ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume 24, Number 1, 2021, 58–78

An Improved User Authentication Protocol Based on Chaotic Maps for Multiserver Environment

Devender KUMAR[1], Satish CHAND[2], and Bijendra KUMAR[3]

[1]Department of Information Technology, NSUT, New Delhi, India
[2]School of Computer and System Sciences, JNU, New Delhi, India
[3]Department of Computer Science and Engineering, NSUT New Delhi, India
devender.kumar@nsit.ac.in

Abstract. Now-a-days researchers have focused on designing authentication schemes using smart card because of advancement in computer and communication technologies. Recently, Li et al. have discussed a protocol for multiserver environment based on chaotic maps. We analyze its security and find that it is not resistant to the privileged insider and smart card loss attacks besides user anonymity. Here, we propose an improved user authentication protocol for multiserver environment using the chaotic maps to overcome these limitations. We formally prove using BAN logic that our protocol provides mutual authentication. We show its formal security analysis using random oracle and discuss its informal security analysis to show that it is resistance to the various known attacks. We simulate our protocol using AVISPA tool to verify its security against active and passive attacks. Our protocol is more secure than the existing protocols.

Key-words: User authentication, smart card; session key agreement; multiserver environment; chaotic maps

1. Introduction

Due to advancements in internet technologies, a person can access various services and resources online. However, to provide the security to the information sent over a public network is a challenging issue. The user authentication with key agreement is a kind of basic security protocol, which authenticates a user before accessing the remote services. Moreover, it ensures secret communication between a server and a user using a session key shared between them.

In 1981, the first user authentication was discussed by Lamport [1]. However, it requires a password table to authenticate a user which can be modified by an attacker. Thereafter, many schemes have been discussed without using a password table [2, 3]. Many user authentication schemes have been presented using smart card [4–7]. A smart card has some inherent properties

such as easy to take, fast computation, storage capability, etc. The earlier authentication protocols were developed for the single server environment. Because of the rise in user's number and the types of services, the single server environments were replaced by several servers located at different places, which constitutes a multiserver environment. The protocols meant for a single environment do not meet the requirements of a multiserver environment. Normally, a user is required to remember multiple passwords to access the services from the different servers, which is not possible for a user to remember distinct passwords for distinct servers.

In 2001, Li et al. developed the first multiserver authentication protocol using neural networks [8]. Lin et al. [9] showed that the protocol [8] was inefficient due to huge communication and computation costs required to train the neural network. Based on ElGamal digital signature, they designed an authentication protocol for multiserver environment. This protocol however does not offer mutual authentication and a user is required to store many system variables for each server. Juang [10] overcame these problems by proposing a light weight multiserver authentication protocol using symmetric cryptography and hash function. In the paper [11], it was found that the scheme [10] is inefficient and discussed an efficient and secure scheme. In 2005, Tsaur et al. [12] developed an authentication protocol for multiserver environment by using Lagrange interpolating polynomial, which requires high computation cost.

In 2008, Tsai [13] discussed a multiserver authentication protocol using hash function, claiming it to be resistant against various attacks like server spoofing, stolen-verifier and replay attacks, etc. Chen et al. [14] and Wang et al. [15] however independently found that the protocol [13] is not resistant to the server spoofing attack. Also, in the paper [16], the authors found the man-in-the-middle attack in the scheme [13]. In 2009, based on dynamic identity, a user authentication protocol for multiserver environment was discussed in the paper [17]. Hsiang and Shih [18] found that the protocol [17] is not resistant to various attacks and overcame these problems in their protocol. The protocol [18] was not secured as pointed out in the Sood et al. [19] and they overcame these limitations in their protocol. In 2012, Li et al. [20] found that stolen smart card, impersonation and leak-of-verifier attacks exist in the protocol [19] and they overcame these weaknesses in their protocol.

In the last decades, many user authentication and key agreement protocols based on chaotic maps have been discussed [21–25]. In 2014, Lee et al. [21] discussed an authentication scheme for multiserver based on extended chaotic maps, claiming it to be resistant to many attacks. But, Li et al. [25] found that the scheme [21] is not resistant to server and registration center spoofing attacks, besides inefficient login phase and absence of password change phase. They discussed a protocol by overcoming these weaknesses for multiserver using chaotic maps. In this paper, we cryptanalyze the Li et al.'s scheme [25] and find that it lacks user anonymity to an insider and is not resistant to privileged insider and smart card loss attacks. In this paper, we propose an improved protocol by overcoming its weaknesses.

1.1. Our contribution

Our contribution is described as follows:

1. We analyze the security of the protocol [25] and find the smart card loss attack, privileged attack and user anonymity problem in it.

2. We propose an improved authentication protocol for multiserver environment to overcome the weaknesses of the protocol [25].

3. We use BAN logic to show that our protocol provides mutual authentication.
4. We discuss the formal security analysis of our scheme using the random oracle model to show that the user identity and password, secret key of the registration center and session key are safe. We also analyze its security informally to show that it is invulnerable to many known attacks.
5. We use the AVISPA (Automated Validation of Internet Security Protocols and Applications) tool for the formal security verification of our protocol and show that the active and passive attacks are not possible in it.
6. We compare our protocol with the related protocols [21, 25] in terms of security features and computational cost and show that our protocol offers more security features than the related protocols.

1.2. Threat model

Here, we present a threat model under which the security of the Li et al.'s scheme [25] and our scheme are analyzed. The following assumptions have been made about the attacker's capabilities:

1. An attacker E can eavesdrop all communicated messages between the participants over a public channel.
2. E can delete, resend, modify, reroute, and insert the eavesdropped messages.
3. E can extract all the information stored in the smart card of a valid user [26, 27] if E somehow gets or steals it.
4. E cannot know the user's password as well as steal the user's smart card at the same time.
5. E can enumerate offline all possible elements in the cartesian product $D_{id} \times D_{pw}$ in a reasonable amount of time [28], where D_{id} and D_{pw} are identity space and password space, respectively.
6. E cannot trap and update any messages over a secure channel.
7. An insider/administrator cab be malicious.

1.3. Organization of the paper

The arrangement of this paper is as follows. In section 2., we give some preliminaries. In sections 3. and 4., we review and cryptanalyze the Li et al.'s protocol [25], respectively. Our proposed protocol is presented in section 5.. Section 6. presents its formal proof of mutual authentication using BAN logic. We present its formal and informal security analysis in section 7.. Its security verification using AVISPA tool is given in section 8.. Its comparison with the related protocols [21, 25] is given in section 9.. Finally, the conclusion is given in section 10..

2. Preliminaries

Here, we present the brief introduction about the Chebyshev chaotic maps.

2.1. Chebyshev chaotic maps

According to [29], the Chebyshev polynomial $T_n(z) : [-1, 1] \rightarrow [-1, 1]$ is a polynomial in z of degree n, where n is an integer, and $z \in [-1, 1]$ and it is defined as below:
$T_n(z) = cos(n.arccos(z))$.
Its recurrence relation is given as below:
$T_n(z) = 2zT_{n-1}(z) - T_{n-2}(z), n \geq 2$, where $T_0(z) = 1$ and $T_1(z) = z$.
Here, cos(z) and arccos(z) are trigonometric functions [30].
Some examples of the Chebyshev polynomials are given as below:

$$\begin{aligned} T_2(z) &= 2z^2 - 1, \\ T_3(z) &= 4z^3 - 3z, \\ T_4(z) &= 8z^4 - 8z^2 + 1, \\ T_5(z) &= 16z^5 - 20z^3 + 5z. \end{aligned}$$

Chebyshev polynomials $T_n(z) : [-1, 1] \rightarrow [-1, 1]$ satisfy the following two properties:

1. Semi-group property: This property is given below:

$$\begin{aligned} T_r(T_s(z)) &= cos(rcos^{-1}(cos(scos^{-1}(z)))) \\ &= cos(rscos^{-1}(z)) \\ &= T_{rs}(z) = T_{sr}(z) \\ &= cos(srcos^{-1}(z)) \\ &= cos(scos^{-1}(cos(rcos^{-1}(z)))) \\ &= T_s(T_r(z)) \end{aligned}$$

   where r and s are positive integers and $z \in [-1, 1]$.

2. Chaotic property:
   The Chebyshev polynomial map $T_n(z) : [-1, 1] \rightarrow [-1, 1]$ of degree $n > 1$ is a chaotic map whose invariant density is given by
   $f^*(z) = \frac{1}{(\pi\sqrt{1-z^2})}$, for Lyapunov exponent ln(n)>0.

Zhang [31] enhanced the semi-group property of the Chebyshev polynomials defined on the extended interval $(-\infty, +\infty)$ as follows:
$T_n(z) = (2zT_{n-1}(z) - T_{n-2}(z))$ mod p, where $n \geq 2, z \in (-\infty, +\infty)$, and p is a large prime number. Apparently, $T_r(T_s(z)) \equiv T_{sr}(z) \equiv T_s(T_r(z))$ mod p.
So, the semigroup property is also satisfied in the enhanced Chebyshev polynomials.

The two computational problems [32] in the Chebyshev polynomials which are not feasible to solve in polynomial time are given below:

1. Given y and z such that $T_r(z) = y$, to compute the integer r is called the discrete logarithm problem (DLP).

2. Given z, $T_r(z)$, and $T_s(z)$, to compute $T_{rs}(z)$ is called the Diffie-Hellman problem (DHP).

3. Review of Li et al.'s protocol

Here, we revisit the Li et al.'s protocol [25] in which there are three parties: RC, U_i and S_j. To initialize the system, RC chooses X and w, a random number and a master key, respectively. Then, it calculates the public key $R \equiv T_w(X)$ mod p. S_j sends its unique identity SID_j to RC for its registration through a secure channel. On obtaining the request, RC calculates $K_j = h(SID_j||w)$ and sends $\{K_j, X\}$ to S_j through a private channel. The symbols used in this paper are given in Table 1. The protocol [25] includes the below three phases:

Table 1. Notations used in paper.

Notations	Description
U_i	i^{th} user
RC	Registration center
S_j	j^{th} server
ID_i	U_i's identity
SID_j	S_j's identity
PW_i	U_i's password
X	Random number selected by RC
R	Public key of RC
E	An attacker
r_i	Random number selected by U_i
p	A large prime number
w	Master secret key of RC
r_j	Random number selected by S_j
SK	Session key between U_i and S_j
$\oplus$	Exclusive-OR operator
SC	Smart card
r_k	Random number selected by RC
$\|\|$	Concatenation operator
h(.)	A one-way hash function

3.1. Registration phase

The below steps are executed to register U_i with RC to get the services from servers $\{S_1, S_2, ..., S_r\}$:

1. U_i selects ID_i, PW_i and N_i as his identity, password and a random number, respectively. He calculates $e_i = h(PW_i||N_i)$ and transmits the message $\{ID_i, e_i\}$ to RC for his registration through a private channel.

2. On getting the request for registration from U_i, RC computes $f_i = h(ID_i||w)$, $A_i = h(ID_i||e_i)$, $B_i = e_i \oplus f_i$ and stores the information $\{A_i, B_i, X, R, h(.), p\}$ into SC. Then, RC transmits the SC to U_i through a secure channel.

3. After obtaining the SC from RC, U_i stores N_i in the SC. Finally, the information $\{A_i, B_i, N_i, X, R, h(.), p\}$ is contained into SC.

3.2. Login and authentication phase

This phase is same as discussed in [25].

3.3. Password change

This phase is same as discussed in [25].

4. Cryptanalysis of Li et al.'s protocol

Here, we cryptanalyse the scheme [25] as follows under the threat model given in the section 1.2.:

4.1. User anonymity

Since user U_i sends the message $\{ID_i, e_i\}$, where $e_i = h(PW_i||N_i)$, to RC in step (1) of registration phase of the scheme [25] in which his identity ID_i is transmitted in plaintext. Thus, his identity is not anonymous from a malicious privileged insider/administrator.

4.2. Smart card loss attack

Assume that an attacker E has acquired the smart card of U_i and extracts the information $\{A_i, B_i, N_i, X, R, h(.)\ , p\}$ stored in his smart card [26, 27], where $A_i = h(ID_i||\ e_i)$, $e_i = h(PW_i||N_i)$, $B_i = e_i \oplus f_i$ and $f_i = h(ID_i||w)$. Then, he can offline guess the identity and password in the following manner [28]:

1. E guesses ID_i^* and PW_i^* as an identity and password of U_i respectively.

2. E computes the following:
   $e_i{}^* = h(PW_i^*||N_i)$
   $A_i{}^* = h(ID_i^*||e_i^*)$

E checks if $A_i{}^* = A_i$. If it is true, then he has guessed the correct values of identity and password. Otherwise, he repeats the steps (1) and (2) until $A_i{}^* = A_i$.

4.3. Privileged insider attack

U_i transmits $\{ID_i, e_i\}$ to RC in registration phase, where $e_i = h(PW_i||N_i)$. Suppose that a privileged insider obtains the SC of U_i [33] and extracts all the stored information $\{A_i, B_i, N_i, X, R, h(.), p\}$ from it [26, 27]. Then, he can guess the password PW_i of the user U_i using the below steps:

1. Select a password PW_i^* and compute $e_i = h(PW_i^*||\ N_i)$

2. Check if $e_i^* = e_i$. If it is true, then the privileged insider gets the correct password PW_i of U_i and stops the procedure. Otherwise, repeat the steps (1) and (2).

Hence, the privileged insider attack exists in their scheme.

5. Proposed protocol

Here, we design an improved protocol to overcome the limitations of the protocol [25]. Our proposed protocol includes three parties: user U_i, the registration center RC and server S_j. It includes the five phases as given below:

5.1. Initialization phase

RC chooses a random number X, and a master key w of size 1024 bits to initialize the system. Then, it calculates the public key $R \equiv T_w(X)$ mod p. S_j sends its unique identity SID_j to RC for its registration via a secure channel. On obtaining the request, RC calculates $K_j = h(SID_j||w)$ and sends $\{K_j, X\}$ to S_j through a secure channel.

5.2. Registration phase

The below steps are executed to register U_i with RC to get the services from servers $\{S_1, S_2, ..., S_r\}$:

1. U_i selects ID_i, PW_i and N_i as his identity, password and a nonce, respectively. He computes $C_i = h(ID_i||N_i)$ and $P_i = h(PW_i||N_i)$. Then, he transmits the registration request $\{C_i, P_i\}$ to RC via a secure channel.

2. After getting the information $\{C_i, P_i\}$ from U_i, RC generates a random number v_i and an integer n such that $2^4 \leq n \leq 2^8$ as the parameter of fuzzy verifier corresponding to user U_i and computes $B_i = h(v_i||w) \oplus h(C_i||P_i)$ and $D_1 = v_i \oplus h(w)$. Then, RC stores the information $\{B_i, D_1, X, R, n, h(.), p\}$ into a SC and transmits the SC to U_i through a secure channel.

3. After obtaining the SC from RC, U_i computes $Z_i = h(PW_i||ID_i) mod\ n \oplus N_i$ and $D_2 = h(C_i||N_i||P_i) mod\ n$, and stores Z_i and D_2 into the SC. Finally, the information $\{B_i, D_1, X, R, n, Z_i, D_2, h(.), p\}$ is contained into his SC.

5.3. Login phase

To log U_i into S_j, the below steps are executed by U_i:

1. U_i chooses SID_j, an identity of S_j on which he wants to login. He puts his SC into the card reader and inputs his identity ID_i and password PW_i. Then, the smart card computes the following:
   $N_i{}^* = Z_i \oplus h(PW_i||ID_i)\ mod\ n$,
   $C_i{}^* = h(ID_i||N_i{}^*)$,
   $P_i{}^* = h(PW_i||N_i{}^*)$,
   $D_2{}^* = h(C_i{}^*||N_i{}^*||P_i{}^*)\ mod\ n$
   The smart card checks if $D_2{}^* = D_2$. If true, then proceed to the next step; otherwise, terminate the session.

2. SC creates a random number r_i and calculates as follows:
   $C_1 = T_{r_i}(X)$ mod p,
   $C_2 = T_{r_i}(R)$ mod p,

$UID_i = D_1 \oplus h(C_1||C_2)$,
$f_i = B_i \oplus h(C_i^*||P_i^*)$,
$M_{ik} = h(SID_j||f_i||C_1||C_2)$
U_i sends the login message $\{UID_i, C_1, M_{ik}\}$ to S_j over a public channel.

5.4. Authentication phase

Below steps are performed by U_i, S_j and RC to authenticate each other:

1. After obtaining the message $\{UID_i, C_1, M_{ik}\}$, S_j creates a random number r_j and computes the following:
   $C_3 = T_{r_j}(X)$ mod p,
   $M_{jk} = h(K_j||C_3)$
   S_j sends the message $\{UID_i, C_1, M_{ik}, SID_j, C_3, M_{jk}\}$ to RC over a public channel.

2. After getting the message $\{UID_i, C_1, M_{ik}, SID_j, C_3, M_{jk}\}$ from S_j, RC calculates $K'_j = h(SID_j||w)$, $M'_{jk} = h(K'_j||C_3)$ and checks if $M'_{jk} = M_{jk}$. If not equal, then reject the session. Otherwise, RC authenticates S_j and calculates the following:
   $C'_2 = T_w(C_1)$ mod p,
   $v'_i = UID_i \oplus h(w) \oplus h(C_1||C'_2)$,
   $f'_i = h(v'_i||w)$,
   $M'_{ik} = h(SID_j||f'_i||C_1||C'_2)$
   RC checks if $M'_{ik} = M_{ik}$. If not equal, then reject the session. Otherwise, RC authenticates U_i. RC chooses a random number r_k and computes the following:
   $D_i = K'_j \oplus r_k$,
   $M_{kj} = h(K'_j||C_1||C_3||r_k)$
   $E_i = h(K'_j||r_k)$,
   $G_i = E_i \oplus f'_i \oplus C'_2$,
   $M_{ki} = h(f'_i||G_i||SID_j||C'_2||C_3)$,
   RC sends the response message $\{D_i, M_{kj}, G_i, M_{ki}\}$ to S_j over a public channel.

3. On receiving the message $\{D_i, M_{kj}, G_i, M_{ki}\}$ from RC, S_j computes the following:
   $r'_k = K_j \oplus D_i$,
   $M'_{kj} = h(K_j||C_1||C_3||r'_k)$
   S_j checks if $M'_{kj} = M_{kj}$. If it is not true, then the session is terminated. Otherwise, RC is authenticated by S_j and S_j calculates the following:
   $E'_i = h(K_j||r'_k)$,
   $SK = T_{r_j}(C_1)$ mod p,
   $M_{ji} = h(SID_j||C_1||C_3||E'_i||SK)$
   S_j sends the message $\{G_i, M_{ki}, C_3, M_{ji}\}$ to U_i over a public channel.

4. On obtaining the message $\{G_i, M_{ki}, C_3, M_{ji}\}$ from S_j, the SC computes $M'_{ki} = h(f_i||G_i||SID_j||C_2||C_3)$ and checks if $M'_{ki} = M_{ki}$. If it is not true, then the session is terminated. Otherwise, U_i authenticates RC. Then, U_i calculates the following:
   $E''_i = G_i \oplus f_i \oplus C_2$,
   $SK' = T_{r_i}(C_3)$ mod p,
   $M'_{ji} = h(SID_j||C_1||C_3||E''_i||SK')$
   U_i checks if $M'_{ji} = M_{ji}$. If it is not true, then the session is terminated. Otherwise, U_i

authenticates S_j. Finally, U_i calculates $M_{ij} = h(C_3||E_i''||SK')$ and transmits it to S_j over an insecure channel.

5. On receiving the message $\{M_{ij}\}$ from U_i, S_j computes $M'_{ij} = h(C_3||E'_i||SK)$ and checks if $M'_{ij} = M_{ij}$. If it is not right, then reject the session. Otherwise, U_i is authenticated by S_j.

Finally, A session key $SK \equiv T_{r_j}(C_1)$ mod p $\equiv T_{r_i r_j}(X)$ mod p $\equiv T_{r_i}(C_3)$ mod p = SK' is shared between U_i and S_j to communicate.

5.5. Password change

Following steps are performed whenever U_i wishes to alter his password:

1. U_i puts his SC into a card reader and then enters his ID_i, PW_i and requests to change the password.

2. Then, SC computes as follows:
   $N_i{}^* = Z_i \oplus h(PW_i||ID_i)\, mod\, n$,
   $C_i{}^* = h(ID_i||N_i{}^*)$,
   $P_i{}^* = h(PW_i||N_i{}^*)$,
   ${D_2}^* = h(C_i{}^*||N_i{}^*||P_i{}^*)\, mod\, n$
   The SC checks if ${D_2}^* = D_2$. If it is not true, then his password change request is rejected. Otherwise, go to next step.

3. U_i is prompted for a new password ${PW_i}^{new}$ and creates a nonce ${N_i}^{new}$. Smart card computes the following:
   ${C_i}^{new} = h(ID_i||{N_i}^{new})$,
   ${P_i}^{new} = h({PW_i}^{new}||{N_i}^{new})$,
   ${D_2}^{new} = h({C_i}^{new}||{N_i}^{new}||{P_i}^{new})\, mod\, n$,
   ${Z_i}^{new} = h({PW_i}^{new}||ID_i)\, mod\, n \oplus {N_i}^{new}$
   Smart card replaces D_2, Z_i with ${D_2}^{new}, {Z_i}^{new}$, respectively.

6. Mutual authentication using BAN-logic

Here, we formally prove using BAN-logic [34] that our protocol provides mutual authentication. We show that U_i and S_j share a session key between them. The notations and main postulates of BAN logic can be found in [34] and [25].

Our protocol will satisfy the goals as given below:

- $Goal1 : U_i| \equiv \left(U_i \overset{\text{SK}}{\longleftrightarrow} S_j\right)$
- $Goal2 : U_i| \equiv S_j| \equiv \left(U_i \overset{\text{SK}}{\longleftrightarrow} S_j\right)$
- $Goal3 : S_j| \equiv \left(U_i \overset{\text{SK}}{\longleftrightarrow} S_j\right)$
- $Goal4 : S_j| \equiv U_i| \equiv \left(U_i \overset{\text{SK}}{\longleftrightarrow} S_j\right)$

The idealized form of our protocol is given below:

- $Message1 : U_i \rightarrow RC : \left(ID_i, SID_j, U_i \overset{\{X\}_{w,r_i}}{\longleftrightarrow} RC \right)_{h(v_i||w)}$
- $Message2 : S_j \rightarrow RC : \left(S_j \overset{h(SID_j||w)}{\longleftrightarrow} RC \right)_{h(SID_j||w)}$
- $Message3 : RC \rightarrow S_j : \left(C_1, C_3, S_j \overset{r_k}{\longleftrightarrow} RC \right)_{h(SID_j||w)}$
- $Message4 : RC \rightarrow U_i : \left(SID_j, C_2, C_3, U_i \overset{\{X\}_{w,r_i}}{\longleftrightarrow} RC \right)_{h(v_i||w)}$
- $Message5 : S_j \rightarrow U_i : \left(SID_j, C_1, U_i \overset{SK}{\longleftrightarrow} S_j \right)_{h(K_j||r_k)}$
- $Message6 : U_i \rightarrow S_j : \left(C_3, U_i \overset{SK}{\longleftrightarrow} S_j \right)_{h(K_j||r_k)}$

Some initial assumptions to analyze our proposed scheme are given below:

- $A_1 : U_i| \equiv \#(r_i)$
- $A_2 : S_j| \equiv \#(r_j)$
- $A_3 : RC| \equiv \#(r_k)$
- $A_4 : U_i| \equiv \#(C_1)$
- $A_5 : S_j| \equiv \#(C_3)$
- $A_6 : U_i| \equiv U_i \overset{h(v_i||w)}{\longleftrightarrow} RC$
- $A_7 : RC| \equiv U_i \overset{h(v_i||w)}{\longleftrightarrow} RC$
- $A_8 : S_j| \equiv S_j \overset{h(SID_j||w)}{\longleftrightarrow} RC$
- $A_9 : RC| \equiv S_j \overset{h(SID_j||w)}{\longleftrightarrow} RC$
- $A_{10} : S_j| \equiv U_i \overset{h(K_j||r_k)}{\longleftrightarrow} S_j$
- $A_{11} : U_i| \equiv U_i \overset{h(K_j||r_k)}{\longleftrightarrow} S_j$
- $A_{12} : U_i| \equiv S_j \Rightarrow (U_i \overset{SK}{\longleftrightarrow} S_j)$
- $A_{13} : S_j| \equiv U_i \Rightarrow (U_i \overset{SK}{\longleftrightarrow} S_j)$

Now, we show that U_i and S_j share SK, a session key, for secure communication using initial assumptions and BAN-logic rules.

- From the $Message5$, we get

$$U_i \triangleleft \left(SID_j, C_1, U_i \overset{\text{SK}}{\longleftrightarrow} S_j\right)_{h(K_j||r_k)} \tag{1}$$

- From (1), A_{11} and apply the message meaning rule, we have

$$U_i| \equiv S_j| \sim \left(SID_j, C_1, U_i \overset{\text{SK}}{\longleftrightarrow} S_j\right) \tag{2}$$

- From (2), A_4 and apply the freshness-conjuncatenation rule, we have

$$U_i| \equiv \#\left(SID_j, C_1, , U_i \overset{\text{SK}}{\longleftrightarrow} S_j\right) \tag{3}$$

- Applying the nonce verification rule on (2), (3), we have

$$U_i| \equiv S_j| \equiv \left(SID_j, C_1, , U_i \overset{\text{SK}}{\longleftrightarrow} S_j\right) \tag{4}$$

- From (4) and apply the believe rule, we have

$$U_i| \equiv S_j| \equiv (U_i \overset{\text{SK}}{\longleftrightarrow} S_j) \qquad Goal2 \tag{5}$$

- From (5), A_{12} and apply jurisdiction rule, we have

$$U_i| \equiv (U_i \overset{\text{SK}}{\longleftrightarrow} S_j) \qquad Goal1 \tag{6}$$

- It is clear from $Message6$ that

$$S_j \triangleleft \left(C_3, U_i \overset{\text{SK}}{\longleftrightarrow} S_j\right)_{h(K_j||r_k)} \tag{7}$$

- From (7), A_{10} and using the message meaning rule, we have

$$S_j| \equiv U_i \sim \left(C_3, U_i \overset{\text{SK}}{\longleftrightarrow} S_j\right) \tag{8}$$

- From (8), A_5 and using the freshness-conjuncatenation rule, we have

$$S_j| \equiv \#\left(C_3, U_i \overset{\text{SK}}{\longleftrightarrow} S_j\right) \tag{9}$$

- After applying the nonce-verification rule on (8), (9), we get

$$S_j| \equiv U_i \equiv \left(C_3, U_i \overset{\text{SK}}{\longleftrightarrow} S_j\right) \tag{10}$$

- From (10) and apply the believe rule, we have

$$S_j| \equiv U_i \equiv \left(U_i \overset{\text{SK}}{\longleftrightarrow} S_j\right) \qquad Goal4 \qquad (11)$$

- From (11), A_{13} and apply the jurisdiction rule, we have

$$S_j| \equiv \left(U_i \overset{\text{SK}}{\longleftrightarrow} S_j\right) \qquad Goal3 \qquad (12)$$

The above discussion clearly shows that our protocol offers the mutual authentication between U_i and S_j and shares a session key SK.

7. Security analysis

In this section, we present the formal and informal security analysis of our scheme.

7.1. Formal security analysis

Here we present the formal security analysis of our protocol to prove that E cannot retrieve the secret variables ID_i, PW_i, w and SK. For its formal security analysis, we use the same method as given in the protocols [35–37]. The random oracles are defined as below:
Reveal1: It will give an output x without any restriction from a comparable hash output y, where $y = f(x)$.
Reveal2: Given x and y, where $x, y \in (-\infty, +\infty)$ such that $T_r(x)\ mod\ p = y$, this oracle will output the integer r without any condition.

Theorem 1. *Assume that the hash function $h(.)$ behaves like a random oracle and the SC of U_i is lost or stolen by E. Also, E have the transmitted messages $\{UID_i, C_1, M_{ik}, SID_j, C_3, M_{jk}\}$. Still our protocol is secure against E for obtaining the ID_i and PW_i of U_i, and the secret key w of the RC.*

Proof. Let us assume that E receives ID_i and PW_i of U_i, and secret key w of the RC from the variables $\{B_i, D_1, X, R, n, Z_i, D_2, h(.), p\}$ stored in the SC of U_i, the experimental algorithm $Exp1^{HASH}_{E,IUAP}$ and the transmitted messages. The success probability for $Exp1^{HASH}_{E,IUAP}$ is given as follows:

$$succ1 = Pr[Exp1^{HASH}_{E,IUAP} = 1] - 1,$$

where $Pr[Z]$ is probability of event Z.

The advantage function for this experiment is given by $Adv(t_1, q_{r1}) = Max_E\{succ1\}$, where the maximum depends on all eveasdroppers with execution time t_1 and number of queries q_{r1} sent to Reveal1 oracles. Our protocol is safe against E to receive ID_i, PW_i and w if, $Adv(t_1, q_{r1}) \leq \epsilon$, for small $\epsilon > 0$. From the experimental algorithm $Exp1^{HASH}_{E,IUAP}$, we see that if E has the ability to find the inverse of the hash function, then only E can obtain ID_i, PW_i and w and wins the game. But, it is not feasible to find the inverse of the hash function in polynomial time, i.e., $Adv(t_1, q_{r1}) \leq \epsilon$ for any sufficiently small $\epsilon > 0$. Hence, our scheme is secured against E to get ID_i and PW_i of U_i, and the secret key w of the RC. □

Algorithm 1 $Exp1_{E,IUAP}^{HASH}$

Input: $\{B_i, D_1, X, R, n, Z_i, D_2, h(.), p, UID_i, C_1, M_{ik}, SID_j\}$
Result: 1 or 0.
Call Reveal1 oracle on M_{ik} to get information SID_j, f_i, C_1 and C_2) as $(SID_j^{'}||f_i^{'}||C_1^{'}||C_2^{'}) \leftarrow Reveal1(M_{ik})$
if $((SID_j^{'} = SID_j)$ & $(C_1^{'} = C_1))$ **then**
    Compute $A = h(C_i||P_i) = B_i \oplus f_i$
    Call Reveal1 oracle on A to get information C_i and P_i as $(C_i^{'}||P_i^{'}) \leftarrow Reveal1(A)$
    Call Reveal1 oracle on $C_i^{'}$ to get information ID_i and N_i as $(ID_i^{'}||N_i^{'}) \leftarrow Reveal1(C_i^{'})$
    Call Reveal1 oracle on $P_i^{'}$ to get information PW_i and N_i as $(PW_i^{'}||N_i^{''}) \leftarrow Reveal1(P_i^{'})$
    if $(N_i^{'} = N_i^{''})$ **then**
        Compute $N_i{}^* = Z_i \oplus h(PW_i^{'}||ID_i^{'})\, mod\, n$
        if $(N_i^{'} = N_i^*)$ **then**
            Compute $D_2{}^* = h(C_i{}^{'}||N_i{}^{'}||P_i{}^{'})\, mod\, n$
            if $(D_2 = D_2^*)$ **then**
                Accept $ID_i^{'}$ and $PW_i^{'}$ as correct identity and password of U_i
                Call Reveal1 oracle on $f_i^{'}$ to get information v_i and w as $(v_i^{'}||w^{'}) \leftarrow Reveal1(f_i^{'})$
                if $(D_1 = v_i^{'} \oplus h(w^{'}))$ **then**
                    Accept $w^{'}$ as the correct secret key of RC
                    return (1) Success
                else
                    return (0) Failure
            else
                return (0) Failure
        else
            return (0) Failure
    else
        return (0) Failure
else
    return (0) Failure

Theorem 2. *Suppose that the extended chebyshev chaotic discrete logarithm problem (CHDLP) and hash function behave as random oracles, and E gets access to the SC of U_i, and intercepts the transmitted messages $\{UID_i, C_1, M_{ik}, SID_j, C_3, M_{jk}\}$. Still E cannot calculate SK .*

Proof. Here we construct an attacker E that will have the capability to compute the session key SK. E uses the random oracles Reveal1 and Reveal2 to run the experimental algorithm $Exp2_{E,IUAP}^{HASH,CHDLP}$ for our scheme. Define the success probability $succ2$ for $Exp2_{E,IUAP}^{HASH,CHDLP}$ as below:

$$succ2 = 2Pr[Exp2_{E,IUAP}^{HASH,CHDLP} = 1] - 1,$$

where Pr[Z] is probability of event Z.

The advantage function for this experiment is defined as $Adv2(t_4, q_{r4}, q_{r5}) = Max_E\{succ2\}$, where the maximum depends on all the attackers with execution time t_4, and number of queries q_{r4} and q_{r5} sent to Reveal1 and Reveal2 oracles, respectively. Our protocol is secure against E to calculate SK if, $Adv2(t_4,\ q_{r4}, q_{r5}) \leq \epsilon$, for small $\epsilon > 0$. From the experimental algorithm $Exp2_{E,IUAP}^{HASH,CHDLP}$, it is seen that if E can find the inverse of the hash function and solves the CHDLP, then only E can calculate SK and wins the game. But, it is not possible to find the inverse of the hash function in polynomial time, i.e., $Adv_E^{HASH}(t_5) \leq \epsilon_1$ for any sufficiently small $\epsilon_1 > 0$. Also, it is not possible to solve the CHDLP in polynomial time, i.e., $Adv_E^{CHDLP}(t_6) \leq \epsilon_2$, for any sufficiently small $\epsilon_2 > 0$. Thus, $Adv2(t_4, q_{r4}, q_{r5}) \leq \epsilon$, for small $\epsilon > 0$, as it depends on two advantages $Adv_E^{HASH}(t_5)$ and $Adv_E^{CHDLP}(t_6)$. □

Algorithm 2 $Exp2_{E,IUAP}^{HASH,CHDLP}$

1: Input: $\{UID_i, C_1, M_{ik}, SID_j, C_3, M_{jk}\}$
2: Result: 1 or 0.
3: Call Reveal2 oracle on C_3 to get information r_j as $(r_j^{'}) \leftarrow Reveal2(C_3)$
4: Compute $SK = T_{r_j^{'}}(C_1) mod\ p$
5: Call Reveal1 oracle on M_{ji} to get information SID_j, C_1, C_3, $E_i^{'}$ and SK as $(SID_j^{'}||C_1^{'}||C_3^{'}||E_i^{''}||SK^{'}) \leftarrow Reveal1(M_{ji})$
6: **if** $((SID_j^{'} = SID_j)\ \&\ (C_1^{'} = C_1)\ \&\ (C_3^{'} = C_3)\ \&\ (SK^{'} = SK))$ **then**
7: Accept SK as the correct value of session key
8: **return** (1) Success
9: **else**
10: **return** (0) Failure

7.2. Informal security analysis

Here, we show informally that our scheme is resistant against the known attacks under the threat model given in section 1.2..

7.2.1. User anonymity

Assume that E has intercepted the login message $\{UID_i, C_1, M_{ik}\}$ of U_i, where $UID_i = D_1 \oplus h(C_1||C_2)$, $C_1 = T_{r_i}(X)$ mod p, $C_2 = T_{r_i}(R)$ mod p, $M_{ik} = h(SID_j||f_i||\ C_1||C_2)$

and $f_i = B_i \oplus h(C_i^*||P_i^*)$. Any of these parameters do not contain the identity ID_i of U_i directly and user identity is kept safe by one-way hash function. Now assume that E gets the SC of U_i and extracts all the stored information $\{B_i, D_1, X, R, n, Z_i, D_2, h(.), p\}$ from it [26, 27], where $B_i = h(v_i||w) \oplus h(C_i||P_i)$, $D_1 = v_i \oplus h(w)$, $Z_i = h(PW_i||ID_i)\ mod\ n \oplus N_i$ and $D_2 = h(C_i||N_i||P_i)\ mod\ n$. Also, in this case, user identity ID_i is not contained directly and is protected by one-way hash function. Moreover, U_i is anonymous from malicious insider in our protocol as he does not send his identity ID_i to RC in plaintext in the registration phase. Thus, our protocol provides user anonymity.

7.2.2. Replay attack

For each session, U_i, S_j and RC generate distinct nonces r_i, r_j and r_k, respectively, in our protocol. The messages sent by U_i, S_j and RC in each session are dependent on these three nonces, respectively. Hence, an attacker cannot launch a replay attack in our protocol and it is resistant to this kind of attack.

7.2.3. Mutual authentication

In our protocol, U_i, S_j and RC authenticate each other by the following ways:

1. Mutual authentication between S_j and RC: In login phase, U_i sends his login request $\{UID_i, C_1, M_{ik}\}$ to the server S_j. Then, in authentication and session key agreement phase, S_j computes some information and transmits the message $\{UID_i, C_1, M_{ik}, SID_j, C_3, M_{jk}\}$ to RC. After that, RC calculates $K_j' = h(SID_j||w)$, $M_{jk}' = h(K_j'||C_3)$ and checks if $M_{jk}' = M_{jk}$. If it is true, then RC authenticates S_j. On the other hand, after obtaining response message $\{D_i, M_{kj}, G_i, M_{ki}\}$ from RC in step 3 of authentication phase, S_j authenticates RC based on the parameter M_{kj}. Hence, mutual authentication is provided between S_j and RC.

2. Mutual authentication between U_i and RC: In login phase, U_i sends his login request $\{UID_i, C_1, M_{ik}\}$ to the server S_j. Then, in authentication and session key agreement phase, S_j computes some information and transmits the message $\{UID_i, C_1, M_{ik}, SID_j, C_3, M_{jk}\}$ to RC. After that, RC does some calculations and authenticates U_i based on the parameter M_{ik}. On the other hand, RC transmits the response message $\{D_i, M_{kj}, G_i, M_{ki}\}$ to S_j and S_j sends the message $\{G_i, M_{ki}, C_3, M_{ji}\}$ further to U_i. U_i authenticates RC based on the parameter M_{ki}. Hence, mutual authentication is provided between U_i and RC.

3. Mutual authentication between U_i and S_j: In login phase, U_i sends his login request $\{UID_i, C_1, M_{ik}\}$ to the server S_j. Then, in authentication and session key agreement phase, S_j computes some information and transmits the message $\{UID_i, C_1, M_{ik}, SID_j, C_3, M_{jk}\}$ to RC. After that, RC does some calculations and transmits the response message $\{D_i, M_{kj}, G_i, M_{ki}\}$ to S_j and S_j sends the message $\{G_i, M_{ki}, C_3, M_{ji}\}$ further to U_i. Then, U_i authenticates S_j based on the parameter M_{ji}. On the other hand, U_i sends the message M_{ij} to S_j and S_j authenticates U_i based on the parameter M_{ij}. Hence, mutual authentication is provided between U_i and S_j.

7.2.4. Perfect forward secrecy

It means that E cannot find the earlier established session keys although he knows the master secret key w of RC. In our protocol, U_i and S_j share a session key $SK \equiv T_{r_i r_j}(X)$ mod p after mutual authentication where r_i and r_j are random nonces generated by U_i and S_j, respectively. The disclosure of the secret key w is not beneficial to E as the session key does not depend on it. The attacker cannot calculate r_i, r_j and $T_{r_i r_j}(X)$ from C_1 and C_2 because of discrete logarithm problem (DLP) or Diffie-Hellman problem (DHP). Thus, the perfect forward secrecy is offered by our protocol.

7.2.5. Stolen smart card attack

In this attack, E gets access of an authorized user's smart card and then, he may try to change/guess the password, or may login to the server to impersonate the user. Assume that E somehow gets the U_i's SC. Now, he can take the stored information $\{B_i, D_1, X, R, n, Z_i, D_2, h(.), p\}$ from it where $B_i = h(v_i||w) \oplus h(C_i||P_i)$, $D_1 = v_i \oplus h(w)$, $Z_i = h(PW_i||ID_i) mod n \oplus N_i$ and $D_2 = h(C_i||N_i||P_i) mod n$. Here, we see that B_i, Z_i and D_2 are related to ID_i and PW_i and E cannot guess the parameters ID_i and PW_i from these as they are protected under one way hash function and fuzzy verifier. Moreover, E cannot alter the password without having ID_i and PW_i. Hence, this attack does not exist in our protocol.

7.2.6. Known-key security

It is an main security feature which means that an attacker cannot find any other session key based on a known session key. The session key $SK \equiv T_{r_i r_j}(X)$ mod p depends on the two random nonces r_i and r_j in our protocol. One session key is not dependent on the other session keys as random nonces are different in each session. Hence, if an attacker knows one session key $SK \equiv T_{r_i r_j}(X)$ mod p, then he cannot find the other session key $SK' \equiv T_{r_i' r_j'}(X)$ mod p without having r_i' and r_j'. Therefore, our protocol provides known-key security.

7.2.7. Privileged insider attack

In our scheme, U_i sends $C_i = h(ID_i||N_i)$ and $P_i = h(PW_i||N_i)$ to RC instead of sending actual ID_i and PW_i in user registration phase. RC's insider cannot get PW_i because of one-way hash function and unknown parameter N_i. Moreover, let us assume that a privileged insider gets access to the U_i's SC and extracts all the stored information $\{B_i, D_1, X, R, n, Z_i, D_2, h(.), p\}$ from it [26, 27]. Then, he cannot guess the identity ID_i and the password PW_i of the user U_i from C_i and P_i as these are protected under one way hash function and he does not know N_i. Hence, our protocol is resistant to this attack.

7.2.8. Efficient login phase

If U_i enters wrong identity ID_i or password PW_i in login phase of our protocol, then SC detects it and discards the login request. Therefore, our schemes does not increase the unnecessary network congestion, communication cost, computation cost and avoids the denial of service attack. Hence, our scheme offers an efficient login phase.

7.2.9. Server spoofing and registration center attacks

RC keeps w, its secret key, only with itself in our scheme and it does not share the key with all servers. E cannot find w from the information sent over the insecure channel as it is kept safe by one way hash function h(.). Thus, in our protocol, E is not able to impersonate RC. Moreover, each S_j keeps different secret key $h(SID_j||w)$. Therefore, E or a valid server but malicious cannot impersonate other server without knowing $h(SID_j||w)$. Thus, our protocol resists server spoofing attack.

8. Security verification using AVISPA tool

Here, we provide the security verification of our protocol using AVISPA tool. The AVISPA stands for Automated Validation of Internet Security Protocols and Applications. This tool is free available and based on the Dolev-Yao [38] intruder model. The details about this tool can be found in [39]. It implements four back-ends and abstraction based methods that are integrated through the High Level Protocol Specification Language (HLPSL). We simulate our scheme using the widely used back-end the On-the-Fly Model-Checker (OFMC). The simulation results are shown in Fig. 1 and its summary section shows that it is SAFE against replay and man-in-the-middle attacks. The AVISPA code of our scheme is omitted because of the space limitation.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/devender/Avispa/avispa-1.1/testsuite/results/MultiServer.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 46.26s
  visitedNodes: 4096 nodes
  depth: 12 plies
```

Fig. 1. Simulation result in OFMC back-end

9. Comparative study

Here, we compare our protocol with the related protocols [21, 25] in terms of computational cost and security features.

Table 2. Security features comparisons

Security features	Lee et al. [21]	Li et al. [25]	Ours
Privileged insider attack	N	N	Y
Registration center spoofing attack	N	Y	Y
Server spoofing attack	N	Y	Y
Single Registration	N	Y	Y
Support password change	N	Y	Y
Efficient login phase	N	Y	Y
User anonymity	Y	N	Y
Mutual authentication	Y	Y	Y
Perfect forward secrecy	Y	Y	Y
Replay attack	Y	Y	Y
Known-key security	Y	Y	Y
Smart card loss attack	Y	N	Y

N: Scheme does not provide protection against the security feature; Y: Scheme provides protection against the security feature

Table 3. Computational cost in login and authentication phases

Computational cost	Lee et al. [21]	Li et al. [25]	Ours
User costs	$3T_C + 5T_H$	$3T_C + 7T_H$	$3T_C + 10T_H$
Server costs	$3T_C + 6T_H$	$2T_C + 5T_H$	$2T_C + 5T_H$
RC costs	–	$1T_C + 8T_H$	$1T_C + 9T_H$
Total costs	$6T_C + 11T_H$	$6T_C + 20T_H$	$6T_C + 24T_H$

T_H: Time complexity of hash operation; T_C: Time complexity of extended chaotic function

9.1. Security features

The security features of our and the other related protocols [21, 25] are shown in Table 2. It is evident from this table that our protocol provides protection against more security features as compared to the related protocols [21, 25].

9.2. Computational cost

For computational cost comparison, we ignore the costs of initialization, user registration, and password change phases as they are performed only once. We compare only the computational costs of the login and authentication phases as these phases are performed in every time whenever a user wants to access any server. Table 3 presents the computational costs of our protocol along with related protocols [21, 25] for user, server and registration center in login and authentication phases. This table shows that our protocol requires 4 and 13 more hash operations than the protocol [25] and the protocol [21], respectively. But, it provides all the security features as given in Table 2 while others do not.

10. Conclusion

In this paper, we have cryptanalyzed the Li et al.'s user authentication and key agreement protocol and found some security problems in it that include user anonymity, privileged insider

and smart card loss attacks. Here, we have proposed an improved protocol by removing the security weaknesses of their scheme. We have proved formally using BAN logic that our protocol offers mutual authentication. We have also showed using AVISPA tool that it is secured against passive and active attacks. Further, we have analyzed its security formally using random oracle and discussed its informal security analysis to show that it provides the required functionalities and resists various possible attacks. Also, it is more secure than the related schemes.

References

[1] Lamport L., *Password authentication with insecure communication*, Communications of the ACM, **24** 11, pp. 770–772, 1981.

[2] Gwoboa H., *Password authentication without using a password table*, Information Processing Letters, **55**5, pp. 247–250, 1995.

[3] Jan J. K. and Chen Y. Y., *Paramita wisdom password authentication scheme without verification tables*, Journal of Systems and Software, **42**(1), pp. 45–57, 1998.

[4] He D., Kumar N., and Chilamkurti N., *A secure temporal-credential-based mutual authentication and key agreement scheme with pseudo identity for wireless sensor networks*, Information Sciences, **321**, pp. 263–277, 2015.

[5] He D. and Wang D., *Robust biometrics-based authentication scheme for multiserver environment*, IEEE Systems Journal, **9**3, pp. 816–823, 2015.

[6] Li X., Niu J., Khan M. K., and Liao J., *An enhanced smart card based remote user password authentication scheme*, Journal of Network and Computer Applications, **36**5, pp. 1365–1371, 2013.

[7] Liu Y., Gong P., Yan X., and Li P., *On the security of a dynamic identity-based remote user authentication scheme with verifiable password update", International Journal of Communication Systems*, **28**5, pp. 842–847, 2015.

[8] Li L. H., Lin L. C., and Hwang M. S., *A remote password authentication scheme for multiserver architecture using neural networks*, IEEE Transactions on Neural Networks, **12**6, pp. 1498–1504, 2001.

[9] Lin I. C., Hwang M. S., and Li L. H., *A new remote user authentication scheme for multi-server architecture*, Future Generation Computer Systems, **19**1, pp. 13–22, 2003.

[10] Juang W. S., *Efficient multi-server password authenticated key agreement using smart cards*, IEEE Transactions on Consumer Electronics, **50**1, pp. 251–255, 2004.

[11] Chang C. C. and Lee J. S., *An efficient and secure multi-server password authentication scheme using smart cards*," International Conference on Cyberworlds, pp. 417–422, IEEE, 2004.

[12] Tsaur W. J., Wu C. C. , and Lee W. B., *An enhanced user authentication scheme for multi-server internet services*, Applied Mathematics and Computation, **170**1, pp. 258–266, 2005.

[13] Tsai J. L., *Efficient multi-server authentication scheme based on one-way hash function without verification table*, Computers & Security, **27**3, pp. 115–121, 2008.

[14] Chen Y., Huang C. H., and Chou J. S., *Comments on two multi-server authentication protocols*, IACR Cryptology ePrint Archive, vol. 2008, p. 544, 2008.

[15] Wang R. C., Juang W. S., and Lei C. L., *User authentication scheme with privacy-preservation for multi-server environment*, IEEE Communications Letters, **13**2, pp. 157–159, 2009.

[16] Tsaur W. J., Li J. H., and Lee W. B., *An efficient and secure multi-server authentication scheme with key agreement*, Journal of Systems and Software, **85**4, pp. 876–882, 2012.

[17] Liao Y. P. and Wang S. S. , *A secure dynamic id based remote user authentication scheme for multi-server environment*, Computer Standards & Interfaces, **31**1, pp. 24–29, 2009.

[18] Hsiang H. C. and Shih W. K. , *Improvement of the secure dynamic id based remote user authentication scheme for multi-server environment*, Computer Standards & Interfaces, **31**6, pp. 1118–1123, 2009.

[19] Sood S. K., Sarje A. K., and Singh K., *A secure dynamic identity based authentication protocol for multi-server architecture*, Journal of Network and Computer Applications, **34**2, pp. 609–618, 2011.

[20] Li X., Xiong Y., Ma J., and Wang W., *An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards*, Journal of Network and Computer Applications, **35**2, pp. 763–769, 2012.

[21] Lee C. C., Lou D. C., Li C. T., and Hsu C. W., *An extended chaotic-maps-based protocol with key agreement for multiserver environments*, Nonlinear Dynamics, **76**1, pp. 853–866, 2014.

[22] Li C. T., Lee C. C., and Weng C. Y., *An extended chaotic maps based user authentication and privacy preserving scheme against dos attacks in pervasive and ubiquitous computing environments*, Nonlinear Dynamics, **74**4, pp. 1133–1143, 2013.

[23] Lee C. C., Chen C. L., Wu C. Y., and Huang S. Y., *An extended chaotic maps-based key agreement protocol with user anonymity*, Nonlinear Dynamics, **69**1, pp. 79–87, 2012.

[24] Zhao F., Gong P., Li S., Li M., and Li P., *Cryptanalysis and improvement of a three-party key agreement protocol using enhanced chebyshev polynomials*, Nonlinear Dynamics, **74**1-2, pp. 419–427, 2013.

[25] Li X., Niu J., Kumari S., Islam S. H., Wu F., Khan M. K., and Das A. K., *A novel chaotic maps-based user authentication and key agreement protocol for multi-server environments with provable security*, Wireless Personal Communications, **89**2, pp. 569–597, 2016.

[26] Kocher P., Jaffe J., and Jun B., *Differential power analysis*, Annual International Cryptology Conference, pp. 388–397, Springer, 1999.

[27] Messerges T. S., Dabbish E. A., and Sloan R. H., *Examining smart-card security under the threat of power analysis attacks*, IEEE transactions on computers, **51**5, pp. 541–552, 2002.

[28] Wang D. and Wang P., *Two birds with one stone: Two-factor authentication with security beyond conventional bound*, IEEE Transactions on Dependable and Secure Computing, **15**4, pp. 708–722, 2016.

[29] Mason J. C. and Handscomb D. C., Chebyshev polynomials. CRC Press, 2002.

[30] Bergamo P., D'Arco P., De Santis A., and Kocarev L., *Security of public-key cryptosystems based on chebyshev polynomials*, IEEE Transactions on Circuits and Systems I: Regular Papers, **52**7, pp. 1382–1393, 2005.

[31] Zhang L., *Cryptanalysis of the public key encryption based on multiple chaotic systems*, Chaos, Solitons & Fractals, **37**3, pp. 669–674, 2008.

[32] He D., Chen Y., and Chen J., *Cryptanalysis and improvement of an extended chaotic maps-based key agreement protocol*, Nonlinear Dynamics, *69*3, pp. 1149–1157, 2012.

[33] Das A. K., *A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks*, Peer-to-peer Networking and Applications, **9**1, pp. 223–244, 2016.

[34] Burrows M., Abadi M., and Needham R. M., *A logic of authentication*, Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, **426**, pp. 233–271, The Royal Society, 1989.

[35] Chandrakar P. and Om H., *A secure and robust anonymous three-factor remote user authentication scheme for multi-server environment using ecc*, Computer Communications, **110**, pp. 26-34, 2017.

[36] Das A.K., *A secure and effective biometric-based user authentication scheme for wireless sensor networks using smart card and fuzzy extractor*, International Journal of Communication Systems, **30**1, pp. e2933, 2017.

[37] Kumar D., Chand S., and Kumar B., *Cryptanalysis and improvement of an authentication protocol for wireless sensor networks applications like safety monitoring in coal mines*, Journal of Ambient Intelligence and Humanized Computing, **10**2 pp. 641–660, 2019.

[38] Dolev D. and Yao A., *On the security of public key protocols*, IEEE Transactions on information theory, **29**2, pp. 198–208, 1983.

[39] *AVISPA. Automated Validation of Internet Security Protocols and Applications* . http://www.avispa-project.org/. Online; Accessed on December, 2016.