An efficient and anonymous Chaotic Map based authenticated key agreement for multi-server architecture

Azeem Irshad[1], Hafiz Farooq Ahmad[2], Bander A. Alzahrani[3], Muhammad Sher[1] , Shehzad Ashraf Chaudhry[1]

[1] Department of Computer Science & Software Engineering, International Islamic University, Islamabad, Pakistan
[e-mail: irshadazeem2@gmail.com, shahzad@iiu.edu.pk, m.sher@iiu.edu.pk]

[2] College of Computer Sciences and Information Technology (CCSIT), King Faisal University, Alahssa, Saudi Arabia
[e-mail: hfahmad@kfu.edu.sa]

[3] Faculty of Computing & Information Technology, King Abdulaziz University, Saudi Arabia
[e-mail: baalzahrani@kau.edu.sa]
*Corresponding author: Azeem Irshad

Received July 24, 2016; revised October 20, 2016; accepted November 15, 2016; published December 31, 2016

Abstract

Multi-server authentication enables the subscribers to enjoy an assortment of services from various service providers based on a single registration from any registration centre. Previously, a subscriber had to register from each service provider individually to avail respective services relying on single server authentication. In the past, a number of multi-server authentication techniques can be witnessed that employed lightweight and even computationally intensive cryptographic operations. In line with this, Zhu has presented a chaotic map based multi-server authentication scheme recently, which is not only vulnerable to denial-of-service attack, stolen-verifier attack, but also lacks anonymity. This research aims at improving the Zhu’s protocol in terms of cost and efficiency. Moreover, the comparative study is presented for the performance of improved model against the existing scheme, and the security of proposed model is formally proved using BAN Logic.

Keywords: Multi-server authentication, cryptography, anonymity, Chebyshev chaotic map, Authentication key agreement

http://dx.doi.org/10.3837/tiis.2016.12.023 **ISSN : 1976-7277**

1. Introduction

The chaos cryptography has been growing popular for its cost efficiency, unpredictability, and the sensitivity to initial parameters. A few chaotic map-based protocols for key agreement authentication have been proposed in the last few years [1-5]. These contributions can be classified on the basis of a number of participants in the protocol i.e., Two Party Authentication Key agreement (2PAKE) [7-18], Three Party Authentication Key agreement (3PAKE) [20-25] and N-Party Authentication Key agreement (NPAKE). Besides, the schemes can also be categorized with respect to password, smart card, anonymity preserving, and other features based security protocols. Recently 3PAKE schemes based on modular exponentiation and scalar multiplications have been researched extensively [22, 24].
Multi-server authentication (MSA) protocols facilitate the user to register at a registration centre and relax the requirement for multiple registrations at many service providers individually. [26-30, 49-50]. We may categorize those techniques into three sections as illustrated below.

Creative phase: The creative phase covers an era of early contribution of Li et al. [33]. Afterwards, Lin et al. [40] demonstrated that the Li et al. protocol is inefficient due to large amount of time required for training neural based networks. Lin et al. then put forward a technique based on ElGamal digital signature [48].

Development phase: Since the research, being an ongoing activity, takes its course out of many evolutions and developments. In this connection, Tsai [35] presented a one-way hash function-based MSA technique without maintaining a stored verifiers table. Even though, it was a low cost operations-based scheme for a distributed network framework, it was found to be vulnerable to privileged insider attack, server spoofing attack, and also the compromise of perfect forward secrecy.

Diversification phase: In the current era, the emphasis of authentication-based research including multiserver techniques, has been switched to functionality based techniques. As a result, we can witness identity-based MSA protocols, dynamic identity based multiserver protocols, bilinear pairing or ECC (Elliptic Curve Cryptography) related MSA techniques, along with other protocols as well [36-38, 43-44].

Following the diversification phase, Zhu [42], recently presented a chebyshev chaotic map based multi-server authentication protocol. This scheme fails to manage user anonymity and is also found to be vulnerable to trace attack. Besides, the scheme is also found vulnerable to Denial of Service (DoS) attack and stolen-verifier attack. The scheme has the potential of improvements regarding overheads, for the same assumptions. In this work, we propose an improved scheme based on Zhu, that not only covers all of the limitations as mentioned above, but also improves the scheme in terms of efficiency.

The paper is structured as follows; Section 2 illustrates preliminaries related to the current work. Section 3 discusses working and weaknesses of Zhu’s scheme. Section 4 presents our proposed model. Section 5 exhibits security and performance evaluation analysis. Section 6 concludes the findings of the paper.

2. Preliminaries

This section describes the preliminaries regarding multi-server to server architecture, Chebyshev Chaotic Map, one-way hash function, and symmetric encryption as below.

2.1 Multi-Servers to Server architecture (MS-S)

In MS-S based environment, there is no need for a subscriber to get registration at servers individually to avail their services. While, in a single authentication environment, a user gets registered to each and every server, and has to memorize all of the respective passwords and parameters for its verification to those servers. In order to ease the process for users, the MS-S architecture has been proposed [33-38]. Basically, this scheme bounds the users to register once, be it from any of the server or registration centre, then the user is free to get authenticated from this server/RC through any of the servers of that the user intends to receive services. In MS-S, there is no fixed registration centre or a server, which distinguishes the proposed scheme with typical MSA-based schemes. In MS-S, whenever, a user wants to avail services of any novel server S_x (part of multi-server system), then S_x verifies authenticity of user from S_y (S_y, server where the user got registered). We acknowledge the concept for Zhu's scheme [42] that this architecture overcomes the single-point of failure.

2.2 Chebyshev Chaotic Maps

Some of the salient properties of Chebyshev polynomial and Chebyshev chaotic maps are stated as follows:
We suppose n as an integer, and assume a variable α having the interval [-1, 1]. Now, we can describe the Chebyshev chaotic based polynomial $T_n(\alpha)$: $[-1, 1] \rightarrow [-1, 1]$ as $T_n(\alpha)=\cos(n \arccos(\alpha))$. A recurrent relation can be used to describe Chebyshev polynomial map $\mathbf{T_n}: R \rightarrow R$ of degree n, in the following manner:

$$T_n(x) = 2\,\alpha\, T_{n-1}(\alpha) - T_{n-2}(\alpha), \quad (1)$$

Given $n \geq 2$, $T_0(\alpha)=1$, and $T_1(\alpha)=\alpha$

The first few Chebyshev polynomials are listed below:

$$T_2(\alpha)=2\,\alpha^2-1 \quad (2)$$

$$T_3(\alpha)=4\,\alpha^3-3\,\alpha \quad (3)$$

$$T_4(x)=8\,\alpha^4-8\,\alpha^2+1 \quad (4)$$

Chebyshev polynomial has two features:

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

The semigroup feature [24]: The semi-group feature of Chebyshev polynomial can be defined on an interval $[-\infty, +\infty]$ as defined below:

$$T_n(\alpha) = (2\,\alpha\, T_{n-1}(\alpha) - T_{n-2}(\alpha)) \bmod p \quad (5)$$

Given that $n>=2$, $\alpha \in [-\infty, +\infty]$, and p be a large range prime number. Besides,

$$T_a(T_b(\alpha)) \equiv T_{ab}(\alpha) \equiv T_b(T_a(\alpha)) \bmod p \quad (6)$$

Chaotic maps-based discrete logarithm problem (CMDLP): It will be a hard problem or requires non-polynomial amount of time to guess s, such that $T_s(a)=b$.
Chaotic maps-based Diffie-Hellman problem (CMDHP): Similarly, it is a hard problem to compute $T_{ab}(\alpha)$, given $T_a(\alpha)$, $T_b(\alpha)$ or α parameters.

2.3 One-way hash function

A proven secure one-way hash operation h: x → y comprises four features:

1. The hash function h inputs a message of arbitrary string of length and outputs a fixed-length message digest.
2. Given h(x)=y, it is a hard problem to calculate $h^{-1}(y)=x$;
3. Given x, it is hard to find x', such that $x' \neq x$, but h(x') =h(x);
4. Besides, it is computationally a hard problem to locate any pair x, x' such that $x' \neq x$, but h(x') =h(x).

2.4 Symmetric Encryption

A symmetric encryption scheme E_k(K *gen*, *E*, *D*) comprises three sub-algorithms as follows:

1. Randomized key generation (K *gen*): This algorithm returns a key k, generated out of the key space *keys* (E_k) randomly.
2. Encryption *E*: This *E* algorithm takes the key k from *keys* (E_k), inputs a plaintext $P \in \{0, 1\}^*$ and generates a ciphertext $C \in \{0, 1\}^*$.
3. Decryption *D*: Likewise, *D* algorithm takes a ciphertext $C \in \{0, 1\}^*$ as input, and decrypts, produces plaintext $P \in \{0, 1\}^*$, using key k from *keys* (E_k).

3. WORKING OF ZHU'S SCHEME AND LIMITATIONS.

This section descibres working of Zhu's scheme and weaknesses of the same protocol. The Zhu [42] protocol is composed of three phases: multiple servers to server setup phase, user registration phase, login and authentication phase. The scheme makes a use of a few symbols as mentioned in **Table 1**.

Table 1. Notations description

Notations	Description
U_i, ID_U, PW_U	User, U_i's identity, U_i's password
S_y	Server S_y , U_i is assumed to be already registered with S_y
S_x	Server S_x , U_i gets mutually authenticated with S_x through S_y
K_y	Secret key of S_y
K_{xy}	Shared key between S_x and S_y
$(x, T_{S_r}(x))$	Temporary public key for server S_x, based on Chebyshev chaotic map
$(x, T_{K_y}(x))$	Permanent public key for server S_y, based on Chebyshev chaotic map
$H(.)$	Chaotic map based hash function
$E_k()/D_k()$	Symmetric encryption/decryption
a, S_r, S_R	Random session variables
SK	Shared session key between U_i and S_x

3.1 Working of Zhu's scheme

This sub-section describes the design and working of Zhu's scheme, as follows.

3.1.1 Multiple servers to server registration phase:

The public and secret keys for servers S_i (1≤ i ≤n) in a system, are defined as (x, T_{ki} (x)) (1 ≤ i ≤n), and k_i (1 ≤ i ≤n). Here, any of the two servers share high entropy secret key as x_{ij} (1 ≤ i ≤n & 1 ≤ j ≤n). This multiple server scheme has its advantages for its flexibility and expansion. Earlier before user registration all of the service providers should have their public keys verified by the authorities.

3.1.2 User Registration Phase

In this phase, a user gets pre-registered with any of its associated authorized server $\boldsymbol{S_y}$ employing a secure channel, using Chebyshev chaotic map-based multi-server to server architecture [42]. After a successful registration the user establishes a shared password PW_U with S_y, which is stored by S_y in its repository $\mathcal{R}$ against ID_U. Next, the former may establish a secure session key with any new server S_x (already authorized from and registered with S_y), by employing a login and authentication phase as described in the next section.

3.1.3 Login and Authentication phase

1. In login and authentication phase, a new random number a is generated. Then the user constructs the related public key as $T_a(x)$, and a shared key $K_{US_y}=T_aT_{K_y}(x)$, using Chebyshev chaotic map, and further calculates $H_U=H(ID_U \,||\, ID_{S_x} \,||\, T_a(x) \,||\, PW_U)$, $C_1=E_{kUS_y}(ID_U \,||\, ID_{S_x} \,||\, ID_{S_y} \,||\, H_U)$. The user sends message $m_1=\{T_a(x), ID_U, ID_{S_y}, C_1\}$ to S_x, finally as shown in **Fig. 1**.
2. Next, S_x receives $m_1=\{T_a(x), ID_U, ID_{S_y}, C_1\}$ and generates a random integer S_r, and compute public key $T_{S_r}(x)$, and shared key $K_{S_xS_y}=T_{S_r}T_{K_y}(x)$ using the S_y's public key $T_{K_y}(x)$. Then computes $H_{S_x}=H(ID_U \,||\, ID_{S_x} \,||\, T_{S_r}(x) \,||\, K_{xy})$ using hash, and $C_2=E_{kS_xS_y}(ID_U \,||\, ID_{S_x} \,||\, H_{S_x})$ by encrypting using the computed shared key $K_{S_xS_y}=T_{S_r}T_{K_y}(x)$. Finally, S_x sends $m_2=\{m_1, T_{S_r}(x), ID_{S_x}, C_2\}$ to S_y for user's verification.
3. After receiving $m_2=\{m_1, T_{S_r}(x), ID_{S_x}, C_2\}$ from S_x, S_y computes $K_{S_yU}=T_{K_y}T_a(x)$, and decrypts $D_{kS_yU}(C_1)= (ID_U \,||\, ID_{S_x} \,||\, ID_{S_y} \,||\, H_U)$. Then computes $K_{S_yS_x}=T_{K_y}T_{S_r}(x)$, and decrypts $D_{kS_yS_x}(C_2)= (ID_U \,||\, ID_{S_x} \,||\, H_{S_x})$. Now it computes $H'_U=H(ID_U \,||\, ID_{S_x} \,||\, T_a(x) \,||\, PW_U))$, $H'_{S_x}=H(ID_U \,||\, ID_{S_x} \,||\, T_{S_r}(x) \,||\, K_{xy}))$ and verifies the equality for $H'_U \;?=\; H_U$ and $H'_{S_x} \;?=\; H_{S_x}$. If found true, then further computes $H_{S_yS_x}=H(T_a(x) \,||\, K_{xy})$, $C_3=E_{kS_yS_x}(ID_U \,||\, ID_{S_x} \,||\, H_{S_yS_x})$, $H_{S_yU} = H(T_{S_r}(x) \,||\, PW_U)$, and $C_4=E_{kS_yU}(ID_U \,||\, ID_{S_y} \,||\, H_{S_yU})$. Finally, S_y sends the message $m_3 = \{ ID_{S_y}, C_3, C_4\}$ to S_x .

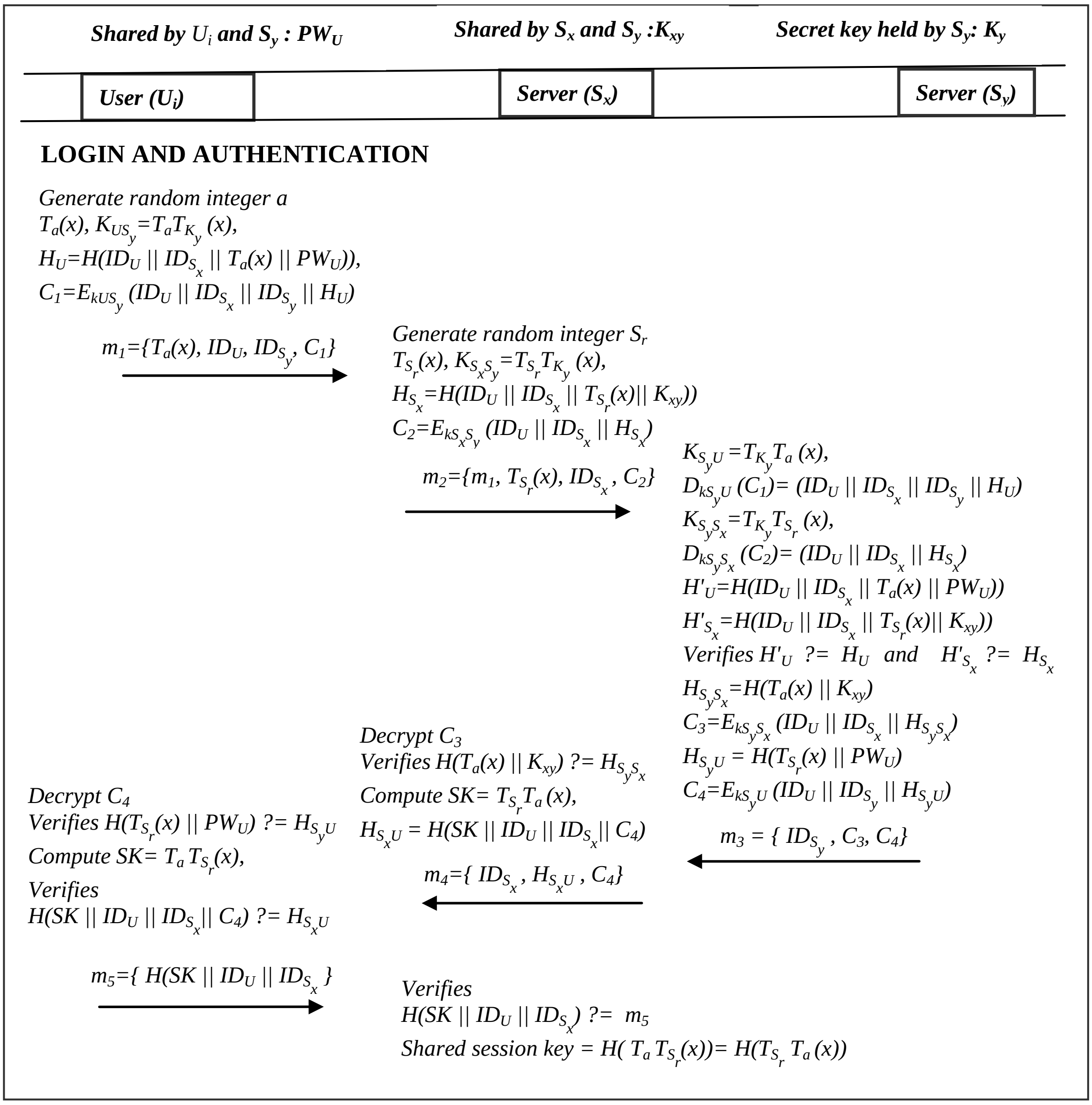

Shared by U_i and S_y : PW_U
Shared by S_x and S_y :K_{xy}
Secret key held by S_y: K_y
User (U_i)
Server (S_x)
Server (S_y)
LOGIN AND AUTHENTICATION
Generate random integer a
$T_a(x)$, $K_{US_y}=T_aT_{K_y}(x)$,
$H_U=H(ID_U \| ID_{S_x} \| T_a(x) \| PW_U))$,
$C_1=E_{kUS_y}(ID_U \| ID_{S_x} \| ID_{S_y} \| H_U)$
$m_1=\{T_a(x), ID_U, ID_{S_y}, C_1\}$
Generate random integer S_r
$T_{S_r}(x)$, $K_{S_xS_y}=T_{S_r}T_{K_y}(x)$,
$H_{S_x}=H(ID_U \| ID_{S_x} \| T_{S_r}(x)\| K_{xy}))$
$C_2=E_{kS_xS_y}(ID_U \| ID_{S_x} \| H_{S_x})$
$m_2=\{m_1, T_{S_r}(x), ID_{S_x}, C_2\}$
$K_{S_yU}=T_{K_y}T_a(x)$,
$D_{kS_yU}(C_1)=(ID_U \| ID_{S_x} \| ID_{S_y} \| H_U)$
$K_{S_yS_x}=T_{K_y}T_{S_r}(x)$,
$D_{kS_yS_x}(C_2)=(ID_U \| ID_{S_x} \| H_{S_x})$
$H'_U=H(ID_U \| ID_{S_x} \| T_a(x) \| PW_U))$
$H'_{S_x}=H(ID_U \| ID_{S_x} \| T_{S_r}(x)\| K_{xy}))$
Verifies H'_U ?= H_U and H'_{S_x} ?= H_{S_x}
$H_{S_yS_x}=H(T_a(x) \| K_{xy})$
$C_3=E_{kS_yS_x}(ID_U \| ID_{S_x} \| H_{S_yS_x})$
$H_{S_yU}=H(T_{S_r}(x) \| PW_U)$
$C_4=E_{kS_yU}(ID_U \| ID_{S_y} \| H_{S_yU})$
$m_3=\{ID_{S_y}, C_3, C_4\}$
Decrypt C_3
Verifies $H(T_a(x) \| K_{xy})$?= $H_{S_yS_x}$
Compute SK= $T_{S_r}T_a(x)$,
$H_{S_xU}=H(SK \| ID_U \| ID_{S_x}\| C_4)$
$m_4=\{ID_{S_x}, H_{S_xU}, C_4\}$
Decrypt C_4
Verifies $H(T_{S_r}(x) \| PW_U)$?= H_{S_yU}
Compute SK= $T_aT_{S_r}(x)$,
Verifies
$H(SK \| ID_U \| ID_{S_x}\| C_4)$?= H_{S_xU}
$m_5=\{H(SK \| ID_U \| ID_{S_x}\}$
Verifies
$H(SK \| ID_U \| ID_{S_x})$?= m_5
Shared session key = $H(T_aT_{S_r}(x))=H(T_{S_r}T_a(x))$

Fig. 1. Login and Authentication for Zhu's protocol

4. S_x, on the receipt of message $m_3=\{ID_{S_y}, C_3, C_4\}$, decrypts C_3 *as* $(ID_U \| ID_{S_x} \| H_{S_yS_x})=D_{kS_xS_y}(C_3)$. Then, it verifies $H(T_a(x) \| K_{xy})$?= $H_{S_yS_x}$. If true, then computes SK= $T_{S_r}T_a(x)$, $H_{S_xU}=H(SK \| ID_U \| ID_{S_x}\| C_4)$ and then generates a message $m_4=\{ID_{S_x}, H_{S_xU}, C_4\}$ to send it to user.
5. The user receives message $m_4=\{ID_{S_x}, H_{S_xU}, C_4\}$ and decrypts C_4 as $(ID_U \| ID_{S_y} \| H_{S_yU})=D_{kS_yU}(C_4)$. Next, it computes $H(T_{S_r}(x) \| PW_U)$ and verifies $H(T_{S_r}(x) \| PW_U)$?= H_{S_yU}. If found true, then computes SK= $T_a T_{S_r}(x)$, and again verifies the equality $H(SK\|ID_U\|ID_{S_x}\|C_4)$?= H_{S_xU}, otherwise, aborts the session. Finally, it generates $m_5=\{H(SK \| ID_U \| ID_{S_x}\}$ and sends to S_x.

6. Next, S_x verifies the message m$_5$ by confirming the equality check $H(SK \,||\, ID_U \,||\, ID_{S_x})$?= m_5. If true, then it develops the shared session key as $H(T_{S_r}\, T_a\,(x))$.

3.2 Inefficiencies in Zhu's scheme

The Zhu's scheme presented a novel multi-server to sever authentication technique to eliminate the centralized registration center; however, has the following limitations.

3.2.1Trace attack and lacks anonymity

The Zhu's scheme fails to comply anonymity on the part of a user U_i, since, the user identity ID_U is openly available in exchanged messages on insecure channel. At the same time, traceability attack may be launched comfortably by an adversary, which might find the same ID_U for various sessions established, leading to the *trace attack*.

3.2.2Stolen-verifier attack

The Zhu's scheme might be vulnerable to stolen-verifier attack. Since, in Zhu, the servers maintain password table or repository $\mathcal{R}$. These passwords are located against the identity ID_U of users. The author does not provide any mechanism to protect the password table from adversary, in case; the stored verifiers' repository $\mathcal{R}$ is exposed. Since, if $\mathcal{R}$'s contents are exposed, an adversary may launch a successful user impersonation attack against server S_y.

3.2.3Denial-of-Service attack (DoS)

This attack makes the server engage in delay-intensive operations like searching the database that might render the server unavailable for other productive tasks. The Zhu's scheme is vulnerable to DoS attack, since the user sends its ID_U towards server, which (server) searches its database to locate the valid password PW_U. This attempt might be succeeded, if an adversary replays the login request, or otherwise end in a failure due to the fake ID_U, in the worst case. In both ways, the server looks in its database and after finding that, the server computes and decides the validity for the user. An adversary might burden the server easily by sending too many fake requests simultaneously. Hence, an adversary may render the server out of computing power for handling those fake requests.

4. Proposed Model

In proposed scheme, we have introduced an improved version of Zhu's scheme that not only provides the same level of security with anonymity and untracebility, at a lesser cost, but also protects the user from DoS and stolen verifier attacks. The proposed model comprises four phases, i.e. the registration phase, login and authentication phase, password update phase, and shared key update phase.

4.1 User registration phase

In this phase, a user gets registered with S_y by adopting the following steps.

1. The user U_i sends identity and password (ID_U, PW_U) to server S_y using a secure channel.
2. The server S_y generates a pseudonym identity $PIDi$ against U_i, and computes $h(K_y||PIDi) \oplus (ID_U|| PW_U)$, and stores in repository against $PIDi$.
3. Next, it sends $\{PIDi, ID_{S_y}, H(), (x, T_{K_y}(x))\}$ to the user. The U_i receives and stores $PIDi$ safely. In this manner, the user establishes a shared password PW_U with S_y, which enables the user to establish a secure session key with a unknown server S_x (already authorized with S_y), by initiating a login and authentication phase as described below.

4.2 Login and authentication phase

1. In this phase, the user defines a long entropy random number a as its secret. Then U_i constructs a public key as $T_a(x)$, using Chebyshev chaotic map. Then, it further computes $H_U = H(ID_U || ID_{S_x} || T_a(x) || T_{K_y}(x) || T_aT_{K_y}(x) || PW_U)$. Finally, the user sends message $m_1 = \{H_U, T_a(x), ID_{S_x}, ID_{S_y}, PIDi\}$ to S_x.
2. Next, S_x receives m_1 and generates a random number S_r, and compute public key $T_{S_r}(x)$. Further, it computes $H_{S_x} = H(T_a(x), ID_{S_x} || T_{S_r}(x) || K_{xy})$ using hash, and sends $m_2 = \{m_1, T_{S_r}(x), ID_{S_x}, H_{S_x}\}$ to S_y for user's verification.
3. After receiving $m_2 = \{m_1, T_{S_r}(x), ID_{S_x}, H_{S_x}\}$ from S_x, S_y computes $(q||ID_U||PW_U) = PID' \oplus K$. Next, it computes $T_{K_y}T_a(x)$, $H'_U = H(ID_U || ID_{S_x} || T_a(x) || T_{K_y}(x) || T_aT_{K_y}(x) || PW_U)$, $H'_{S_x} = H(T_a(x), ID_{S_x} || T_{S_r}(x) || K_{xy})$, and verifies $H'_U ?= H_U$ and $H'_{S_x} ?= H_{S_x}$. If these equations hold true, then generates random integer q, $PID' = ((q||ID_U||PW_U) \oplus K_y)$**,** and computes $Di = h(T_aT_{K_y}(x)) \oplus PIDi'$, $Dx = h(T_a(x) || K_{xy}) \oplus ID_U$, $H_{S_yS_x} = H(Dx||H_{S_yU} ||PIDi || T_a(x) || K_{xy})$, $H_{S_yU} = H(ID_{S_x} || T_a(x) || T_{S_r}(x) ||Di|| T_aT_{K_y}(x)|| PW_U)$. Finally, it sends the message $m_3 = \{ ID_{S_y} T_a(x), Di, Dx, H_{S_yS_x}, H_{S_yU} \}$ to S_x.
4. After receiving $m_3 = \{ID_{S_y} T_a(x), Di, Dx, H_{S_yS_x}, H_{S_yU}\}$ from S_y, S_x verifies the equation $H_{S_yS_x} ?= H(Dx|| H_{S_yU} || PIDi || T_a(x) || K_{xy})$. If it does not hold true, it aborts the session. Otherwise, computes $ID_U = h(T_a(x) || K_{xy}) \oplus Dx$, $SK = T_{S_r}T_a(x)$, $H_{S_xU} = H(SK || T_{S_r}(x) ||ID_U || ID_{S_x}|| H_{S_yU})$. It sends the message $m_4 = \{ ID_{S_x}, Di, H_{S_yU}, H_{S_xU}, T_{S_r}(x)\}$ to user finally.
5. The user verifies the equality $H_{S_yU} ?= H(ID_{S_x} || T_a(x) || T_{S_r}(x) || Di || T_aT_{K_y}(x) || PW_U)$, and computes $SK = T_a T_{S_r}(x)$, and verifies $H_{S_xU} ?= H(SK || T_{S_r}(x) || ID_U || ID_{S_x}|| H_{S_yU})$. If it holds true, it validates the S_x as a legal server, establishes a shared session key as $H(T_aT_{S_r}(x))$. Then, it computes $PIDi' = h(T_aT_{K_y}(x)) \oplus Di$, and replaces $PIDi$ with $PIDi'$ in smart card. Finally, it generates a message $m_5 = \{ H(SK || ID_U || ID_{S_x} \}$ for sending towards S_x to conclude the session establishment.

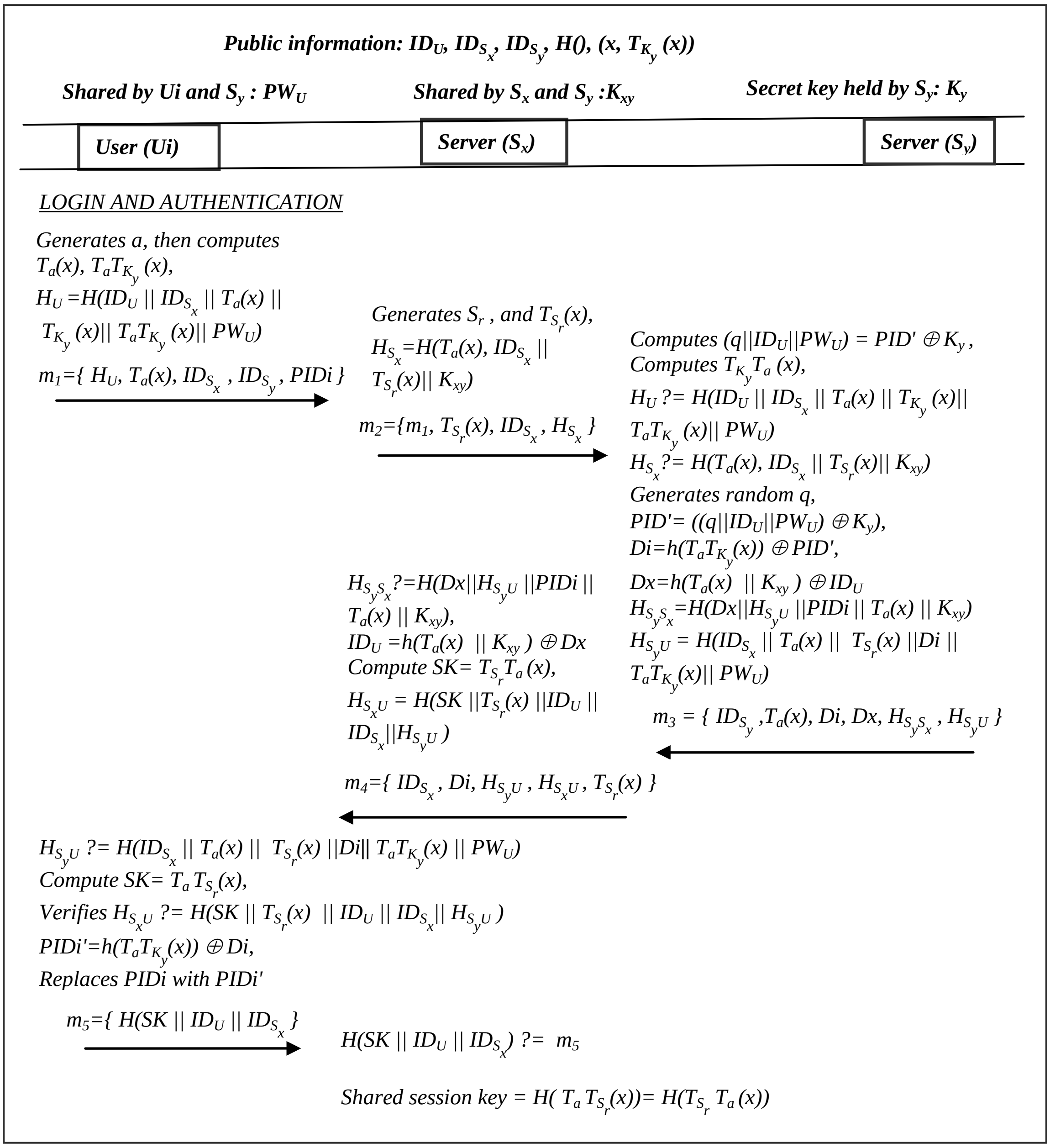

Public information: ID_U, ID_{S_x}, ID_{S_y}, H(), (x, $T_{K_y}(x)$)
Shared by Ui and S_y : PW_U
Shared by S_x and S_y :K_{xy}
Secret key held by S_y: K_y
User (Ui)
Server (S_x)
Server (S_y)
LOGIN AND AUTHENTICATION
Generates a, then computes
$T_a(x)$, $T_aT_{K_y}(x)$,
$H_U = H(ID_U \| ID_{S_x} \| T_a(x) \| T_{K_y}(x) \| T_aT_{K_y}(x) \| PW_U)$
$m_1 = \{ H_U, T_a(x), ID_{S_x}, ID_{S_y}, PIDi \}$
Generates S_r , and $T_{S_r}(x)$,
$H_{S_x} = H(T_a(x), ID_{S_x} \| T_{S_r}(x) \| K_{xy})$
$m_2 = \{m_1, T_{S_r}(x), ID_{S_x}, H_{S_x} \}$
Computes $(q\|ID_U\|PW_U) = PID' \oplus K_y$,
Computes $T_{K_y}T_a(x)$,
H_U ?= $H(ID_U \| ID_{S_x} \| T_a(x) \| T_{K_y}(x) \| T_aT_{K_y}(x) \| PW_U)$
H_{S_x} ?= $H(T_a(x), ID_{S_x} \| T_{S_r}(x) \| K_{xy})$
Generates random q,
$PID' = ((q\|ID_U\|PW_U) \oplus K_y)$,
$Di = h(T_aT_{K_y}(x)) \oplus PID'$,
$Dx = h(T_a(x) \| K_{xy}) \oplus ID_U$
$H_{S_yS_x} = H(Dx\|H_{S_yU} \|PIDi \| T_a(x) \| K_{xy})$
$H_{S_yU} = H(ID_{S_x} \| T_a(x) \| T_{S_r}(x) \|Di \| T_aT_{K_y}(x) \| PW_U)$
$m_3 = \{ ID_{S_y}, T_a(x), Di, Dx, H_{S_yS_x}, H_{S_yU} \}$
$H_{S_yS_x}$?= $H(Dx\|H_{S_yU} \|PIDi \| T_a(x) \| K_{xy})$,
$ID_U = h(T_a(x) \| K_{xy}) \oplus Dx$
Compute SK= $T_{S_r}T_a(x)$,
$H_{S_xU} = H(SK \|T_{S_r}(x) \|ID_U \| ID_{S_x}\|H_{S_yU})$
$m_4 = \{ ID_{S_x}, Di, H_{S_yU}, H_{S_xU}, T_{S_r}(x) \}$
H_{S_yU} ?= $H(ID_{S_x} \| T_a(x) \| T_{S_r}(x) \|Di\| T_aT_{K_y}(x) \| PW_U)$
Compute SK= $T_a T_{S_r}(x)$,
Verifies H_{S_xU} ?= $H(SK \| T_{S_r}(x) \| ID_U \| ID_{S_x}\| H_{S_yU})$
$PIDi' = h(T_aT_{K_y}(x)) \oplus Di$,
Replaces PIDi with PIDi'
$m_5 = \{ H(SK \| ID_U \| ID_{S_x} \}$
$H(SK \| ID_U \| ID_{S_x})$?= m_5
Shared session key = $H(T_a T_{S_r}(x)) = H(T_{S_r} T_a(x))$

Fig. 2. Login and authentication phase (Proposed Model)

6. Next, S_x verifies the message m_5 by confirming the equality check $H(SK \| ID_U \| ID_{S_x})$?= m_5. If true, then it establishes the shared session key as $H(T_{S_r} T_a(x))$. Hence, both parties user and S_x, now share the session key as $SK = H(T_a T_{S_r}(x)) = H(T_{S_r} T_a(x))$.

4.3 Password modification phase

In the password modification phase, both participants U_i and S_y mutually authenticate each other before updating the password, as shown in **Fig. 3**. The procedure for password modification has been described below.

1. The user sends a password modification request to server S_y.
2. S_y generates a nonce n_1 and sends to U_i as a challenge.
3. Next, the user generates a, and computes $T_a(x)$, $T_aT_{K_y}(x)$, $Q=h(T_aT_{K_y}(x)) \oplus PW_U'$, $H_U = H(ID_U || ID_{S_y} || T_a(x) || T_{K_y}(x) || T_aT_{K_y}(x) || Q || PW_U || PW_U' || n_1)$ by assuming new password PW_U'. Now user sends the message $m_1=\{T_a(x), H_U, ID_{S_y}, Q, PIDi\}$ to server S_y to update the password.
4. The server S_y computes $(q||ID_U||PW_U) = PID' \oplus K_y$, $T_{K_y}T_a(x)$, $PW_U' = h(T_aT_{K_y}(x)) \oplus Q$ and checks the equation $H'_U \mathrel{?=} H(ID_U || ID_{S_y} || T_a(x) || T_{K_y}(x) || T_aT_{K_y}(x) || Q || PW_U || PW_U' || n_1)$. If this holds true, it generates random number q and computes $PID' = ((q||ID_U||PW_U') \oplus K_y)$, $r=h(T_aT_{K_y}(x) || ID_U || PW_U)$, $r1=E_r(PIDi')$, $H_{S_yU} = H(ID_U || ID_{S_y} || T_a(x) || PIDi' || T_aT_{K_y}(x) || r1 || PW_U || PW_U')$. Finally it sends message $m_2 = \{ID_{S_y}, r1, H_{S_yU}\}$ to user.
5. The user computes $r=h(T_aT_{K_y}(x) || ID_U || PW_U)$, $r1=D_r(PIDi')$ and verifies $H_{S_yU} \mathrel{?=} H(ID_U || ID_{S_y} || T_a(x) || PIDi' || T_aT_{K_y}(x) || r1 || PW_U || PW_U')$. If it holds true, then updates PW_U as PW_U' and replaces $PIDi$ as $PIDi'$.

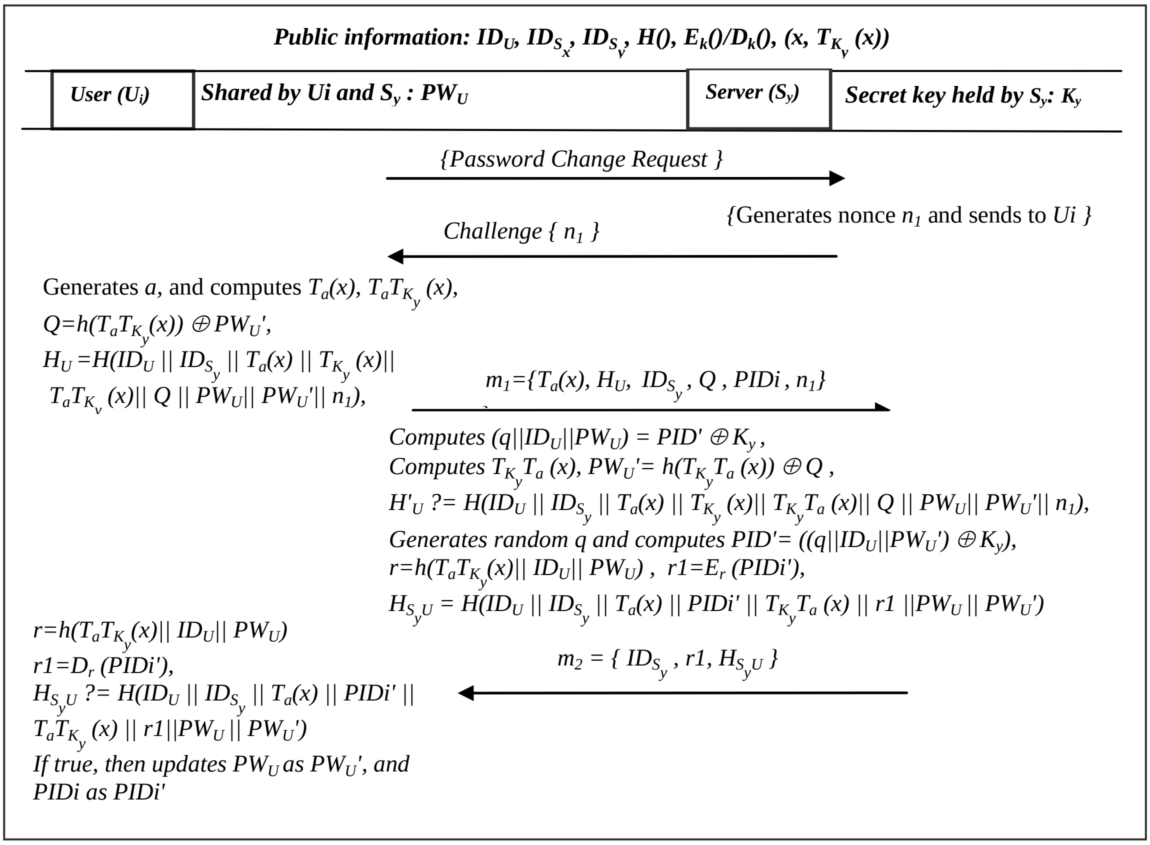

Public information: ID_U, ID_{S_x}, ID_{S_y}, $H()$, $E_k()/D_k()$, $(x, T_{K_y}(x))$
User (U_i)
Shared by Ui and S_y : PW_U
Server (S_y)
Secret key held by S_y: K_y
{Password Change Request }
{Generates nonce n_1 and sends to Ui }
Challenge { n_1 }
Generates a, and computes $T_a(x)$, $T_aT_{K_y}(x)$,
$Q=h(T_aT_{K_y}(x)) \oplus PW_U'$,
$H_U = H(ID_U || ID_{S_y} || T_a(x) || T_{K_y}(x) ||$
$T_aT_{K_y}(x) || Q || PW_U || PW_U' || n_1)$,
$m_1=\{T_a(x), H_U, ID_{S_y}, Q, PIDi, n_1\}$
Computes $(q||ID_U||PW_U) = PID' \oplus K_y$,
Computes $T_{K_y}T_a(x)$, $PW_U' = h(T_{K_y}T_a(x)) \oplus Q$,
$H'_U \mathrel{?=} H(ID_U || ID_{S_y} || T_a(x) || T_{K_y}(x) || T_{K_y}T_a(x) || Q || PW_U || PW_U' || n_1)$,
Generates random q and computes $PID' = ((q||ID_U||PW_U') \oplus K_y)$,
$r=h(T_aT_{K_y}(x) || ID_U || PW_U)$, $r1=E_r(PIDi')$,
$H_{S_yU} = H(ID_U || ID_{S_y} || T_a(x) || PIDi' || T_{K_y}T_a(x) || r1 || PW_U || PW_U')$
$m_2 = \{ ID_{S_y}, r1, H_{S_yU} \}$
$r=h(T_aT_{K_y}(x) || ID_U || PW_U)$
$r1=D_r(PIDi')$,
$H_{S_yU} \mathrel{?=} H(ID_U || ID_{S_y} || T_a(x) || PIDi' ||$
$T_aT_{K_y}(x) || r1 || PW_U || PW_U')$
If true, then updates PW_U as PW_U', and
PIDi as PIDi'

Fig. 3. Password modification phase in the Proposed Model

4.4 Shared key update phase

In shared key update phase, the servers S_x and S_y update their shared keys K_{xy} after getting mutually authenticated, as shown in **Fig. 4**. The procedure of shared key update phase has been elaborated below.

1. The server S_x generates a random integer S_r, then compute $T_{S_r}(x)$, = $T_{S_r}\, T_{K_y}(x)$, $H_{S_x}=H(ID_{S_x} \,||\, ID_{S_y} \,||\, T_{S_r}(x)||\, K_{xy} \,||\, K_{xy}' \,||\, T_{S_r}T_{K_y}(x))$, and $C_1= K_{xy}\oplus K_{xy}'\oplus T_{S_r}T_{K_y}(x)$. Finally, it sends message $m_1=\{ T_{S_r}(x), ID_{S_x}, ID_{S_y}, H_{S_x}, C_1\}$ to S_y for updating shared key K_{xy}.
2. The server S_y receives the message $m_1=\{ T_{S_r}(x), ID_{S_x}, ID_{S_y}, H_{S_x}, C_1\}$ and compute $T_{K_y}T_{S_r}(x)$, $K_{xy}'=\{K_{xy}\oplus T_{K_y}\, T_{S_r}(x)\oplus C_1\}$, $H_{S_x}'= H(ID_{S_x} \,||\, ID_{S_y} \,||\, T_{S_r}(x)||\, K_{xy}||\, K_{xy}'||\, T_{S_r}T_{K_y}(x))$. Next, it verifies $H_{S_x}' ?= H_{S_x}$. If true, then it generates $\mathrm{S_N}$ and further computes $T_{S_N}(x)$, $T_{S_r}T_{S_N}(x)$, $H_{S_yS_x} = H(ID_{S_y} \,||\, ID_{S_x} \,||\, T_{S_r}T_{S_N}(x)||\, K_{xy} \,||\, h(K_{xy}'))$ and $C_2 = K_{xy}\oplus h(K_{xy}')\oplus T_{S_r}T_{S_N}(x)$. Finally, it sends $m_1=\{ T_{S_N}(x), ID_{S_x}, ID_{S_y}, H_{S_yS_x}, C_2\}$ *to* S_x for verification.
3. The S_x, then computes $T_{S_N}\, T_{S_r}(x)$, and $h(K_{xy}')=\{K_{xy}\oplus T_{S_N}\, T_{S_r}(x) \oplus C_2\}$. Then it verifies $H_{S_yS_x}' ?= H(ID_{S_y}||\, ID_{S_x} \,||\, T_{S_N}\, T_{S_r}(x)||\, K_{xy} \,||\, h(K_{xy}'))$. On finding the equality match, it updates
$K_{xy} = K_{xy}'$, and sends the message $m_3=\{H(T_{S_N}T_{S_r}(x)||ID_{S_y}||K_{xy}||K_{xy}')\}$ to S_y for acknowledgement.
4. The server S_y, verifies the equation $(T_{S_N}\, T_{S_r}(x)||\, ID_{S_y} \,||\, K_{xy} \,||\, K_{xy}') ?= m_3$. If it is true, then updates its key as $(K_{xy} = K_{xy}')$ and stores in its database against the right user ID.

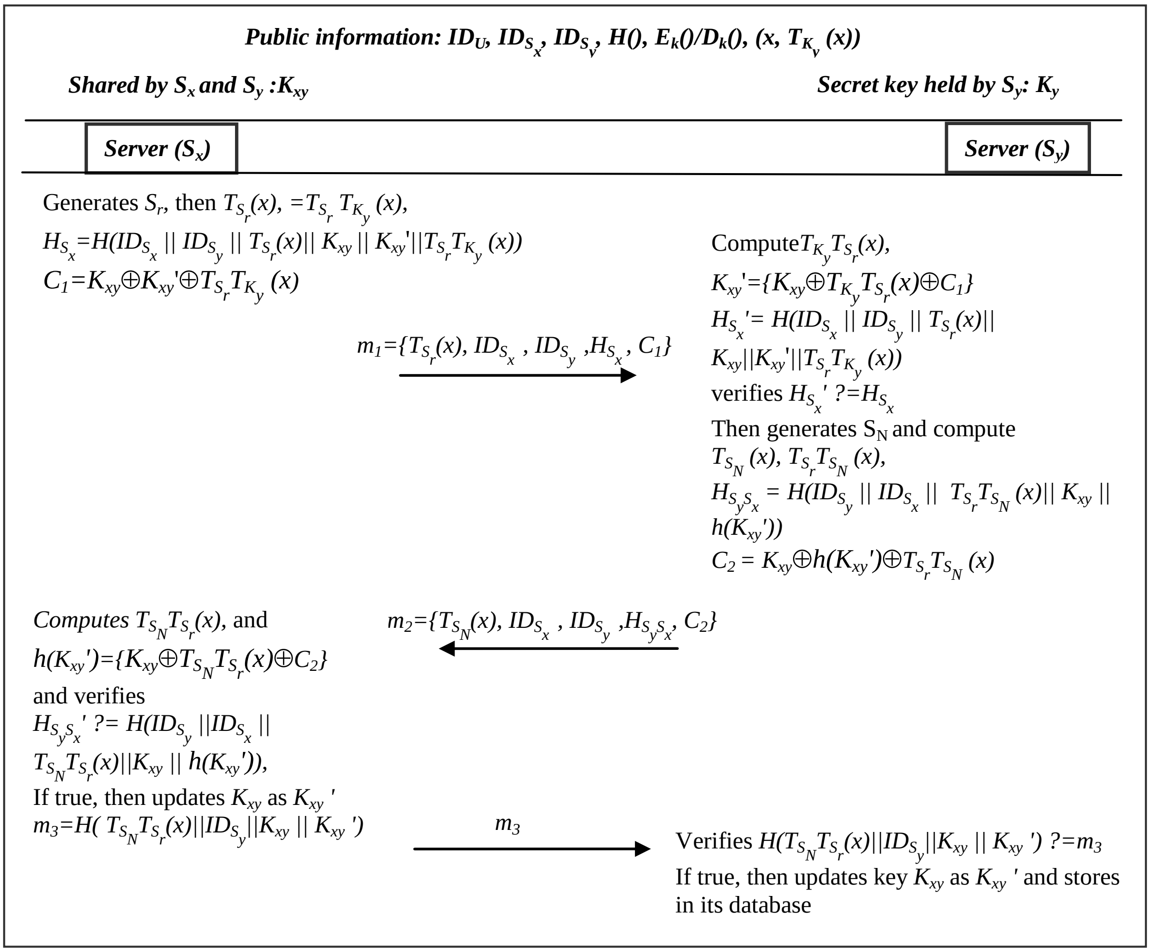

Public information: ID_U, ID_{S_x}, ID_{S_y}, $H()$, $E_k()/D_k()$, $(x, T_{K_y}(x))$
Shared by S_x and S_y :K_{xy}
Secret key held by S_y: K_y
Server (S_x)
Server (S_y)
Generates S_r, then $T_{S_r}(x)$, $=T_{S_r}\, T_{K_y}(x)$,
$H_{S_x}=H(ID_{S_x} \,||\, ID_{S_y} \,||\, T_{S_r}(x)||\, K_{xy} \,||\, K_{xy}'||T_{S_r}T_{K_y}(x))$
$C_1=K_{xy}\oplus K_{xy}'\oplus T_{S_r}T_{K_y}(x)$
$m_1=\{T_{S_r}(x), ID_{S_x}, ID_{S_y}, H_{S_x}, C_1\}$
Compute$T_{K_y}T_{S_r}(x)$,
$K_{xy}'=\{K_{xy}\oplus T_{K_y}T_{S_r}(x)\oplus C_1\}$
$H_{S_x}'= H(ID_{S_x} \,||\, ID_{S_y} \,||\, T_{S_r}(x)||$
$K_{xy}||K_{xy}'||T_{S_r}T_{K_y}(x))$
verifies $H_{S_x}' ?=H_{S_x}$
Then generates $\mathrm{S_N}$ and compute
$T_{S_N}(x)$, $T_{S_r}T_{S_N}(x)$,
$H_{S_yS_x} = H(ID_{S_y} \,||\, ID_{S_x} \,||\, T_{S_r}T_{S_N}(x)||\, K_{xy} \,||$
$h(K_{xy}'))$
$C_2 = K_{xy}\oplus h(K_{xy}')\oplus T_{S_r}T_{S_N}(x)$
$m_2=\{T_{S_N}(x), ID_{S_x}, ID_{S_y}, H_{S_yS_x}, C_2\}$
Computes $T_{S_N}T_{S_r}(x)$, and
$h(K_{xy}')=\{K_{xy}\oplus T_{S_N}T_{S_r}(x)\oplus C_2\}$
and verifies
$H_{S_yS_x}' ?= H(ID_{S_y} \,||ID_{S_x} \,||$
$T_{S_N}T_{S_r}(x)||K_{xy} \,||\, h(K_{xy}'))$,
If true, then updates K_{xy} as K_{xy}'
$m_3=H(T_{S_N}T_{S_r}(x)||ID_{S_y}||K_{xy} \,||\, K_{xy}')$
m_3
Verifies $H(T_{S_N}T_{S_r}(x)||ID_{S_y}||K_{xy} \,||\, K_{xy}') ?=m_3$
If true, then updates key K_{xy} as K_{xy}' and stores in its database

Fig. 4. Shared key update phase

5. SECURITY ANALYSIS

This section illustrates the security proof, formal model-based security analysis, and performance efficiency analysis of the proposed model.

5.1 Security Proof

The security proof from various threats, for the proposed scheme has been elaborated as below:

5.1.1Mutual authentication

Mutual authentication suggests that the participating entities authenticate one another in the same authentication protocol. The proposed scheme complies mutual authentication, since the server S_y authenticates the user U_i by computing $H'_U = H(ID_U \,||\, ID_{S_x} \,||\, T_a(x) \,||\, T_{K_y}(x) \,||\, T_a T_{K_y}(x) \,||\, PW_U)$, and checking $H'_U \;?= \; H_U$. Likewise, the user also authenticates S_y and in turn S_x, by checking the equality first $H_{S_y U} \;?= H(ID_{S_x} \,||\, T_a(x) \,||\, T_{S_r}(x) \,||Di||\, T_a T_{K_y}(x) \,||\, PW_U)$ for verifying S_y. Then S_x, by computing $SK = T_a T_{S_r}(x)$, and checking the equation $H_{S_x U} \;?= H(SK| \; T_{S_r}(x) \,||\, ID_U \,||\, ID_{S_x} ||\, H_{S_y U})$.The S_y knows the fact, that PW_U is only known by U_i. Likewise, the U_i believes that these passwords are only in the knowledge of S_y, where from the trust is smoothly transferred towards S_x.

5.1.2Impersonation Attack

This attack can be launched by an adversary who may act as a silent mediator among the legitimate participants. This might let the participants perceive one another as the actual parties; however these are not the right participants, though. The proposed scheme is secure, since, an adversary cannot construct $H_U = H(ID_U \,||\, ID_{S_x} \,||\, T_a(x) \,||\, T_{K_y}(x) \,||\, T_a T_{K_y}(x) \,||\, PW_U)$. Similarly, $H_{S_y U} = H(ID_{S_x} \,||\, T_a(x) \,||\, T_{S_r}(x) \,||Di||\, T_a T_{K_y}(x) \,||\, PW_U)$ sent by S_y to S_x, and in turn to U_i, cannot be reproduced by an adversary since the passwords PW_U is only known to S_y. Any kind of impersonation attack will be successfully foiled by the legal participants by verifying the checks for $H_{S_x U} \;?= H(SK \,||\, T_{S_r}(x) \,||\, ID_U \,||\, ID_{S_x} ||\, H_{S_y U})$ and $H(SK \,||\, ID_U \,||\, ID_{S_x}) \;?= \; m_5$. If any of these checks fail, the session is aborted by the concerned participant.

5.1.3Replay Attack

The replay attacks could be initiated by an adversary by replaying the intercepted content messages to forge or impersonate any legitimate participant. An attacker may intercept the messages m_1, m_2, m_3, m_4 and m_5 on an insecure channel, and attempt to replay in other timings. In proposed scheme, this replay attack can be easily foiled by verifying the equality checks for $H_{S_x U} \;?= H(SK \,||\, T_{S_r}(x) \,||\, ID_U \,||\, ID_{S_x} ||\, H_{S_y U})$ and $H(SK \,||\, ID_U \,||\, ID_{S_x}) \;?= \; m_5$, since those equality checks will not hold true for any old values of $T_a(x)$ and $T_{S_r}(x)$. In this manner, impersonation attack will be defeated by any of the legal participant verifying the check.

5.1.4Known-Key Security

The known-key security signifies towards guessing the private keys of the related participants, on condition that the session key is compromised. For instance, if the session key $H(T_a T_{S_r}(x)) = H(T_{S_r} T_a(x))$ gets exposed, it will not let the attacker guess any S_y, S_x or U_i secrets i.e., a, PW_U, S_r, *or* K_{xy}, K_y. Although PW_U is a low entropy password, it would be a hard problem to guess the password from public messages. The rest of the secret parameters a, S_r, K_{xy}, K_y are very hard to guess for being high entropy integers.

5.1.5Perfect Forward Secrecy

The perfect forward secrecy makes certain that previous session keys are protected, if the long-term private secrets of any of the participants gets compromised. The proposed scheme fulfills the rule of perfect forward secrecy, since the disclosure of S_y, S_x and U_i's secrets i.e., PW_U, K_{xy}, K_y cannot lead to the recovery of session based temporary parameters a and S_r, as supposed by S_x and U_i. Besides, an adversary cannot recover the secrets i.e. a or S_r, from public keys $(x, T_a(x))$ and $(x, T_{S_r}(x))$, which is a hard problem due to CMDLP.

5.1.6Data Integrity

The data integrity ensures the intact delivery of message as sent by the sender to a receiver without any modification. The proposed scheme ensures the data integrity as the three corresponding entities are capable enough to detect any modification in the generated hash functions at various levels i.*e*., H_U (produced at U_i), H_{S_x}, H_{S_xU} (produced at S_x), and $H_{S_yS_x}$, H_{S_yU} (produced at S_y).

5.1.7Guessing Attacks

In these attacks, an attacker might intercept all public messages available on insecure channel among intended participants. Then, it may attempt to guess information by trying all different possible combinations of low entropy parameters. As far, the secrets for the participants, PW_U, K_{xy}, K_y , the PW_U is a low entropy password that can be attempted by an adversary for guessing it from intercepted messages i.e. $\{H_U$ in m_1, and H_{S_yU} in $m_3\}$. However, an adversary cannot extract password PW_U from $H_U = H(ID_U \,||\, ID_{S_x} \,||\, T_a(x)||T_{K_y}(x)||T_aT_{K_y}(x)||PW_U)$ and $H_{S_yU} = H(ID_{S_x}||T_a(x)|| \; T_{S_r}(x)||Di||T_aT_{K_y}(x)||PW_U)$. Since, the adversary does not know about the parameter $T_aT_{K_y}(x)$, which is only known by the user and server, having the secrets (a and K_y). Hence, the adversary cannot guess the password in polynomial time.

5.1.8Session Key Security

The session key security signifies that the constructed session key is only shared among the legitimate participants, i.e., U_i and S_x, and nobody else. In proposed scheme, an attacker can not forge or impersonate, since not having the legitimate secrets and passwords. Hence, the modification in the message could not be successful for not being capable of generating the valid hash $\mathrm{H_U}$ with a secret password PW_U. Hence, the proposed scheme provides session key security.

5.1.9Anonymity

An anonymous authentication protocol provides anonymity to U_i, besides authentication, and an adversary cannot discern the identity of any of the participants by accessing the intercepted open message parameters. The identity for the user in proposed scheme has been masked behind pseudonym *PIDi*. The U_i sends *PIDi* instead of ID_U every time it initiates a session, while the $\mathrm{S_y}$ extracts ID_U and PW_U from *PIDi* by using its secret key K_y. In this manner the user enjoys anonymity and untraceability. The user in our scheme cannot be traced, since the user sends a novel *PIDi* for every session, which is updated and generated in every session by $\mathrm{S_y}$, and secretly communicated towards user. This way, an adversary cannot derive or guess that

ID_U. While, the Yoon's [47] and Zhu [42], both fail to provide anonymity to user, owing to the exposure of users' identities during login request.

5.1.10 Key compromise impersonation attack

An adversary tries for impersonating an entity B to another entity A, while the later is malicious and will accept the session with B as one of the session peers, however, B hadn't meant to establish a session with A, currently. In proposed protocol, an adversary can steal the user's password though. It will also have to know proper identity before launching this attack, and cheat S_x or S_y. The attack is foiled, since an adversary does not know the shared key K_{xy}, hence, it cannot generate the proper $H_{S_yS_x}$ and H_{S_yU}.

5.1.11 Resistance to Denial-of-Service attack and stolen-verifier attack

The proposed scheme provides resistance to DoS attack and stolen-verifier attacks. The main reason for that is the proposed scheme does not maintain any database or repository at the server S_y's end. This feature provides an edge to the proposed scheme against DoS and stolen-verifier attacks.

5.2 Formal Security Analysis

This section covers the formal security analysis of proposed scheme under Burrows-Abadi-Needham (BAN) logic [46] , while, this model analyzes the security based on mutual authentication, key distribution, and the strength against session key disclosure. In this logic analysis, *Principals* are such agents that are involved in a protocol, while *Keys* are to be used for symmetric message encryption.

Few notations that have been used in the BAN security analysis are given as follows:
$P \mid\equiv X$: The principal P believes X, or alternatively, P believes the statement X.
$P \triangleleft X$: P sees X. P receives some message X and may read or repeat it in any message.
$P \mid\sim X$: P once said X. Earlier in time; P had sent some message X and P believed that message.
$P \Rightarrow X$: P has got jurisdiction over X; or P has authority over X and could be trusted.
$\sharp(X)$: The message X may be treated as fresh.
(X, Y): X or Y being the part of message (X, Y).
$\langle X\rangle_Y$: The formulae X is combined with formulae Y.
$\{X, Y\}_K$: X or Y is encrypted with the key K.
$(X, Y)_K$: X or Y is hashed with the key K.
$P \xleftrightarrow{K} Q$: P and Q can communicate with the shared key K.

Some rules or logical postulates used in the BAN Logic are given as follows:

Rule 1. Message meaning rule: $\frac{P\mid\equiv P \overset{K}{\leftrightarrow} Q,\ \ P\triangleleft\langle X\rangle_Y}{P\mid\equiv Q \mid\sim X}$

Rule 2. Nonce verification rule: $\frac{P\mid\equiv \sharp(X),\ \ P\mid\equiv Q\mid\sim X}{P\mid\equiv Q\mid\equiv X}$

Rule 3. Jurisdiction rule: $\frac{P\mid\equiv Q \Rightarrow X,\ \ P\mid\equiv Q\mid\equiv X}{P\mid\equiv X}$

Rule 4. Freshness conjuncatenation rule: $\frac{P\mid\equiv \sharp(X)}{P\mid\equiv \sharp(X,\ Y)}$

Rule 5. Belief rule: $\frac{P\mid\equiv(X),\ \ P\mid\equiv(Y)}{P\mid\equiv(X,\ Y)}$

Rule 6. Session keys rule: $\frac{P\mid\equiv \sharp(X),\ \ P\mid\equiv Q\mid\equiv X}{P\mid\equiv P \overset{K}{\leftrightarrow} Q}$

The proposed protocol needs to satisfy the following goals to ensure its security under BAN logic, using the above assumptions and postulates.

Goal 1 : $S_x \mid\equiv S_x \xleftrightarrow{SK} U_i$

Goal 2 : $S_x \mid\equiv U_i \mid\equiv S_x \xleftrightarrow{SK} U_i$

Goal 3 : $U_i \mid\equiv S_x \xleftrightarrow{SK} U_i$

Goal 4 : $U_i \mid\equiv S_x \mid\equiv S_x \xleftrightarrow{SK} U_i$

Goal 5 : $S_y \mid\equiv S_y \xleftrightarrow{T_{K_y} T_a (x)} U_i$

Goal 6 : $S_y \mid\equiv U_i \mid\equiv S_y \xleftrightarrow{T_{K_y} T_a (x)} U_i$

Goal 7 : $U_i \mid\equiv S_y \xleftrightarrow{T_a T_{K_y} (x)} U_i$

Goal 8 : $U_i \mid\equiv \mathrm{S_y} \mid\equiv S_y \xleftrightarrow{T_a T_{K_y} (x)} U_i$

Initially, messages exchanged in the proposed protocol can be transformed into idealized form in the following manner.

M$_1$: $U_i \rightarrow S_x$: $H_U, T_a(x), ID_{S_x}, ID_{S_y}, PIDi$:

$\{ \langle ID_U \,||\, ID_{S_x} \,||\, T_a(x) \,||\, T_{K_y}(x) \rangle_{(T_aT_{K_y}(x)||\, PW_U)}, T_a(x), ID_{S_x}, ID_{S_y}, PIDi \}$

M$_2$: $S_x \rightarrow S_y$: $m_1, T_{S_r}(x), ID_{S_x}, H_{S_x}$: $\{ m_1, T_{S_r}(x), ID_{S_x}, \langle T_a(x), ID_{S_x} \,||\, T_{S_r}(x) \rangle_{K_{xy}} \}$

M$_3$: $S_y \rightarrow S_x$: $ID_{S_y}, T_a(x), Di, Dx, H_{S_yS_x}, H_{S_yU}$:

$\{ID_{S_y}, T_a(x), Di, Dx,, \langle Dx||H_{S_yU}||PIDi||T_a(x) \rangle_{K_{xy}}, \langle ID_{S_x} \,||\, T_a(x) \,||\, T_{S_r}(x) \,||Di \rangle_{(T_{K_y}T_a(x)||\, PW_U)}\}$

M$_4$: $S_x \rightarrow U_i$: $ID_{S_x}, Di, H_{S_yU}, H_{S_xU}, T_{S_r}(x)$:

$\{ID_{S_x}, Di, \langle ID_{S_x} \,||\, T_a(x) \,||\, T_{S_r}(x) \,||Di \rangle_{(T_{K_y}T_a(x)||\, PW_U)}, \langle T_{S_r}(x) \,||ID_U \,||\, ID_{S_x}||\, H_{S_yU} \rangle_{(T_{S_r}T_a(x))}, T_{S_r}(x)\}$

M$_5$: $U_i \rightarrow \mathrm{S_x}$: m5: $\{\langle ID_U \,||\, ID_{S_x} \rangle_{(T_aT_{S_r}(x))} \}$:

Secondly, the following premises are established to prove the security of proposed protocol.

P1 : $U_i \mid\equiv \sharp a$

P2 : $\mathrm{S_x} \mid\equiv \sharp S_r$

P3 : $\mathrm{S_y} \mid\equiv \sharp K_y$

P4 : $U_i \mid\equiv U_i \xleftrightarrow{PW_U} \mathrm{S_y}$

P5 : $U_i \mid\equiv U_i \xleftrightarrow{T_aT_{S_r}(x)} \mathrm{S_x}$

P6 : $\mathrm{S_x} \mid\equiv \mathrm{S_x} \xleftrightarrow{T_{S_r} T_a(x)} U_i$

P7 : $\mathrm{S_x} \mid\equiv \mathrm{S_x} \xleftrightarrow{K_{xy}} \mathrm{S_y}$

P8 : $\mathrm{S_y} \mid\equiv \mathrm{S_y} \xleftrightarrow{PW_U} U_i$

P9 : $\mathrm{S_y} \mid\equiv \mathrm{S_y} \xleftrightarrow{K_{xy}} \mathrm{S_x}$

P10 : $U_i \mid\equiv \mathrm{S_x} \Rightarrow T_{S_r}(x)$

P11 : $\mathrm{S_x} \mid\equiv U_i \Rightarrow T_a(x)$

P12 : $\mathrm{S_y} \mid\equiv U_i \Rightarrow T_a(x)$

P13: $\mathrm{S_x} \mid\equiv \mathrm{S_y} \Rightarrow T_{K_y}(x)$

P14 : $\mathrm{S_y} \mid\equiv \mathrm{S_x} \Rightarrow T_{S_r}(x)$

P15 : $U_i \mid\equiv \mathrm{S_y} \Rightarrow T_{K_y}(x)$

Thirdly, the idealized form i.e., M$_1$-M$_5$ of the proposed protocol can be examined and verified in the light of above mentioned postulates and assumptions.

Considering the M1 and M2 of the idealized form:

$\mathbf{M_1}$: $U_i \rightarrow S_x : H_U, T_a(x), ID_{S_x}, ID_{S_y}, PIDi$:

$\{ \langle ID_U \| ID_{S_x} \| T_a(x) \| T_{K_y}(x)\rangle_{(T_aT_{K_y}(x)\| PW_U)}, T_a(x), ID_{S_x}, ID_{S_y}, PIDi \}$

$\mathbf{M_2}$: $S_x \rightarrow S_y : m_1, T_{S_r}(x), ID_{S_x}, H_{S_x}$: $\{ m_1, T_{S_r}(x), ID_{S_x}, \langle T_a(x), ID_{S_x} \| T_{S_r}(x)\rangle_{K_{xy}} \}$

By applying seeing rule for M1 and M2, we get

D1: $S_x \lhd \{\langle ID_U \| ID_{S_x} \| T_a(x) \| T_{K_y}(x)\rangle_{(T_aT_{K_y}(x)\| PW_U)}, T_a(x), ID_{S_x}, ID_{S_y}, PIDi \}$

D2: $S_y \lhd \{ m_1, T_{S_r}(x), ID_{S_x}, \langle T_a(x), ID_{S_x} \| T_{S_r}(x)\rangle_{K_{xy}} \}$

According to D1, D2, P8, P9 and message meaning rule, we get

D3: $S_y \mid\equiv U_i \sim \{\langle ID_U \| ID_{S_x} \| T_a(x) \| T_{K_y}(x)\rangle_{(T_aT_{K_y}(x)\| PW_U)}, T_a(x), ID_{S_x}, ID_{S_y}, PIDi \}$

D4: $S_y \mid\equiv S_x \sim \{ m_1, T_{S_r}(x), ID_{S_x}, \langle T_a(x), ID_{S_x} \| T_{S_r}(x)\rangle_{K_{xy}} \}$

According to D3, P1, freshness conjucatenation and nonce verification rules we get

D5: $S_y \mid\equiv U_i \mid\equiv \{\langle ID_U \| ID_{S_x} \| T_a(x) \| T_{K_y}(x)\rangle_{(T_aT_{K_y}(x)\| PW_U)}, T_a(x), ID_{S_x}, ID_{S_y}, PIDi \}$

According to D4, P2, freshness conjucatenation and nonce verification rules, we get

D6: $S_y \mid\equiv S_x \mid\equiv \{ m_1, T_{S_r}(x), ID_{S_x}, \langle T_a(x), ID_{S_x} \| T_{S_r}(x)\rangle_{K_{xy}} \}$

According to D5, P12, and Jurisdiction rule

D7: $S_y \mid\equiv \{\langle ID_U \| ID_{S_x} \| T_a(x) \| T_{K_y}(x)\rangle_{(T_aT_{K_y}(x)\| PW_U)}, T_a(x), ID_{S_x}, ID_{S_y}, PIDi \}$

According to D6, P14, and Jurisdiction rule

D8: $S_y \mid\equiv \{ m_1, T_{S_r}(x), ID_{S_x}, \langle T_a(x), ID_{S_x} \| T_{S_r}(x)\rangle_{K_{xy}} \}$

According to D5, D7 and session key rule, we get

D9: $S_y \mid\equiv S_y \xleftrightarrow{T_{K_y}T_a(x)} U_i$ **(Goal 5)**

According to D5, D7, P8 and nonce-verification rule, we get

D10: $S_y \mid\equiv U_i \mid\equiv S_y \xleftrightarrow{T_{K_y}T_a(x)} U_i$ **(Goal 6):**

Considering the M3 of the idealized form:

$\mathbf{M_3}$: $S_y \rightarrow S_x : ID_{S_y}, T_a(x), Di, Dx, H_{S_yS_x}, H_{S_yU}$:

$\{ID_{S_y}, T_a(x), Di, Dx,, \langle Dx\|H_{S_yU}\|PIDi\|T_a(x)\rangle_{K_{xy}}, \langle ID_{S_x} \| T_a(x) \| T_{S_r}(x) \|Di \rangle_{(T_{K_y}T_a(x)\| PW_U)}\}$

By applying seeing rule for M3, we get

D11: $S_x \lhd ID_{S_y}, T_a(x), Di, Dx, H_{S_yS_x}, H_{S_yU}$: $\{ID_{S_y}, T_a(x), Di, Dx,, \langle Dx\|H_{S_yU}\|PIDi\|T_a(x)\rangle_{K_{xy}}, \langle ID_{S_x} \| T_a(x) \| T_{S_r}(x) \|Di \rangle_{(T_{K_y}T_a(x)\| PW_U)}\}$

For D11, P7 and message meaning rule, we get

D12: $S_x \mid\equiv U_i \sim \{ID_{S_y}, T_a(x), Di, Dx,, \langle Dx\|H_{S_yU}\|PIDi\|T_a(x)\rangle_{K_{xy}}, \langle ID_{S_x} \| T_a(x) \| T_{S_r}(x) \| Di \rangle_{(T_{K_y}T_a(x)\| PW_U)}\}$

According to D12, P2, P6, P13, freshness conjucatenation and nonce verification rules we get

D13: $S_x \mid\equiv U_i \mid\equiv \{ID_{S_y}, T_a(x), Di, Dx,, \langle Dx\|H_{S_yU}\|PIDi\|T_a(x)\rangle_{K_{xy}}, \langle ID_{S_x} \| T_a(x) \| T_{S_r}(x) \| Di \rangle_{(T_{K_y}T_a(x)\| PW_U)}\}$

Next, considering M4 idealized message

$\mathbf{M_4}$: $S_x \rightarrow U_i: ID_{S_x}, Di, H_{S_yU}, H_{S_xU}, T_{S_r}(x)$:

$\{ ID_{S_x}, Di, \langle ID_{S_x} \| T_a(x) \| T_{S_r}(x) \|Di\rangle_{(T_{K_y}T_a(x)\| PW_U)}, \langle T_{S_r}(x) \|ID_U \| ID_{S_x}\| H_{S_yU} \rangle_{(T_{S_r}T_a(x))}, T_{S_r}(x)\}$

By applying seeing rule for M4, we get

D14: $Ui \lhd ID_{S_x}, Di, H_{S_yU}, H_{S_xU}, T_{S_r}(x)$: $\{ ID_{S_x}, Di, \langle ID_{S_x} \| T_a(x) \| T_{S_r}(x) \|Di \rangle_{(T_{K_y}T_a(x)\| PW_U)},$

$\langle T_{S_r}(x)\ \| ID_U \| ID_{S_x} \| H_{S_yU} \rangle_{(T_{S_r}T_a(x))}$, $T_{S_r}(x)\}$

For D14, P4, P5 and message meaning rule, we get

D15: $U_i \mid\equiv S_x \sim \{ ID_{S_x}, Di, \langle ID_{S_x} \| T_a(x) \| T_{S_r}(x) \| Di \rangle_{(T_{K_y}T_a(x) \| PW_U)}, \langle T_{S_r}(x)\ \| ID_U \| ID_{S_x} \| H_{S_yU} \rangle_{(T_{S_r}T_a(x))}, T_{S_r}(x)\}$

According to D15, P2, P3, freshness conjucatenation and nonce verification rules we get

D16: $U_i \mid\equiv S_x \mid\equiv \{ ID_{S_x}, Di, \langle ID_{S_x} \| T_a(x) \| T_{S_r}(x) \| Di \rangle_{(T_{K_y}T_a(x) \| PW_U)},$
$\langle T_{S_r}(x)\ \| ID_U \| ID_{S_x} \| H_{S_yU} \rangle_{(T_{S_r}T_a(x))}, T_{S_r}(x)\}$

For D16, P10, P15 and jurisdiction rule, we get

D17: $U_i \mid\equiv \{ ID_{S_x}, Di, \langle ID_{S_x} \| T_a(x) \| T_{S_r}(x) \| Di \rangle_{(T_{K_y}T_a(x) \| PW_U)}, \langle T_{S_r}(x)\ \| ID_U \| ID_{S_x} \| H_{S_yU} \rangle_{(T_{S_r}T_a(x))}, T_{S_r}(x)\}$

According to D17, we apply the session key rule as

D18: $U_i \mid\equiv S_x \xleftrightarrow{SK} U_i$ **(Goal 3)**

$U_i \mid\equiv S_y \xleftrightarrow{T_aT_{K_y}(x)} U_i$ **(Goal 7)**

For D18, P1 we apply the session key rule as

D19: $U_i \mid\equiv S_x \mid\equiv S_x \xleftrightarrow{SK} U_i$ **(Goal 4)**

$U_i \mid\equiv S_y \mid\equiv S_y \xleftrightarrow{T_aT_{K_y}(x)} U_i$ **(Goal 8)**

Next, considering M5 idealized message

M$_5$: $U_i \rightarrow S_x$: m5: $\{\langle ID_U \| ID_{S_x} \rangle_{(T_aT_{S_r}(x))}\}$

By applying seeing rule for M5, we get

D20: $S_x \triangleleft$ m5: $\{\langle ID_U \| ID_{S_x} \rangle_{(T_aT_{S_r}(x))}\}$

According to D20, P6, P7 and message meaning rule, we get

D21: $S_x \mid\equiv U_i \sim \{\langle ID_U \| ID_{S_x} \rangle_{(T_aT_{S_r}(x))}\}$

According to D21, P1, freshness conjucatenation and nonce verification rules we get

D22: $S_x \mid\equiv U_i \mid\equiv \{\langle ID_U \| ID_{S_x} \rangle_{(T_aT_{S_r}(x))}\}$

According to D22, P11 and jurisdiction rule, we get

D23: $S_x \mid\equiv U_i \mid\equiv \{\langle ID_U \| ID_{S_x} \rangle_{(T_aT_{S_r}(x))}\}$

According to D23, we apply the session key rule as

D24: $S_x \mid\equiv S_x \xleftrightarrow{SK} U_i$ **(Goal 1)**

According to D24, P2 we apply the session key rule as

D25: $S_x \mid\equiv U_i \mid\equiv S_x \xleftrightarrow{SK} U_i$ **(Goal 2)**

The above BAN logic analysis formally proves that the proposed protocol achieves mutual authentication and the session key *SK* is mutually established between U_i and S_x.

At the same time, we implement a formal analysis to prove that the proposed protocol has been secure under random oracle model [47]. For the use of contradiction proof method, in this formal security analysis, we employ two oracles as assumption, as shown below:

reveal1: The Reveal1 oracle gives output ω from the corresponding hash function $y=H(\omega)$, unconditionally.

reveal2: The Reveal2 oracle outputs random number *a* from the public key $T_a(x)$, following Chebyshev Chaotic map [2, 45].

Theorem1

In consideration with the Chaotic maps-based discrete logarithm problem (CMDLP) assumption, the proposed protocol stands secure, if the adversary accesses the public messages like{ m_1 , m_2 , m_3 ,m_4 }, and attemtps to construct a valid session key, given that one way hash function H(.) behaves nearly as a random oracle.

Algorithm 1. $\boldsymbol{EXP1^{HASH}_{\mathrm{ECMAKAMS}}}$

```
Eavesdrop the message $m_1=\{ H_U, T_a(x), ID_{S_x} , ID_{S_y} , PIDi \}$ in the authentication phase.
Eavesdrop the message $m_2=\{m_1, T_{S_r}(x), ID_{S_x} , H_{S_x} \}$ in the authentication phase, and then
Call reveal1 oracle on input $H_{S_x}$ to retrieve $(T_a(x)', ID_{S_x} , T_{S_r}(x)', K_{xy}) \leftarrow reveal1\ (H_{S_x})$
Then, Call reveal1 oracle on input $T_a(x)'$ to retrieve the secret number $a' \leftarrow reveal2\ (T_a(x)')$
Compute $T_aT_{S_r} (x)'$ using $a'$ and $T_{S_r}(x)'$, that is equivalent to parameter as $SK'= T_a T_{S_r} (x)'$
Eavesdrop the message $m_4=\{ ID_{S_x} , Di, H_{S_yU} , H_{S_xU} , T_{S_r}(x)\}$ in the authentication phase, and then
Call reveal1 oracle on input $H_{S_xU}$ to retrieve $(SK'' , ID_U ' , T_{S_r}(x), ID_{S_x}', H_{S_yU} ) \leftarrow reveal1\ (H_{S_xU})$
Next, compute $M= H(SK' \,||\, ID_U '\,||\, T_{S_r}(x) \,||ID_{S_x}' \,||\, H_{S_yU} )$
If $(M= H_{S_xU})$Then
    Accept $ID_U '$ as the correct identity $ID_U$ of  the user $U_i$, and $H(SK')= H(T_a T_{S_r} (x)')$ as the correct session key for $U_i$ and $S_x$
    Return 1 (success)
Else
    Return 0 (failure)
End if
```

Proof. In this proof, a crafty attacker $\mathbb{A}$, capable of approaching the public parameters like {m_1 , m_2 , m_3, m_4 }, makes a use of the random oracles Reveal1 and Reveal2 for implementing the given algorithm $EXP1^{HASH}_{\mathrm{ECMAKAMS}}$. The probability of success for the experiment $EXP1^{HASH}_{\mathrm{ECMAKAMS}}$ is Suc1=Pr.2[$EXP1^{HASH}_{\mathrm{ECMAKAMS}}$=1] - 1, where the probability Pr[E] indicates an event E's probability. Thus, the advantage function related to this experiment becomes as $Adv^{HASH}_{\mathrm{ECMAKAMS}}$ (t_1,q_{R1}, q_{R2})=max $_{\mathbb{A}}$ [$Suc1^{HASH}_{\mathrm{ECMAKAMS}}$], with the execution time t_1 and random Reveal queries q_{R1} and q_{R2} maximized on $\mathbb{A}$. We can call the proposed protocol as verifiably secure against an adversary $\mathbb{A}$ for extracting a legitimate session key *H(SK)*, if $Adv^{HASH}_{PSBASME}$(t_1,q_{R1}, q_{R2}) $\le \varepsilon$ for any sufficiently small $\varepsilon > 0$. For the above experiment, if an attacker $\mathbb{A}$ is assumed to be able to invert a one way hash function *H(.)*, and solve the intractable problem CMDLP, it can easily derive the legal user identity ID_U and shared session key *H(SK)* between U_i and S_x, and as a result wins the game. However, in relation to definition (section 2.3), this is computationally incapable to invert the hash function, since $Adv^{HASH}_{\mathrm{ECMAKAMS}}(t_1) \le \varepsilon$ for any sufficiently small $\varepsilon > 0$.

5.3 Performance Analysis

As we saw previously, that Chebyshev polynomial-based computation is nearly three times more efficient than ECC (Elliptic Curve Cryptography) and RSA based encryption [1-3]. The Chebyshev polynomial-based computation provides lesser key sizes with fast computation, and takes less memory and bandwidth consumption. In proposed scheme, we can say that there are no operations like modular exponentiation or elliptic curve-based scalar multiplication. In this section, the comparison of the cost for Zhu, and Yoon et al. and proposed protocols has been shown, which also employed Chebyshev polynomial map in their presented techniques, and is described below. A few notations used in the comparison are as follows.

T_{XOR}: The time for the execution of XOR operation.

T_H: The time taken for the hash operation;

T_{SYM}: The time for executing symmetric key operation;

T_{ECM}: The time for the execution of elliptic curve-based scalar point multiplication;

T_{CCM}: The time for the execution of Chebyshev Chaotic map operation $T_n(x) \ mod \ p$ using an algorithm [40].

Now, we may compare the costs on the basis of estimation of execution times for different cryptographic operations (using the PBC library, Ubuntu-12.04.1 (32-bit operating system), on 2.4 GHz processor, and 3.0 GB RAM). According to this, taking N and P as 1,024 bits long, computational time of hash function-based operation, symmetric operation (encryption/decryption), elliptic curve-based point multiplication, and chaotic map polynomial operation is 0.00058s, 0.0086s, 0.063165s, and 0.02104s respectively [39-41]. The computational cost of XOR operation is quite negligible as compared with other cryptographic operations, and thus can be ignored. The following **Table 2** depicts the comparision for different security features for three protocols, i.e., Yoon's scheme [38], Zhu [42], and proposed scheme.

Table 2. Comparison for Yoon et al., Zhu's scheme, and Proposed scheme

	Yoon et al. [38]	**Zhu [42]**	**Proposed protocol**
Anonymity	No	No	Yes
Mutual Authentication	Yes	Yes	Yes
Resist Insider Attack	Yes	Yes	Yes
Key Compromise Impersonation	No	Yes	Yes
Resist secret/ password guessing attack	Yes	Yes	Yes
Provides data integrity	Yes	Yes	Yes
Resist Masquerading attack	Yes	Yes	Yes
Resist Replay attack	Yes	Yes	Yes
Session key agreement	Yes	Yes	Yes
Perfect forward secrecy	Yes	Yes	Yes
Known key secrecy	Yes	Yes	Yes
Resists Stolen Verifier attack	Yes	No	Yes
Resists Trace attack	No	No	Yes
Resists Denial-of-Service attack	Yes	No	Yes

Table 3. Estimated cost for Yoon, Zhu, and proposed scheme

	Yoon's protocol [38]	**Zhu [42]**	**Proposed protocol**
Authentication messages	$5T_H + 1\ T_{ECM} \approx 0.12923$	$14T_H + 8T_{SYM} + 4T_{CCM} \approx 0.16108$	$14T_H + 4T_{CCM} \approx 0.09228s$
Password modification phase	$2T_H \approx 0.001s$	$6T_H + 4T_{SYM} + 4T_{CCM} \approx 0.12204$	$10T_H + 2T_{SYM} + 2T_{CCM} \approx 0.06508$
Shared key update among servers	*N/A*	$6T_H + 4T_{SYM} + 4T_{CCM} \approx 0.12204$	$8T_H + 4T_{CCM} \approx 0.08764$
Number of rounds	5	5	5

Therefore, in the light of demonstrated performance efficiency analysis, we can safely deduce that the proposed protocol is more efficient and secure than Zhu's scheme. The authentication phase and shared key update phases are completed at less delay, with an equivalent security, as shown in **Table 3**. While, the password update phase take equal amount of cost for both schemes. On the whole, we can say that our proposed protocol is efficient as far as cost, but also provides additional security features like anonymity.

6. CONCLUDING REMARKS

A multi-server authentication scheme may ensure the provision of services to subscribers by the use of one-time registration from a single server or any registration centre. The current research work comments on the Zhu's multi-server authentication scheme based on Chebyshev polynomial computation. Since, the focus has been shifting from high overhead cryptographic algorithms towards lightweight cryptography. Although, the Zhu's scheme used Chaotic map architecture that incurs much less cost in comparison with its contemporary schemes. Despite, the Zhu scheme was unable to provide anonymity, resistance to traceability, and efficiency. The proposed scheme provides the same level of security along with anonymity, without DoS and stolen-verifier attacks, with less overhead and more efficiency. The scheme also demonstrates formal security analysis and performance efficiency evaluation.

References

[1] Behnia, S., Akhshani, A., Ahadpour, S., Mahmodi, H., & Akhavan, A., "A fast chaotic encryption scheme based on piecewise nonlinear chaotic maps," *Physics Letters A*, 366, 391–396, 2007. Article (CrossRef Link)

[2] Baptista, M. S. (1998). Cryptography with chaos. Physics Letters A, 240, 50–54. Article (CrossRef Link)

[3] Xiao, D., Liao, X.,&Wong, K., "An efficient entire chaos-based scheme for deniable authentication," *Chaos Solitons Fractals*, 23, 1327–1331, 2005. Article (CrossRef Link)

[4] Li, X., Niu, J., Kumari, S., Khan, M. K., Liao, J., Liang, W., "Design and analysis of a chaotic maps-based three-party authenticated key agreement protocol," *Nonlinear Dynamics*, Vol. 80, No. 3, pp. 1209–1220, 2015. Article (CrossRef Link)

[5] Hussain, I., Shah, T., &Gondal, M., "A novel approach for designing substitution-boxes based on nonlinear chaotic algorithm," *Nonlinear Dynamics*, 70, 1791–1794, 2012. Article (CrossRef Link)

[6] Hussain, I., Shah, T., Gondal, M., & Mahmood, H., "An efficient approach for the construction of LFT S-boxes using chaotic logistic map," *Nonlinear Dynamics*, 71, 133–140, 2013. Article (CrossRef Link)

[7] Xiao, D., Liao, X.,& Deng, S., "A novel key agreement protocol based on chaotic maps," *Information Sciences*, 177, 1136–1142, 2007. Article (CrossRef Link)

[8] Özkaynak, F., & Yavuz, S., "Designing chaotic S-boxes based on time-delay chaotic system," *Nonlinear Dynamics*, 2013. Article (CrossRef Link)

[9] Alvarez, G., "Security problemswith a chaos-based deniable authentication scheme," *Chaos Solitons Fractals*, 26, 7–11, 2005. Article (CrossRef Link)

[10] Xiao, D., Liao, X., & Deng, S., "Using time-stamp to improve the security of a chaotic maps-basedkey agreement protocol," *Information Sciences*, 178, 1598–11602, 2008. Article (CrossRef Link)

[11] Han, S., "Security of a key agreement protocol based on chaotic maps," *Chaos Solitons Fractals*,38, 764–768, 2008. Article (CrossRef Link)

[12] Xiang, T., Wong, K., & Liao, X., “On the security of a novel key agreement protocol based on chaotic maps,” *Chaos Solitons Fractals*, 40, 672–675, 2009. Article (CrossRef Link)

[13] He, D. Cryptanalysis of a key agreement protocol based on chaotic Hash.eprint.iacr.org/2011/333.pdf.

[14] Guo, X., & Zhang, J., “Secure group key agreement protocol based on chaotic Hash,” *Information Sciences*, 180, 4069–4074, 2010. Article (CrossRef Link)

[15] Niu, Y., & Wang, X., “An anonymous key agreement protocol based on chaotic maps,” *Communications in Nonlinear Science and Numerical Simulation*, 16(4), 1986–1992, 20111. Article (CrossRef Link)

[16] Li, X., Niu, J., Kumari, S., Liao, J., Liang, W., “An enhancement of a smart card authentication scheme for multi-server architecture,” *Wireless Personal Communications*, Vol. 80, No. 1, pp. 175–192, 2015. Article (CrossRef Link)

[17] Tan, Z., “A chaotic maps-based authenticated key agreement protocol with strong anonymity,” *Nonlinear Dynamics*, 72, 311–320, 2013. Article (CrossRef Link)

[18] Wang, X., & Zhao, J., “An improved key agreement protocol based on chaos,” *Communications in Nonlinear Science and Numerical Simulation*, 15, 4052–4057, 2010. Article (CrossRef Link)

[19] Lee, C., & Hsu, C., “A secure biometric-based remote user authentication with key agreement scheme using extended chaotic maps,” *Nonlinear Dynamics*, 71, 201–211, 2013. Article (CrossRef Link)

[20] Zhao, F., Gong, P., Li, S., Li, M., & Li, P., “Cryptanalysis and improvement of a three-party key agreement protocol using enhanced Chebyshev polynomials,” *Nonlinear Dynamics*, 74, 419–427, 2013. Article (CrossRef Link)

[21] Lai, H., Xiao, J., Li, L., & Yang, Y., “Applying semi-group property of enhanced Chebyshev polynomials to anonymous authentication protocol,” *Mathematical Problems in Engineering*, 2012. Article (CrossRef Link)

[22] Wu, S., Chen, K., Pu, Q., & Zhu, Y., “Cryptanalysis and enhancements of efficient three-party password-based key exchange scheme,” *International Journal of Communication Systems*, 2012. Article (CrossRef Link)

[23] Lee, C., Li, C., & Hsu, C., “A three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps,” *Nonlinear Dynamics*, 73, 125–132, 2013. Article (CrossRef Link)

[24] Yang, J., & Cao, T., “Provably secure three-party password authenticated key exchange protocol in the standard model,” *The Journal of Systems and Software*, 85, 340–350, 2012. Article (CrossRef Link)

[25] Xie, Q., & Zhao, J., “Chaotic maps-based three-party password-authenticated key agreement scheme,” *Nonlinear Dynamics*, 74, 1021–1027, 2013. Article (CrossRef Link)

[26] Lamport, L., “Password authentication with insecure communication,” *Communications of the ACM*, 24(11), 770–772, 1981. Article (CrossRef Link)

[27] Lee, N. Y., & Chiu, Y. C., “Improved remote authentication scheme with smart card,” *Computer Standards & Interfaces*, 27(2), 177–180, 2005. Article (CrossRef Link)

[28] Sun, H. M., “An efficient remote use authentication scheme using smart cards,” *IEEE Transactions on Consumer Electronics*, 46(4), 958–961, 2005. Article (CrossRef Link)

[29] Lin, C. H., & Lai, Y. Y., “A flexible biometrics remote user authentication scheme,” *Computer Standards & Interfaces*, 27(1), 19–23, 2004. Article (CrossRef Link)

[30] Khan, M. K., & Zhang, J., “Improving the security of a flexible biometrics remote user authentication scheme,” *Computer Standards & Interfaces*, 29(1), 82–85, 2007. Article (CrossRef Link)

[31] Li, X., Niu, J., Liao, J., Liang, W., “Cryptanalysis of a dynamic identity based remote user authentication scheme with verifiable password update,” *International Journal of Communication Systems*, Vol. 28, No. 2, pp. 374–382, 2015. Article (CrossRef Link)

[32] Lin, I. C., Hwang, M. S., & Li, L. H., “A new remote user authentication scheme for multi-server architecture,” *Future Generation Computer Systems*, 19(1), 13–22, 2003. Article (CrossRef Link)

[33] Li, L. H., Lin, I. C., & Hwang, M. S., “A remote password authentication scheme for multi-server architecture using neural networks,” *IEEE Transactions on Neural Networks*, 12(6), 1498–1504, 2001. Article (CrossRef Link)

[34] Li, X., Niu, J., Wang, Z., Chen, C., “Applying biometrics to design three-factor remote user authentication scheme with key agreement,” *Security and Communication Networks*, Vol. 7, No. 10, pp. 1488–1497, 2014. Article (CrossRef Link)

[35] Tsai, J. L., “Efficient multi-server authentication scheme based on one-way hash function without verification table,” *Computers & Security*, 27(3–4), 115–121, 2008. Article (CrossRef Link)

[36] Ravi, S. P., Jaidhar, C. D., & Shashikala, T., “Robust smart card authentication scheme for multiserver architecture,” *Wireless Personal Communications*, 72, 729–745, 2013. Article (CrossRef Link)

[37] Zhang, L., “Cryptanalysis of the public key encryption based on multiple chaotic systems,” *Chaos Solitons Fractals*, 37(3), 669–674, 2008. Article (CrossRef Link)

[38] Yoon, E.-J.,&Yoo, K.-Y., “Robust biometrics-based multi-server authenticationwith key agreementscheme for smart cards on elliptic curve cryptosystem,” *The Journal of Supercomputing*, 63, 235–255, 2013. Article (CrossRef Link)

[39] Li, C., Hwang, M., & Chung, Y., “A secure and efficient communication scheme with authenticated key establishment and privacy preserving for vehicular ad hoc networks,” *Computer Communication*, 31,2803–2814, 2008. Article (CrossRef Link)

[40] Kocarev, L., & Lian, S., “Chaos-based cryptography” *Theory, algorithms and applications*, Berlin: Springer, 2011. Article (CrossRef Link)

[41] Hsieh, W., & Leu, J., “Anonymous authentication protocol based on elliptic curve Diffie-Hellman for wireless access networks,” *Wireless Communications and Mobile Computing*, 2012. Article (CrossRef Link)

[42] Zhu, H., “Flexible and Password-Authenticated Key Agreement Scheme Based on Chaotic Maps for Multiple Servers to Server Architecture,” *Wireless Personal Communications*, 82(3), 1697-1718, 2015. Article (CrossRef Link)

[43] Ren, Y. J., Shen, J., Wang, J., Han, J., & Lee, S. Y., “Mutual Verifiable Provable Data Auditing in Public Cloud Storage,” *Journal of Internet Technology*, vol. 16, no. 2, pp. 317-323, 2015.

[44] Guo, P., Wang, J., Geng, X. H., Kim, C. S., & Kim, J. U., “A Variable Threshold-value Authentication Architecture for Wireless Mesh Networks,” *Journal of Internet Technology*, vol. 15, no. 6, pp. 929-936, 2014.

[45] Irshad, A., Sher, M., Ch, S. A., Naqvi, H., & Farash, M. S., “An efficient and anonymous multi-server authenticated key agreement based on chaotic map without engaging Registration Centre,” *The Journal of Supercomputing*, 1-22, 2016. Article (CrossRef Link)

[46] Burrow M, Abadi M, Needham R., “A logic of authentication,” *ACM Transactions on Computer Systems*, 8: 18–36, 1990. Article (CrossRef Link)

[47] Bellare, M., & Rogaway, P., “Random oracles are practical: A paradigm for designing efficient protocols,” in *Proc. of the 1st ACM conference on Computer and communications security*, pp. 62-73, ACM, 1993. Article (CrossRef Link)

[48] ElGamal T, “A public key cryptosystem and signature scheme based on the discrete logarithms,” *IEEE Trans Inform Theory* 31:469–472, 1985. Article (CrossRef Link)

[49] Nguyen, H. T. T., Guizani, M., Jo, M., & Huh, E. N., “An efficient signal-range-based probabilistic key predistribution scheme in a wireless sensor network,” *IEEE Transactions on Vehicular Technology*, 58(5), 2482-2497, 2009. Article (CrossRef Link)

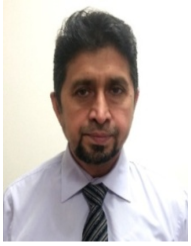
[50] Nguyen, H. T., Jo, M., Nguyen, T. D. & Huh, E. N., "A Beneficial Analysis of Deployment Knowledge for Key Distribution in Wireless Sensor Networks," *Security and Communication Networks*, Vol.5, No.5 pp. 485-495, May 2012. Article (CrossRef Link)

[51] Jiang, Q., Wei, F., Fu, S., Ma, J., Li, G., & Alelaiwi, A., “Robust extended chaotic maps-based three-factor authentication scheme preserving biometric template privacy,” *Nonlinear Dynamics*, 83(4), 2085-2101, 2016. Article (CrossRef Link)

[52] Farash, M. S., and Mahmoud A. A., “An efficient and provably secure three-party password-based authenticated key exchange protocol based on Chebyshev chaotic maps,” *Nonlinear Dynamics* 77.1-2, 399-411, 2014. Article (CrossRef Link)

[53] Islam, S. H., “Design and analysis of a three party password-based authenticated key exchange protocol using extended chaotic maps,” *Information Sciences*, 2015. Article (CrossRef Link)

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. 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.



Hafiz Farooq Ahmad completed his MSc and MPhill in the field of Electronics from Quaid-i-Azam University, Islamabad in 1990 and 1993 respectively. He holds PhD from Tokyo Institute of Technology (Tokyo Japan in Distributed Computing). He is Associate Professor at College of Computer Sciences and Information Technology. Dr. Farooq has been working on semantics systems, health informatics and application security projects. He has been participating in standardization of agent systems at international level through FIPA (Foundation for Intelligent Physical Agents). He contributed in agent cites project, a European funded research and development project for agent systems. Dr. Farooq initiated SAGE (Scalable fault tolerant Agent Grooming Environment) project and proposed the concept of developing decentralized Multi agent system SAGE back in 2002. He has more than 100 international publications including a book on security in sensors. He has been awarded a number of national and international awards such as PSF/COMSTECH best researcher of the year 2005 and Star Laureate awards 2004. In recognition of his research excellence, he was awarded the Best Researcher Award of the year 2011 by NUST.

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.

Muhammad Sher is a Professor having more than 120 scientific publications. He is chairman of the Department of Computer Science & Software Engineering, International Islamic University. He is also Dean of the Faculty of Basic & 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.

Shehzad Ashraf Chaudhry received distinction in his Masters and PhD from International Islamic University Islamabad, Pakistan in 2009 and 2016 respectively. He was awarded Gold Medal for achieving 4.0/4.0 CGPA in his Masters. Currently, he is working as an Assistant Professor at the Department of Computer Science & Software Engineering, International Islamic University, Islamabad. He authored more than 45 scientific publications appeared in different international journals and proceedings including 31 in SCI/E journals. His research interests include Lightweight Cryptography, Elliptic/Hyper Elliptic Curve Cryptography, Multimedia Security, E- Payment systems, MANETs, SIP authentication, Smart Grid Security, IP Multimedia sub-system and Next Generation Networks.