_U \,||\, SIDj \,||T_1|| \, T_sT_a(x) \,||\, T_sT_u(x) \,||\, T_a(x))$,
Generate a random number b, $T_b(x)$, and $T_bT_a(x)$
$SKji=h(ID_U \,||\, SIDj \,||T_1|| \, T_sT_a(x) \,||\, T_sT_u(x) \,||\, T_bT_a(x))$,
$H_S=h(ID_U \,||\, SKji \,||T_3|| \, T_sT_u(x) \,||\, T_bT_a(x) \,||\, T_b(x))$

$\xleftarrow{m_2=\{T_b(x), H_S, T_3\}}$

3. Checks $T_4 - T_3 ?> \Delta T$
Compute $T_a T_b(x)$ using $T_b(x)$,
$SKij=h(ID_U \,||\, SIDj \,||T_1|| \, T_aT_s(x) \,||\, T_uT_s(x) \,||\, T_aT_b(x))$
$H_S' ?= h(ID_U \,||\, SKij \,||T_3|| \, T_uT_s(x) \,||\, T_aT_b(x) \,||\, T_b(x))$

Shared Session key = SKij = SKji

Fig. 1 Proposed multi-server authentication protocol

cates the Sj positively by establishing the session key as $SKij = h(ID_U||SIDj||T_1||T_aT_s(x)||T_uT_s(x)||T_aT_b(x))$. However, if it receives the negative acknowledgement, it will have to abort the session.

4.4 Password update phase

The user Ui may update its password PWi without consulting RC by initiating the following procedure.

1. In password update phase the user inputs IDi, PWi and Bi. Then it computes $RPWi = h(\mathrm{ID_U}||PWi||n)$, $u = Zi \oplus RPWi$, $Xi' = h(u||h(PWi||Bi)||ID_U)$, and checks the equality for $Xi? = Xi'$.
2. If matches positively, then it selects a password PWi^{new} and computes $RPWi = h(ID_U||PWi^{\mathrm{new}}||n)$, $Zi = u \oplus RPWi$, and $Xi = h(u||h(PWi^{\mathrm{new}}||Bi)||ID_U)$.
3. Next, it stores the updated contents $\{Xi, Zi, PK_{Ui}, h()\}$ in SC.

5 Security analysis

This section shows the security proof, formal security analysis, and performance efficiency analysis.

5.1 Security proof

The proposed scheme is immune to various threats, as elaborated below:

1. *Mutual authentication*
   The mutual authentication defines that both entities authenticate each other in the same authentication protocol. The proposed scheme provides mutual authentication, as the Sj authenticates Ui on the basis of $K_{SU} = T_s T_u(x)$ which was constructed using $T_u(x)$ of ID_U, and the comparison of equality check $H_U? = h(ID_U||SIDj||T_1||T_s T_a(x)||T_s T_u(x)||T_a(x))$. This way, the Sj authenticates the Ui, as the parameter $T_s T_u(x)$ can only be constructed by the legitimate user, who got registered from RC. Likewise, the user Ui authenticates Sj on the basis of checking the equality for $H_S'? = h(ID_U||SKij||T_3||T_u T_s(x)||T_a T_b(x)||T_b(x))$, while $SKij$ is constructed by computing $h(ID_U||SIDj||T_1||T_a T_s(x)||T_u T_s(x)|| T_a T_b(x))$. As, an attacker can never approach the private key s of Sj, and $T_u T_s(x)$ in $SKij$ and H_S' can only be generated by a legitimate server Sj, both entities authenticate mutually each other.
2. *Impersonation attack/man-in-the-middle attack*
   This attack could be initiated by an attacker who acts as silent intermediary between the intended participants and let the other participant perceive it as the legitimate participant. The proposed scheme stands secure. An adversary cannot reproduce $H_U = h(ID_U||SIDj||T_1||T_a T_s(x)||T_u T_s(x)||T_a(x))$ or $H_S = h(ID_U||SKji||T_3||T_s T_u(x)||T_b T_a(x)||T_b(x))$ with updated timestamps T_1 and T_3, as, an attacker does not know about the secret keys u(user) and s (server) to disclose $T_u T_s(x)$, and construct the legitimate H_U and H_S values. The attacker may construct all other values in H_U and H_S except $T_s T_u(x)$, which requires the knowledge of either u or s to reconstruct it.

3. *Replay attack*
   The replay attacks can be launched while an attacker replays the original message parameters at some other time to betray or impersonate any legal participant. In the proposed scheme the messages $m_1 = \{Di, T_a(x), H_U, T_1\}$ and $m_2 = \{T_b(x), H_S, T_3\}$ are publicly available on public channel. An attacker might try m_1 and m_2 to use to launch replay attacks. However, the replay attack in our proposed model can be easily thwarted as an adversary cannot reproduce $H_U = h(ID_U||SIDj||T_1||T_aT_s(x)||T_uT_s(x)||T_a(x))$ with an updated T_1 timestamp. The Sj on the receipt of message m_1, checks the timestamp against T_2 or threshold i.e., $T_2 - T_1? > \Delta T$. If the difference exceeds this threshold it shall abort the session. On the Ui's end, the replay attack cannot be possible because, if an attacker replays m_2, then it would not be able to meet the equality check $H'_S? = h(ID_U||SKij||T_3||T_uT_s(x)||T_aT_b(x)||T_b(x))$ on the user side, and would be easily thwarted, given that, an attacker cannot construct m_2 with an updated timestamp.
4. *Known-key security*
   The known-key security means to guess the private secret keys of the involved participants, provided the session key has been compromised by an adversary. If the shared session key $SKij = SKji = h(ID_U||SIDj||T_1||T_aT_s(x)||T_uT_s(x)|| T_aT_b(x))$ is exposed, it will not lead to any extraction or guessing of Sj or Ui's secrets, i.e., s, u or password PWi,as it is a hard problem to extract s or u from $T_s(x)$ or $T_u(x)$. If the $SKij$ is leaked then the attacker cannot guess any of the secrets from the publicly available parameter $H_S = h(ID_U||SKji||T_3||T_sT_u(x)||T_bT_a(x)||T_b(x))$.
5. *Perfect forward secrecy*
   The perfect forward secrecy means to ensure the secrecy of previous session keys, if the long-term private key of either a user or a server is compromised. The proposed scheme maintains perfect forward secrecy, as the disclosure of secret keys, u(user) and s (server) can only disclose $T_uT_s(x)$, but not $T_aT_b(x)$ in the session key $SKij = h(ID_U||SIDj||T_1||T_aT_s(x)||T_uT_s(x)||T_aT_b(x))$.The reproduction of $T_aT_b(x)$ requires the knowledge of a or b,which are random numbers generated randomly, and cannot be guessed in polynomial time or accessed easily. Hence, the proposed scheme provides complete forward secrecy.
6. *Resistance to password guessing/stolen smart card attack*
   In guessing attacks, an adversary tries to approach all public messages available; which are exchanged on insecure channel among concerned parties, and derive information with the input of all possible combinations by applying brute force attack. In proposed scheme, the password PWi is used in the $RPWi$, Zi and Xi functions. If the smart card gets stolen, attacker may access Zi and Xi,but it may not extract PWi from $Zi = u \oplus RPWi, Xi = h(u||h(PWi||Bi)||ID_U)$, as for extracting PWi it needs u, Bi and ID_U parameters.
7. *Session key security*
   The session key security indicates that the established session key is only known among the legal participants, i.e., U_i and *Sj*, and nobody else. In proposed scheme, an adversary cannot impersonate and masquerade with the session, as long as it does not have the knowledge of legitimate secrets and passwords. The legit-

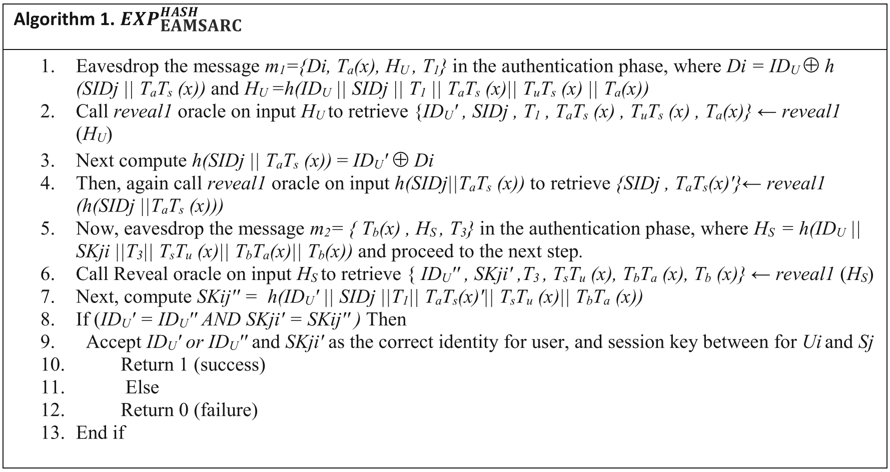
imate session key $SKij = SKji = h(ID_U||SIDj||T_1||T_aT_s(x)||T_uT_s(x)||T_aT_b(x))$cannot be constructed without having the knowledge of at least a or b,and u or s secrets.Hence, our scheme provides session key security.

8. *Anonymity*
   The anonymous authentication provides anonymity to U_is along with its authentication to Sj, and attacker cannot tell the identity of the communicating participants by approaching publicly open message parameters. The user and server exchange includes $m_1 = \{Di, T_a(x), H_U, T_1\}$ and $m_2 = \{T_b(x), H_S, T_3\}$ messages anonymously, in proposed scheme. An attacker is not able to recover the ID_U of user from Di, H_U and H_S parameters. Hence the proposed scheme assures privacy to the user Ui.
9. *Service access to only privileged non-revoked users*
   The proposed scheme maintains a certificate revocation list (CRL) on the RC's end. The RC regularly updates and publishes this CRL list so that the corresponding service providers SPjs may consult the CRL list and validate the users' status before authenticating these users. Whenever, an SPj receives a login request, it consults a CRL for verifying the user's revocation status. In this manner, any of the revoked users having its identity listed in CRL, will not be able to avail the services of a server, and shall be negatively acknowledged upon login request.

5.2 Formal security analysis

This subsection describes the formal security analysis for the proposed model. Using random oracle model, we conduct a formal security analysis to prove that the proposed scheme has been secure [33]. For this objective, we use a *reveal1* oracle as under:

Reveal1 The *reveal1* oracle outputs x from the corresponding hash value $y = h(x)$, unconditionally.



Algorithm 1. $EXP^{HASH}_{\mathbf{EAMSARC}}$

```
Eavesdrop the message $m_1=\{Di, T_a(x), H_U, T_1\}$ in the authentication phase, where $Di = ID_U \oplus h(SIDj || T_aT_s(x))$ and $H_U = h(ID_U || SIDj || T_1 || T_aT_s(x) || T_uT_s(x) || T_a(x))$
Call reveal1 oracle on input $H_U$ to retrieve $\{ID_U', SIDj, T_1, T_aT_s(x), T_uT_s(x), T_a(x)\} \leftarrow reveal1(H_U)$
Next compute $h(SIDj || T_aT_s(x)) = ID_U' \oplus Di$
Then, again call reveal1 oracle on input $h(SIDj||T_aT_s(x))$ to retrieve $\{SIDj, T_aT_s(x)'\} \leftarrow reveal1(h(SIDj||T_aT_s(x)))$
Now, eavesdrop the message $m_2=\{T_b(x), H_S, T_3\}$ in the authentication phase, where $H_S = h(ID_U || SKji ||T_3|| T_sT_u(x)|| T_bT_a(x)|| T_b(x))$ and proceed to the next step.
Call Reveal oracle on input $H_S$ to retrieve $\{ID_U'', SKji', T_3, T_sT_u(x), T_bT_a(x), T_b(x)\} \leftarrow reveal1(H_S)$
Next, compute $SKij'' = h(ID_U' || SIDj ||T_1|| T_aT_s(x)'|| T_sT_u(x)|| T_bT_a(x))$
If ($ID_U' = ID_U''$ AND $SKji' = SKij''$) Then
  Accept $ID_U'$ or $ID_U''$ and $SKji'$ as the correct identity for user, and session key between for $Ui$ and $Sj$
        Return 1 (success)
         Else
        Return 0 (failure)
End if
```

Theorem 1 *By undertaking the chaotic map-based discrete logarithm problem (CMDLP) assumption, the proposed scheme stands secure, in case an attacker approaches the public messages* $\{m_1, m_2\}$ *and tries to find the legitimate session key, if one-way hash function h(.) behaves closely as random oracle.*

Proof In this proof, an attacker $\not{A}$ having access to the public parameters like $\{m_1, m_2\}$, employ random oracle Reveal1 for the implementation of algorithm $EXP^{HASH}_{\mathbf{EAMSARC}}$. The success probability for $EXP^{HASH}_{\mathbf{EAMSARC}}$ is Suc1 = Pr.2[$EXP^{HASH}_{\mathbf{EAMSARC}}$ = 1] − 1, while Pr[E] suggests an event E probability. The advantage function for this experiment becomes as $Adv^{HASH}_{\mathbf{EAMSARC}} t_1, q_{\mathrm{R1}}) = max_{\not{A}} [Suc1^{HASH}_{\mathbf{EAMSARC}}]$, with the execution time t $_1$ and random Reveal query q_{R1} maximized on $\not{A}$. We call our proposed technique as provably secure against an attacker $\not{A}$ for deriving the valid session key $SKij$ if $Adv^{HASH}_{\mathbf{EAMSARC}}(t_1, q_{\mathrm{R1}}) \leq \varepsilon$ for any sufficiently small $\varepsilon > 0$. According to this experiment, if an attacker $\not{A}$ has the ability of inverting a one-way hash function h(.), and solving the hard problem CMDLP, then it can easily extract the legal user ID$_{\mathrm{U}}$ and shared session key $SKij$ between Ui and Sj, and wins the game. However, according to definition (1), this is computationally infeasible to invert hash function, as $Adv^{HASH}_{\mathbf{EAMSARC}}(t_1) \leq \quad \varepsilon$ for any sufficiently small $\varepsilon > 0$.

5.3 Performance efficiency analysis

As we have described earlier, the Chebyshev polynomial computation is almost three times efficient than elliptic curve cryptography (ECC) and even more efficient than RSA-based cryptography [1–9]. The Chebyshev polynomial computation provides fast computation with less key size, and requires less bandwidth and memory consumption [38]. Hence, in our scheme, there are no modular exponentiations, elliptic curve-based scalar multiplications. This section deals with the comparison for the cost of proposed model with Zhu et al., Shen et al., Tsai et al., and Jiang et al. protocols, which have also used Chebyshev polynomial map in their protocols except Shen and jiang et al. schemes, as described below. The following Table 2 presents the vulnerability and drawback analysis for five schemes [9–12].

A few notations used in the comparison are as follows.

T_{XOR} The time executing the XOR operation.

T_{H} The time taken for the hash operation;

T_{SYM} The time for symmetric key cryptography;

T_{ECM} The time for elliptic curve-based scalar multiplication;

T_{CCM} The time for executing the Chebyshev Chaotic polynomial mapping $T_n(x) modp$ following the algorithm [31].

In this section, we compare the schemes' costs by estimating the running times for various cryptographic operations (based on the PBC library, Ubuntu 12.04.1 32 bit operating system, with 2.4 GHz CPU, and 2.0 GB RAM). Accordingly, the computational time of hash-based operation, symmetric encryption or decryption, elliptic curve scalar multiplication, and Chebyshev polynomial operations are 0.0006, 0.0088, 0.063073, and 0.02104s, respectively. The XOR operation cost is negligible as compared to other cryptographic operations, hence, could be ignored. The following

Table 2 Comparison for Shen et al., Tsai et al., Jiang et al., Zhu et al. and the proposed protocol

	Shen et al. [9]	Tsai et al. [10]	Jiang et al. [11]	Zhu et al. [12]	Proposed protocol
Anonymity	No	Yes	No	Yes	Yes
Mutual authentication	Yes	Yes	Yes	Yes	Yes
Resist insider attack	Yes	Yes	Yes	Yes	Yes
Mutual authentication without RC	No	No	No	No	Yes
Resist secret/password guessing attack	Yes	Yes	Yes	Yes	Yes
Resist impersonation attack	Yes	Yes	Yes	Yes	Yes
Resist replay attack	Yes	Yes	Yes	Yes	Yes
Session key agreement	Yes	Yes	Yes	Yes	Yes
Perfect forward secrecy	Yes	Yes	Yes	Yes	Yes
Known key secrecy	Yes	Yes	Yes	Yes	Yes
Resistant to location traceability	Yes	Yes	No	Yes	Yes
Resist stolen verifier attack	Yes	Yes	No	Yes	Yes

Table 3 Cost comparison for Shen et al., Tsai et al., Jiang et al., Zhu et al. and proposed scheme

	Shen et al. protocol [9]	Tsai et al. protocol [10]	Jiang et al. protocol [11]	Zhu et al. protocol [12]	Proposed protocol
Computation cost for authentication messages	$16T_H + 6T_{ECM} \approx 0.388038$	$21T_H + 2T_{CCM} \approx 0.05468$	$21T_H + 4T_{ECM} \approx 0.264892$	$9T_H + 4T_{SYM} + 2T_{CCM} \approx 0.08268$	$10T_H + 6T_{CCM} \approx 0.13224$
Communication latency in rounds (R)	5R	5R	5R	3R	2R

Table 1 shows the result of cost estimation for these five protocols, i.e., Zhu et al. [12], Shen et al. [9], Tsai et al. [10] and Jiang et al. [11] and proposed scheme.

Hence, in the light of above performance analysis, for authentication phase, the proposed scheme bears less computation cost as compared to Shen and Jiang et al. schemes, while it incurs more cost than Zhu et al. and Tsai et al. schemes for the same phase as shown in Table 3. The less cost of Zhu et al. is also attributed to one-way authentication, as the two-way authentication bears an additional cost. The Tsai scheme, though less costly, is prone to server-spoofing attack as mentioned in previous section. The proposed scheme incurs an average cost regarding computational cost of authentication. Nonetheless, the proposed scheme is far efficient in terms of communication latency or the number of round-trips, and thwarts almost all of the known attacks in that average cost. With the ever-increasing power of computation, the focus needs to be put on minimizing the communication delay or latency for which the physical medium or infrastructure has been responsible. Hence, in proposed protocol, we have eliminated the RC involvement for the authentication phase that has helped in optimizing the communication latency. The role of RC is limited to user and server registration phase in our protocol, and not afterwards (mutual authentication phase). As we know that with the increase in the number of users in a system, a central entity becomes a bottleneck for the increasing load and service requests. Besides, the proposed protocol has been proven resistant to attacks as shown in Table 2 and is also proved in the random oracle model as elaborated above. Overall, we can say that the proposed scheme is not only an efficient scheme for less computational cost, but also provides additional security like anonymity.

6 Conclusion

A multi-server authentication scheme provides the multiplicity of services of different servers to subscribers by one-time registration of an RC. The current study reviews few recent multi-server authentication techniques, i.e., Zhu et al., Shen et al., Tsai et al., and Jiang et al., schemes. These schemes are not only vulnerable to attacks, but also suffer communication latency due to the RC involvement in each mutual authentication. The RC involvement in each authentication phase might prove to be a bottleneck for a system that requires scalability. Hence, in such a system, the number of users cannot be added beyond a certain level, without making the system inefficient and overloaded. The proposed scheme employs Chebyshev chaotic map to optimize the scheme as compared to costly Shen et al. and Jiang et al. schemes based on elliptic cryptography. The proposed scheme is not only robust against attacks as identified in earlier schemes, but also efficient in terms of communication-latency as proved in above sections. Moreover, the findings in proposed model are backed by formal security analysis and performance evaluation.

References

1. Lamport L (1981) Password authentication with insecure communication. Commun ACM 24(11):770–772
2. Lee NY, Chiu YC (2005) Improved remote authentication scheme with smart card. Comput Stand Interfaces 27(2):177–180
3. Sun HM (2000) An efficient remote use authentication scheme using smart cards. IEEE Trans Consum Electron 46(4):958–961
4. Lin CH, Lai YY (2004) A flexible biometrics remote user authentication scheme. Comput Stand Interfaces 27(1):19–23
5. Khan MK, Zhang J (2007) Improving the security of a flexible biometrics remote user authentication scheme. Comput Stand Interfaces 29(1):82–85
6. Li LH, Lin IC, Hwang MS (2001) A remote password authentication scheme for multi-server architecture using neural networks. IEEE Trans Neural Netw 12(6):1498–1504

7. Lin IC, Hwang MS, Li LH (2003) A new remote user authentication scheme for multi-server architecture. Future Gener Comput Syst 19(1):13–22
8. Tsai JL (2008) Efficient multi-server authentication scheme based on one-way hash function without verification table. Comput Secur 27(3–4):115–121
9. Shen H et al (2015) New biometrics-based authentication scheme for multi-server environment in critical systems. J Ambient Intell Humaniz Comput 6(6):825–834. doi:10.1007/s12652-015-0305-8
10. Tsai Jia L, Nai WL (2014) A chaotic map based anonymous multi-server authenticated key agreement protocol using smart card. Int J Commun Syst 28(13). doi:10.1002/dac.2829
11. Jiang P et al (2015) An anonymous and efficient remote biometrics user authentication scheme in a multi server environment. Front Comput Sci 9(1):142–156. doi:10.1007/s11704-014-3125-7
12. Zhu H (2015) A provable one-way authentication key agreement scheme with user anonymity for multi-server environment. KSII Trans Internet Inf Syst 9(2):811–829. doi:10.3837/tiis.2015.02.19
13. Ravi SP, Jaidhar CD, Shashikala T (2013) Robust smart card authentication scheme for multiserver architecture. Wirel Pers Commun 72:729–745. doi:10.1007/s11277-013-1039-6
14. Zhang L (2008) Cryptanalysis of the public key encryption based on multiple chaotic systems. Chaos Solitons Fractals 37(3):669–674
15. Yoon E-J, Yoo K-Y (2013) Robust biometrics-based multi-server authentication with key agreement scheme for smart cards on elliptic curve cryptosystem. J Supercomput 63:235–255
16. Liao YP, Wang SS (2009) A secure dynamic ID based remote user authentication scheme for multi-server environment. Comput Stand Interfaces 31(1):24–29
17. Wen FT, Li XL (2011) An improved dynamic ID-based remote user authentication with key agreement scheme. Comput Electr Eng 38(2):381–387
18. Hsiang HC, Shih WK (2009) Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment. Comput Stand Interfaces 31(6):1118–1123
19. Lee CC, Lin TH, Chang RX (2011) A secure dynamic ID based remote user authentication scheme for multi-server environment using smart cards. Expert Syst Appl 38(11):13863–13870
20. Guo DL, Wen FT (2014) Analysis and improvement of a robust smart card based-authentication scheme for multi-server architecture. Wirel Pers Commun 78(1):475–490
21. Wen FT, Susilo W, Yang GM (2013) A robust smart card based anonymous user authentication protocol for wireless communications. Secur Commun Netw 7(6):987–993
22. Sood SK, Sarje AK, Singh K (2011) A secure dynamic identity based authentication protocol for multi-server architecture. J Netw Comput Appl 34(2):609–618
23. Li X, Xiong YP, Ma J, Wang WD (2012) An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards. J Netw Comput Appl 35(2):763–769
24. Xue KP, Hong PL, Ma CS (2014) A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture. J Comput Syst Sci 80(1):195–206
25. Khan MK, Zhang J (2007) Improving the security of a flexible biometrics remote user authentication scheme. Comput Stand Interfaces 29(1):82–85
26. Kim HS, Lee JK, Yoo KY (2003) ID-based password authentication scheme using smart cards and fingerprints. ACM SIGOPS Oper Syst Rev 37(4):32–41
27. Lee JK, Ryu SR, Yoo KY (2002) Fingerprint-based remote user authentication scheme using smart cards. Electron Lett 38(12):554–555
28. 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
29. Lin H, Fengtong W, Chunxia D (2015) An improved anonymous multi-server authenticated key agreement scheme using smart cards and biometrics. Wirel Pers Commun 84:2351–2362
30. Masuda N, Aihara K (2002) Cryptosystems with discretized chaotic maps. IEEE Trans Circuits Syst 49:28–40
31. Kocarev L, Lian S (2011) Chaos-based cryptography: theory, algorithms and applications. Springer, Berlin
32. Koblitz N (1987) Elliptic curve cryptosystems. Math Comp 48:203–209
33. Bellare M (1999) Practice-oriented provable security. In: Lectures on data security. Lecture notes in computer science, vol 1561. Springer, Berlin, pp 1–15
34. Behnia S, Akhshani A, Ahadpour S, Mahmodi H, Akhavan A (2007) A fast chaotic encryption scheme based on piecewise nonlinear chaotic maps. Phys Lett A 366:391–396
35. Baptista MS (1998) Cryptography with chaos. Phys Lett A 240:50–54

36. Xiao D, Liao X, Wong K (2005) An efficient entire chaos-based scheme for deniable authentication. Chaos Solitons Fractals 23:1327–1331
37. Khan M, Shah T, Mahmood H, Gondal M (2013) An efficient method for the construction of block cipher with multi-chaotic systems. Nonlinear Dyn 71:489–492
38. Han S (2008) Security of a key agreement protocol based on chaotic maps. Chaos Solitons Fractals 38:764–768
39. Xiang T, Wong K, Liao X (2009) On the security of a novel key agreement protocol based on chaotic maps. Chaos Solitons Fractals 40:672–675
40. Guo X, Zhang J (2010) Secure group key agreement protocol based on chaotic Hash. Inf Sci 180:4069–4074
41. Yoon E, Jeon I (2011) An efficient and secure Diffie-Hellman key agreement protocol based on Chebyshev chaotic map. Commun Nonlinear Sci Numer Simul 16:2383–2389
42. Lai H, Xiao J, Li L, Yang Y (2012) Applying semigroup property of enhanced Chebyshev polynomials to anonymous authentication protocol. Math Probl Eng. doi:10.1155/2012/454823
43. Stolbbnunov A (2009) Reductionist security arguments for public-key cryptographic schemes based on group action. In: The Norwegian information security conference (NISK), pp 97–109
44. Wang B, Ma M (2012) A smart card based efficient and secured multi-server authentication scheme. Wirel Pers Commun. doi:10.1007/s11277-011-0456-7
45. Xiao D, Shih F, Liao X (2010) A chaos-based hash function with both modification detection and localization capabilities. Commun Nonlinear Sci Numer Simul 15:2254–2261
46. Hsieh W, Leu J (2012) Anonymous authentication protocol based on elliptic curve Diffie-Hellman for wireless access networks. Wirel Commun Mobile Comput. doi:10.1002/wcm.2252