Wireless Pers Commun (2018) 101:1307–1333
https://doi.org/10.1007/s11277-018-5764-8

An Improved and Secure Two-factor Dynamic ID Based Authenticated Key Agreement Scheme for Multiserver Environment

Shreeya Swagatika Sahoo[1] · Sujata Mohanty[1] · Banshidhar Majhi[1]

Published online: 23 April 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract The smart card based password authentication scheme is one of the most important and efficient security mechanism, which is used for providing security to authorized users over an insecure network. In this paper, we analyzed major security flaws of Jangirala et al.'s scheme and proved that it is vulnerable to forgery attack, replay attack, user impersonation attack. Also, Jangirala et al.'s scheme fail to achieve mutual authentication as it claimed. We proposed an improved two factor based dynamic ID based authenticated key agreement protocol for the multiserver environment. The proposed scheme has been simulated using widely accepted AVISPA tool. Furthermore, mutual authentication is proved through BAN logic. The rigorous security and performance analysis depicts that the proposed scheme provides users anonymity, mutual authentication, session key agreement and secure against various active attacks.

Keywords Smart card · Password · Authentication · Data security · BAN logic · AVISPA

1 Introduction

The rapid growth of Internet and telecommunication has made the people work easier. In present days, more and more online activities, such as online shopping, online ticket booking, online bill payment, online gaming, and online medical services, etc. are provided through the Internet. To provide security in open channel is a prominent challenge in internet-based service. Generally to validate the legitimacy of a user, two mechanisms namely mutual authentication and secure key exchange are widely used. During the last decade, many passwords based mutual authentication schemes have become a prominent research topic. The smart card based user authentication scheme for the multi-server

✉ Shreeya Swagatika Sahoo
shreeya.swagatika@gmail.com

1 National Institute of Technology Rourkela, Rourkela, India

environment is simple and user friendly, which establishes the secure and authorized communication over the insecure channel. In the multi-server environment, three communication parties are involved, namely server, registration center, and client. Before the server and the client starts a new session, the identity of the two parties must be authenticated.

In 1981, Lamport [1] introduced the password based authentication scheme, in which the server stores the password. It was vulnerable to passive attack in case the password table is leaked or compromised, and the intruders could modify stored password in the system. Later, Hwang et al. [2] proposed an improved protocol which overcomes the weaknesses of the Lamport's scheme. Later Yang and Shieh [3] intended two types of password authentication schemes based on timestamp and nonce with the smart card. In 2000, Hwang et al. [4] proposed an advanced remote user authentication scheme based on the ElGamals public key cryptosystem. Many smart card based authentication schemes have been suggested for single server environment [5–15]. The major drawback in a single server scheme is maintaining verification table, protecting user IDs and passwords.

In 2001, Li et al. [16] suggested a remote password-based authentication scheme using neural networks for multiserver environment. In this scheme, the user has to register himself once with various servers and remember his credentials, such as user *ID* and password. Unlike the other schemes, it does not need verification table to verify the legitimacy of the user. In 2003, Lin et al. [17] proposed a new scheme in which the user does not have to remember the user *ID* and password. However, Juang [18] demonstrated that Lin et al.'s scheme cannot achieve mutual authentication and session key security. To overcome these weaknesses, Juang suggested an enhanced scheme for the multiserver environment using the symmetric encryption algorithm in which, the user has to register himself at the registration center once and thereby can access various servers. The scheme can overcome repeated registration problem and also a large amount of memory is not needed to store the parameters for authentication. In the same year, Chang et al. [19] showed that Juang's scheme cannot resist dictionary attack and also the computational cost along with the communicational overhead of the scheme is high. Chang et al. proposed a scheme which is more efficient and secure than the Juang's scheme. Unfortunately, their scheme suffers from insider attack and lack of wrong password checking. In 2004, Tsaur et al. [20] suggested a new scheme based on the RSA cryptosystem and Lagrange interpolating polynomial. But, this scheme does not achieve user anonymity and also computational, and the communicational cost is high. In 2006, Yang et al. suggested a password-based authentication scheme based on two server architecture in which the front-end server communicates with directly to the end user, and the back-end control server stays behind the scene [21].

In 2008, Tsai [22] constructed a multiserver authentication scheme using one-way hash function and nonce. Most of the authentication schemes for the multiserver environment are based on static user *ID* which helps the adversary to intercept user *ID* from the public networks and impersonate as an authentic user. In 2009, Liao et al. [23] suggested a scheme based on dynamic user *ID* which achieved user anonymity, mutual authentication, session key agreement, and can withstand various attacks. Later, Hsiang et al. [24] depicted that Lio et al.'s scheme is insecure to server and registration center spoofing attack, insider attack, masquerade attack. To resolve these issues, they proposed a new scheme which provides more security than other schemes. Unfortunately, Hsiang et al.'s scheme suffers from a server spoofing attack, masquerade attack, and does not achieve mutual authentication. In 2011, to overcome these weaknesses, Lee et al. [25] and Sood et al. [26] suggested two authentication schemes for the multiserver environment.

Li et al. [27] noticed that Sood et al.'s scheme suffers from stolen smart card attack, leak-of-verifier attack, and impersonation attack. In 2013, Li et al. [28] found out that Lee et al.'s scheme could not resist forgery attack and server spoofing attack, and not provides mutual authentication. They proposed a scheme which can overcome these flaws and claimed that their scheme is more secure. Also, their scheme can achieve user anonymity and mutual authentication. Zhao et al. [29] proposed a new scheme against Li et al.'s as they noticed that Li et al.'s scheme is susceptible to smart card lost attack, offline dictionary attack, replay attack, impersonation attack, and server spoofing attack. In 2014, Xue et al. [30] came up with a new scheme and demonstrated that Li et al.'s scheme is inefficient to replay attack, denial-of-service attack, smart card, forgery attack, and eavesdropping attack. Their scheme not only overcomes the flaws of the Li et al.'s scheme but also achieves some security features such as traceability and identity protection. Many two factor and three factor based authentication schemes have been suggested for multi-server environment [31–33]. In 2015, Shunmuganathan et al. [34] suggested that Li et al.'s scheme is vulnerable to offline password guessing, forgery attack, and smart card loss attack. Later, in 2017 Jangirala et al. demonstrated that Shunmuganathan et al.'s scheme cannot withstand password guessing attack, user impersonation attack, stolen smart card attack, forgery attack, and replay attack. Furthermore, this scheme fails to achieve forward secrecy and also has poor repairability [35].

We made a rigorous cryptanalysis of Jangirala et al.'s scheme and proved that the scheme is unable to withstand user impersonation attack, replay attack, and forgery attack. Furthermore, this scheme is failed to achieve mutual authentication. To overcome these weaknesses, we proposed an enhanced dynamic ID-based mutual authentication scheme for the multiserver environment using a smart card. In addition, the mutual authentication of the proposed scheme has been proved using BAN logic. Also, the simulation of the proposed scheme has been done using widely accepted AVISPA tool.

The rest of the paper is organized as follows: in Sect. 2, we provide a brief review of Jangirala et al.'s protocol. Section 3 points out the security weaknesses of Jangirala et al.'s protocol. In Sect. 4, we propose our protocol for the multiserver environment using dynamic identity. Security analysis and simulation of the proposed scheme has been shown in Sects. 5 and 6 respectively. The comparison of the cost and functionality of the proposed protocol with other related protocols is discussed in Sect. 7. Finally, we concluded in Sect. 8.

2 Overview of Jangirala et al.'s Scheme

This section, we briefly review Jangirala et al.'s scheme [35], which is an enhancement over Shunmuganathan et al.'s scheme [34]. The notations used throughout this paper are given in Table 1. The scheme is composed of four phases, such as registration phase, login phase, authentication phase, and password change phase. In this scheme, there are three participants: the user (U_r), the server (S_u), and the registration cente (RS). The RS chooses the master key R_x and the secret number R_y to calculate $h(R_x \| R_y)$ and $h(R_y)$ assuming that RS is the trusted party. The details of each phase are given in Table 2.

Table 1 Notation used

Notation	Description
U_r	rth user
S_u	uth server
RS	Registration Center
SC	Smart Card
A_k	Adversary
ID_u	User (uth) identity
PW_u	User's password
CID_u	Dynamic ID of user
SID_j	Identity of server
R_x	Master key generated by RS
R_y	Secret number generated by RS
b	Random number generated by user
SK_f	Session key
$h(\cdot)$	Cryptographic one way hash function
$\|\|$	Concatenation operation
$\oplus$	Bitwise XOR operator

2.1 Registration Phase

A user (U_r) initially registers with the RS by proceeding through the following steps:

Step 1: The user U_r first selects ID_u, password PW_u and a random number b. Then submits ID_u and $A_u = h(ID_u \oplus PW_u \oplus b)$ to the RS for registration through a secure channel.

Step 2: RS calculates B_u, C_u, D_u, E_u as follows and embeds the parameters into SC.
$B_u = h(ID_u \| R_x)$
$C_u = h(ID_u \| A_u \| h(R_y))$
$D_u = h(B_u \| h(R_x \| R_y))$
$E_u = h(R_x \| R_y) \oplus B_u$

Step 3: Now, the server issues smart card to the user. On receiving SC, the user calculates $L_u = b \oplus h(ID_u \| PW_u)$ and stores L_u into the smart card. Finally, the smart card contains $(C_u, D_u, E_u, L_u, h(R_y), h(.))$.

2.2 Login Phase

In this phase the user U_r sends the login message to the server S_u as follows:

Step 1: User U_r inserts his SC into the card reader and enters his ID_u and password PW_u. Then, the SC computes A_u and C_u^* and compares C_u with C_u^* .

Step 2: If the condition satisfies, then the SC proceeds to the next steps. Otherwise, it terminates the session.

Step 3: The smart card generates random number N_i and computes the following parameters.
$P_{us} = E_u \oplus h(h(SID_u \| h(R_y) \| N_i))$
$CID_u = A_u \oplus h(D_u \| SID_u \| N_i)$

Table 2 Registration and Login phase of Jangirala et al.

User(U_r)		Registration center(RS)
Selects a random number b		
Chooses ID_u and PW_u		
Computes $A_u = h(ID_u \oplus b \oplus PW_u)$		
	(ID_u, A_u) $\xrightarrow{}$ $(Securechannel)$	
		$B_u = h(A_u \| R_x)$
		$C_u = h(ID_u \| A_u \| h(R_y))$
		$D_u = h(B_u \| h(R_x \| R_y))$
		$E_u = h(R_x \| R_y) \oplus B_u$
	$(C_u, D_u, E_u, h(R_y), h(.))$ $\xleftarrow{}$ $(SecureChannel)$	
Computes $L_u = b \oplus h(ID_u \| PW_u)$		
Stores L_u into SC		
In login phase insert		
SC into card reader		
Inputs ID_u and PW_u		
SC calculates		
$b = L_u \oplus h(ID_u \| PW_u)$		
$A_u = h(ID_u \oplus b \oplus PW_u)$		
Then calculates C_u^* as follows		
$C_u^* = h(ID_u \| A_u \| h(R_y))$		
Checks $C_u \stackrel{?}{=} C_u^*$		
Generates a random number N_i and calculates		
$CID_u = A_u \oplus h(D_u \| SID_u \| N_i)$		
$P_{us} = E_u \oplus h(h(SID_j \| h(R_y) \| N_i))$		
$M_1 = h(P_{us} \| CID_u \| A_u \| N_i)$		
$M_2 = h(SID_u \| h(R_y)) \oplus N_i$		
	$(P_{us}, CID_u, M_1, M_2)$ $\xrightarrow{}$ $(Publicchannel)$	

$M_1 = h(P_{us} \| CID_u \| A_u \| N_i)$
$M_2 = h(SID_u \| h(R_y)) \oplus N_i$
Then U_r sends $(P_{us}, CID_u, M_1, M_2)$ to the server as login message.
The registration and login phase of Jangirala et al. scheme are given in Table 2.

2.3 Authentication Phase

To verify the login message and perform the mutual authentication, the server proceeds as per the following steps.

Step 1: Upon receiving the login message, S_u computes $h(h(SID_u \| h(R_y)) \oplus M_2$ to find N_i. Then computes $P_{us} \oplus h(h(SID_u \| h(R_y) \| N_i))$ to find E_u and obtain B_u by computing $(E_u \oplus h(R_x \| R_y))$. Then, the server computes $CID_u \oplus h(D_u \| SID_u \| N_i)$ to computes A_u.

Step 2: S_u calculate $h(P_{us} \| CID_u \| A_u \| N_i)$ and compares with M_1. If they are not equal server rejects the login message. Otherwise, server generates a random number N_j and computes $M_3 = h(SK_{ij} \| A_u \| SID_u \| N_j)$ and $M_4 = (SK_{ij} \oplus N_j)$, where $SK_{ij} = h(h(B_u \| h(R_x \| R_y)) \| A_u)$. Then, S_u sends (M_3, M_4) to the U_r for authentication.

Step 3: After receiving the message (M_3, M_4) the U_r computes $SK_{ij} = h(D_u \| A_u)$ which is available previously. Next, extract $N_j = SK_{ij} \oplus M_4$. Then calculate $h(SK_{ij} \| SID_u \| A_u \| N_j)$ and compares it with M_3. If both are equal, the server is successfully authenticated and proceed to the next step. Otherwise the user rejects the message and terminates the session. The details of the authentication scheme is given in Table 3.

2.4 Password Change Phase

A valid user can change his password as follows:

Step 1: The user inserts his smart card into a card reader and enters his/her identity ID_u and password PW_u. The smart card computes $b^* = L_u \oplus h(ID_u \| PW_u)$, $A_u^* = h(ID_u \oplus PW_u \oplus b^*)$ and $C_u^* = h(ID_u \| h(R_y) \| A_u^*))$.

Step 2: Then SC checks for the computed $C_u^* \stackrel{?}{=} C_u$. If they are not equal, then it will reject the password change request. Otherwise, the user is allowed to input a new password PW_u^n.

Step 3: The smart card computes the following parameters:
$A_u^n = h(ID_u \oplus b^* \oplus PW_u^n)$
$C_u^n = h(ID_u \| A_u^n \| h(R_y))$
$L_u^n = b^* \oplus h(ID_u \| PW_u^n)$

And finally, the SC replaces C_u, L_u with C_u^n, L_u^n respectively and replaces the new password.

3 Cryptanalysis of Jangirala et al.'s Scheme

In this section, the weaknesses of Jangirala et al. scheme is discussed. We analyzed that this scheme is vulnerable to forgery attack, replay attack, user impersonation attack and does not achieve forward secrecy. Also, this scheme failed to achieve mutual authentication.

3.1 Forgery Attack

For instance, the smart card is lost or stolen. An adversary (A_k) can obtained information $(C_u, D_u, E_u, h(.), h(R_y))$ from the smart card and can intercept the login message $(CID_u, P_{us}, M_1, M_2)$ from the public channel. Then A_k first computes the random number $N_i = M_2 \oplus h(SID_u \| h(R_y))$. Next, A_k gets A_u by computing $A_u = CID_u \oplus h(D_u \| SID_j \| N_i)$.

Table 3 Authentication phase of Jangirala et al.'s scheme

User (U_r)		Server (S_u)
		After getting the message Server checks for the validation of message, Computes $N_i = h(SID_u \| h(R_y)) \oplus M_2$ $E_u = P_{us} \oplus h(h(SID_u \| h(R_y)) \| N_i))$ $B_u = E_u \oplus h(R_x \| R_y))$ $D_u = h(B_u \| h(R_x \| R_y))$ $A_u = CID_u \oplus h(D_u \| SID_u \| N_i)$ Checks $h(P_{us} \| CID_u \| A_u \| N_i) \stackrel{?}{=} M_1$ If it holds then, User is authenticated by server S. Server generates a random number N_j and Computes the following equations $SK_{ij} = h(h(B_u \| h(R_x \| R_y) \| A_u))$ $M_3 = h(SK_{ij} \| A_u \| N_j \| SID_u)$ $M_4 = SK_{ij} \oplus N_j$
	(M_3, M_4) $\xleftarrow{\hspace{2cm}}$ $(Publicchannel)$	
Computes $SK_{ij} = h(D_u \| A_u)$ $N_j = SK_{ij} \oplus M_4$ Checks $h(SK_{ij} \| SID_u \| A_u \| N_j) \stackrel{?}{=} M_3$ Then compute M_5 $M_5 = h(SK_{ij} \| A_u \| SID_u \| N_i \| N_j)$		
	(M_5) $\xrightarrow{\hspace{2cm}}$ $(Publicchannel)$	
		Checks $h(SK_{ij} \| A_u \| SID_j \| N_i \| N_j\ ) \stackrel{?}{=} M_5$ If both are equal, then it will calculate the session key
	$\xleftrightarrow{SK_f = h(SK_{ij} \| A_u \| SID_j \| N_i \| D_u \| N_j)}$	

To forge the login message, he can generate a new random number N_i^* and calculates $P_{us}^* = E_u \oplus h(h(SID_j \| h(R_y)) \| N_i^*)$, $M_1^* = h(P_{us}^* \| CID_u \| A_u \| N_i^*)$. Then, sends the login message $(CID_u, P_{us}^*, M_1^*, M_2)$ to S_u. Thus, this scheme cannot withstand forgery attacks.

3.2 Replay Attack

An intruder attempts to eavesdrop a valid login message and replay the message $(CID_u, P_{us}^*, M_1^*, M_2)$ to S_u. Upon receiving the login message, the server sends the message (M_3, M_4) to the U_r. Next, to acknowledge the S_u, adversary A_k computes

$M_5 = h(SK_{ij} \| A_u \| SID_j \| N_i \| N_j)$. Now A_k can compute $SK_{ij} = h(D_u \| A_u)$, where D_u is obtained from the SC and A_u is calculated by A_k as explained in forgery attack. After computing SK_{ij}, an adversary can calculate $N_j = SK_{ij} \oplus M_4$ and sends M_5 to the S_u. So, this scheme can not resist replay attack.

3.3 User Impersonation Attack

This scheme does not withstand impersonation attack. Without knowing the user ID_u and PW_u, the adversary can transmit the login message $(CID_u, P^*_{us}, M^*_1, M_2)$ to S_u as we discussed in replay attack. Upon getting the login message, the server tries to calculate $N^*_i = h(SID_j \| h(R_y)) \oplus M_2$, $A^*_u = CID_u \oplus h(D_u \| SID_u \| N^*_i)$. Then, the received M_1 will be equal to $h(P^*_{us} \| CID_u \| A^*_u \| N^*_i)$. Thus, the attacker can successfully impersonate the user.

3.4 Mutual Authentication

The above discussed replay attack proves that an adversary can authorize the server and sends M^*_5 by using its own information, i.e. $M^*_5 = h(SK^*_{ij} \| A^*_u \| SID_j \| N^*_i \| N_j)$. Then S_u checks M^*_5 and does not know about the random number N^*_i, A^*_u and SK^*_{ij}. So, this scheme does not achieve proper mutual authentication.

4 Proposed Scheme

In this section, we proposed an improved smart card based authentication scheme for the multiserver environment, which can overcome all the weaknesses of Jangirala et al.'s scheme. The proposed scheme comprises of three participants the server(S_u), the user(U_r), and the registration center(RS). The proposed scheme has four phases: registration phase, login phase, authentication phase, and password change phase. The notations used in this scheme are described in Table 1.

4.1 Registration Phase

A new user U_r registers himself with registration center RS before communicating with server S_u. The RS generates a master key R_x and a secret number R_y. Then registration center calculates $h(R_x)$ and $h(R_x \| R_y)$ and shares it with the server in a secure channel, where S_u is registered before with RS. To complete the registration phase, U_r and RS execute the following steps as given in Table 4.

Step 1: The U_r freely chooses his/her user name and password. And also select a random number b to compute $PW_n = h(PW_u \| b)$. Then, U_r sends the user ID_u and PW_n to the RS through secure channel.

Step 2: Upon receiving the message (ID_u, PW_n) from the U_r, RS computes the following parameters:
$B_u = h(ID_u \| R_x)$
$C_u = h(h(R_x \| R_y) \| h(R_y)) \oplus B_u$
$D_u = h(ID_u \| PW_n \| C_u) \oplus h(R_x \| R_y)$
$E_u = h(ID_u \| PW_n \| h(R_y))$

Table 4 Registration and Login phase of proposed scheme

User(U_r)		Registration Center(RS)
Selects a random number b		
Chooses ID_u and PW_u		
$P_u = b \oplus h(ID_u \| PW_u)$		
Computes $PW_n = h(PW_u \| b)$		
	(ID_u, PW_n) $\xrightarrow{\hspace{2cm}}$ $(Securechannel)$	
		$B_u = h(ID_u \| R_x)$
		$C_u = h(h(R_x \| R_y) \| h(R_y)) \oplus B_u$
		$D_u = h(ID_u \| PW_n \| C_u) \oplus h(R_x \| R_y)$
		$E_u = h(ID_u \| PW_n \| h(R_y))$
	$(C_u, D_u, E_u, h(R_y), h(.))$ $\xleftarrow{\hspace{2cm}}$ $(Publicchannel)$	
Stores P_u into SC		

User(U_r)		Server(S_u)
In login phase inserts		
SC into card reader		
Inputs ID_u and PW_u		
SC calculates		
$b = P_u \oplus h(ID_u \| PW_u)$		
$PW_n = h(PW_u \| b)$		
$h(R_x \| R_y) = h(ID_u \| PW_n \| C_u) \oplus D_u$		
$B_u = h(h(R_x \| R_y) \| h(R_y)) \oplus C_u$		
Then calculates E_u^* as follows		
$E_u^* = h(ID_u \| PW_n \| h(R_y))$		
If $E_u \stackrel{?}{=} E_u^*$, Then generates		
a random number N_i and computes		
$P_{us} = h(SID_j \| h(R_x \| R_y) \| h(R_y) \| N_i) \oplus C_u$		
$F_u = h(h(R_x \| R_y) \| N_i)$		
$R_u = h(F_u \| C_u \| h(R_y))$		
$CID_u = h(R_u \| D_u \| h(R_y)) \oplus F_u$		
$M_1 = h(P_{us} \| CID_u \| F_u)$		
	$(CID_u, P_{us}, M_1, N_i)$ $\xrightarrow{\hspace{2cm}}$ $(Publicchannel)$	

Step 3: Then, the RS embeds the parameters $(C_u, D_u, E_u, h(R_y), h(.))$ into SC and issues it to the U_r. After receiving the information, U_r calculates $P_u = b \oplus h(ID_u \| PW_u)$ and stores P_u into the SC.

4.2 Login Phase

In this phase, the U_r inserts the SC to the smart card reader to login with the S_u. The details of the login phase described as follows and presented in Table 4.

Step 1: The U_r submits his user ID_u and PW_u. Then SC calculates b by using the stored information P_u as $b = P_u \oplus h(ID_u \| PW_u)$.

Step 2: Now, the SC calculates $E_u^* = h(ID_u \| PW_n \| h(R_y))$, where ID_u is given by the user and $PW_n = h(PW_u \| b)$. Then, it checks whether the stored E_u is equal to E_u^*. If they are not equal, SC terminates the session. Otherwise, SC authenticates the U_r and generates a nonce to compute the following parameters.
$P_{us} = h(SID_u \| h(R_x \| R_y) \| h(R_y) \| N_i) \oplus C_u$
$F_u = h(h(R_x \| R_y) \| N_i)$
$R_u = h(F_u \| C_u \| h(R_y)\ )$
$CID_u = h(R_u \| D_u \| h(R_y)) \oplus F_u$
$M_1 = h(P_{us} \| CID_u \| F_u)$

Then, the SC sends the login message $(CID_u, P_{us}, M_1, N_i)$ to the S_u through open channel.

4.3 Authentication Phase

Upon receiving the login message, S_u performs the following steps for authentication. The details of the authentication phase are given in Table 5.

Step 1: After obtaining the login message, S_u computes C_u, F_u, and R_u. Then, it validates the user by computing $h(P_{us} \| CID_u \| F_u) \stackrel{?}{=} M_1$, where $F_u = h(h(R_x \| R_y) \| N_i)$. If the condition is not satisfied, S_u terminates the session. Otherwise, it generates random number N_k and computes SK_{us}, Z_1, M_2 as follows.
$B_u = h(h(R_x \| R_y) \| (R_y)) \oplus C_u$
$SK_{us} = h(B_u \| SID_j \| R_u \| N_k)$
$Z_1 = N_k \oplus R_u$
$M_2 = h(P_{us} \| SK_{us} \| F_u \| N_k)$
Then S_u sends the message (Z_1, M_2) to the user through the public channel.

Step 2: U_r extracts N_k to compute SK_{us} for authentication of the server. First, the U_r compares $h(P_{us} \| SK_{us} \| F_u \| N_k \stackrel{?}{=} M_2)$. If this condition is not satisfied, then the user declines the session. Otherwise, the S_u will be successfully authenticated by the user.

Step 3: To complete the mutual authentication process, U_r computes $M_3 = h(SID_u \| F_u \| SK_{us} \| N_k)$ and sends it to the S_u through the public channel.

Step 4: Upon receiving the message M_3, S_u computes $h(SID_j \| F_u \| SK_{us} \| N_k)$ and checks whether it is equal to received message or not. If both are not equal, the server rejects the session. Otherwise, U_r is successfully authenticated by the server and the mutual authentication process is complete.

Step 5: Then both U_r and S_u compute the session key $SK_f = h(SID_j \| B_u \| SK_{us} \| F_u \| N_i \| N_k)$ for future communication.

Table 5 Authentication and session key agreement phase of proposed scheme

User(U_r)		Server(S_u)
		After getting the message Server checks for the validation of message, Computes
		$C_u = h(SID_u \| h(R_x \| R_y) \| h(R_y) \| N_i) \oplus P_{us}$
		$F_u = h(h(R_x \| R_y) \| N_i)$
		$R_u = h(F_u \| C_u \| h(R_y))$
		Checks $h(P_{us} \| CID_u \| F_u) \stackrel{?}{=} M_1$
		If it holds then, User is authenticated by server S_u.
		Server generates a random number N_k and computes the following equations
		$B_u = h(h(R_x \| R_y) \| h(R_y)) \oplus C_u$
		$SK_{us} = h(B_u \| SID_j \| R_u \| N_k)$
		$Z_1 = N_k \oplus R_u$
		$M_2 = h(P_{us} \| SK_{us} \| F_u \| N_k)$
	$\xleftarrow{(Z_1, M_2)}$ (*Publicchannel*)	
Computes $N_k = Z_1 \oplus R_u$		
$SK_{us} = h(B_u \| SID_j \| R_u \| N_k)$		
Checks $h(P_{us} \| SK_{us} \| F_u \| N_k) \stackrel{?}{=} M_2$		
Then compute M_3		
$M_3 = h(SID_j \| F_u \| SK_{us} \| N_k)$		
	$\xrightarrow{(M_3)}$ (*Publicchannel*)	
		Checks $h(SID_u \| F_u \| SK_{us} \| N_k) \stackrel{?}{=} M_3$
		If both are equal, then it will calculate the session key
	$\xleftrightarrow{SK_f = h(SID_j \| B_u \| SK_{us} \| F_u \| N_i \| N_k)}$	
		Stores the session key SK_f
Stores the session key SK_f		

4.4 Password Change Phase

In this phase, an authentic user can change the old password to new password as follows.

Step1: U_r inserts SC into card reader and inputs ID_u, PW_u and PW_{new}.

Step 2: SC calculates $b = P_u \oplus h(ID_u \| PW_u)$ and verifies the authenticity of the user by comparing $E_u \stackrel{?}{=} E_u^*$. If the condition does not satisfied, SC rejects the request. If satisfied, the user is allowed to input a new password PW_{new}.

Step 3: Finally SC calculates $PW_{new^*} = h(b \| PW_{new})$, $P_u^{new} = b \oplus h(ID_u \| PW_{new^*})$, $E_u^{new} = h(ID_u \| PW_u \| h(R_y))$, and replaces P_u, E_u with P_u^{new}, E_u^{new} respectively. The new password is successfully updated.

5 Security Analysis of the Scheme Using BAN Logic

To prove the security of the session key between the user and server, we have used the BAN logic, which is one of the most popular and widely used logic for analyzing the authentication protocols [36, 37]. It depicts beliefs of both user and server, which are involved in communication. Then, we demonstrate the security properties of the proposed scheme. We used the symbols P and Q are principals, X and Y range over statements and K ranges over the cryptographic key.

The notations of the BAN logic for the proposed scheme are as follows:

- $P \mid\equiv S_X$: P believes that S_X is true.
- $\#(S_X)$: S_X is fresh, which means S_X has not been used before.
- $P \Rightarrow S_X$: P has complete authority on S_X and P believes S_X.
- $P \triangleleft S_X$: Someone sends message containing S_X to P.
- $P \mid\sim S_X$: Sometime(may be long time ago or current time) P sent a message including S_X.
- $< S_X > S_Y$: S_X is concatenated with the secret formula S_Y.
- $(S_X)_h$: The formula S_X is hashed.
- $(S_X)_K$: The formula S_X is encrypted with key K.
- $P \overset{K}{\leftrightarrow} Q$: P and Q use K as secret key between them. Only P and Q know about the K and not others.
- SK: The session key between the user and server which is used for secure communication.

  Ban logic rules as follows:
- The message meaning rule

  $\frac{P \mid\equiv P \overset{K}{\leftrightarrow} Q, P \triangleleft (S_X)_K}{P \mid\equiv Q \mid\sim S_X}$

  P believes that P and Q shared the secret key K and P receives the message which is encrypted by K. P believes that, Q sometimes sent message including S_X.
- The nonce verification rule

  $\frac{P \mid\equiv S_X (S_X), P \mid\equiv Q \mid\sim X}{P \mid\equiv Q \mid\equiv S_X}$

  If P believes that S_X is fresh and Q sends the message containing S_X once, then P believes that Q believes S_X.
- The jurisdiction rule

  $\frac{P \mid\equiv Q \Rightarrow S_X, P \mid\equiv Q \mid\equiv S_X}{P \mid\equiv S_X}$

  P believes that Q has complete authority on S_X and P believes that Q believes S_X. P believes S_X is true.
- The freshness rule

  $\frac{P \mid\equiv \# S_X}{P \mid\equiv \#(S_X, S_Y)}$

  If P believes that S_X is fresh, then the P believes that (S_X, S_Y)must be fresh.
- The belief rule

  $\frac{P \mid\equiv Q \mid\equiv (S_X, S_Y)}{P \mid\equiv Q \mid\equiv (S_X)}$

  If P believes that the Q believes message S_X and S_Y, then P believes Q believes the message S_X.

According to BAN logic, the proposed scheme satisfies the following goals.

Goal 1: $U_r \mid\equiv U_r \overset{SK_{us}}{\leftrightarrow} S_u$

Goal 2: $U_r \mid\equiv S_u \mid\equiv U_r \overset{SK_{us}}{\leftrightarrow} S_u$

Goal 3: $S_u \mid\equiv U_r \overset{SK_{us}}{\leftrightarrow} S_u$

Goal 4: $S_u \mid\equiv U_r \mid\equiv U_r \overset{SK_{us}}{\leftrightarrow} S_u$

We transform our scheme to the idealized form as follows:

Message 1: $U_r \rightarrow S_u : <P_{us}, CID_u, M_1, N_i>_{U_r \overset{F_u}{\leftrightarrow} S_u}$

Message 2: $S_u \rightarrow U_r : <Z_1, M_2>_{U_r \overset{SK_{us}}{\leftrightarrow} S_u}$

Message 3: $U_r \rightarrow S_u : <M_3>_{U_r \overset{SK_{us}}{\leftrightarrow} S_u}$

Based on BAN logic, the following assumptions are taken.:

A_1 : $U_r \mid\equiv \#N_1$

A_2 : $S_u \mid\equiv \#N_2$

A_3 : $U_r \mid\equiv (U_r \overset{F_u}{\leftrightarrow} S_u)$

A_4 : $S_u \mid\equiv (U_r \overset{F_u}{\leftrightarrow} S_u)$

A_5 : $U_r \mid\equiv S_u \mid\Rightarrow (U_r \overset{SK_f}{\leftrightarrow} S_u)$

A_6 : $S_u \mid\equiv U_r \mid\Rightarrow (U_r \overset{SK_f}{\leftrightarrow} S_u)$

The SC generates the random number and computes some parameters. Then the U_r sends the message to the S.

Step 1: $S_u \triangleleft (P_{us}, CID_u, M_1, N_i)_{U_r \overset{F_u}{\leftrightarrow} S_u}$

According to Step 1, assumption A_3, by applying message meaning rule we obtain

Step 2: $S_u \mid\equiv U_r \mid\sim (U_r \overset{F_u}{\leftrightarrow} S_u, N_i, U_r \overset{h(SID_u \| h(R_y))}{\leftrightarrow} S_u)$

According to Step 2 and assumption A_1, by using freshness rule, Step 3 can be applied

Step 3: $S_u \mid\equiv \#(U_r \overset{F_u}{\leftrightarrow} S_u, N_i, U_r \overset{h(SID_u \| h(R_y))}{\leftrightarrow} S_u)$

According to Step 2 and Step 3, by applying the nonce-verification rule, we obtain Step 4

Step 4: $S_u \mid\equiv U_r \mid\equiv (U_r \overset{F_u}{\leftrightarrow} S_u, N_i, U_r \overset{h(SID_u \| h(R_y))}{\leftrightarrow} S_u)$

According to Step 4 and belief rule, we obtain Step 5 as follows

Step 5: $S_u \mid\equiv U_r \mid\equiv (U_r \overset{F_u}{\leftrightarrow} S_u)$

According to Step 5 and assumption A_4, by applying jurisdiction rule we get Step 6

Step 6: $S_u \mid\equiv (U_r \overset{F_u}{\leftrightarrow} S_u)$

According to Message 2, we find Step 7 as follows

Step 7: $U_r \lhd (P_{us}, U_r \overset{F_u}{\leftrightarrow} S_u, N_k)_{(U_r \overset{SK_{us}}{\leftrightarrow} S_u)}$

According to Step 7 and assumption A_5, by applying message meaning rule, Step 8 can be applied

Step 8: $U_r \mid\equiv S_u \mid\sim (P_{us}, U_r \overset{F_u}{\leftrightarrow} S_u, N_k)_{(U_r \overset{SK_{us}}{\leftrightarrow} S_u)}$

By Step 8 and assumption A_2, by applying the freshness rule, we can get

Step 9: $U_r \mid\equiv S_u \#(P_{us}, U_r \overset{F_u}{\leftrightarrow} S_u, N_k)_{(U_r \overset{SK_{us}}{\leftrightarrow} S_u)}$

By Step 8 and Step 9, applying the nonce verification rule to proceed to Step 10

Step 10: $U_r \mid\equiv S_u \mid\equiv (P_{us}, U_r \overset{F_u}{\leftrightarrow} S_u, N_k)_{(U_r \overset{SK_{us}}{\leftrightarrow} S_u)}$

According to Step 10 and belief rule, we obtained Step 11 as follows

Step 11: $U_r \mid\equiv S_u \mid\equiv (U_r \overset{SK_{us}}{\leftrightarrow} S_u)$ **(Goal-2)**

According to Step 11, assumption A_5, and the jurisdiction rule, we have Step 12 as follows

Step 12: $U_r \mid\equiv (S_u \overset{SK_{us}}{\leftrightarrow} S_u)$ **(Goal-1)**

According to message 3, Step 13 has been obtained

Step 13: $S_u \lhd (SID_u, U_r \overset{F_u}{\leftrightarrow} S_u, N_k)_{(U_r \overset{SK_{us}}{\leftrightarrow} S_u)}$

According to Step 13 and Assumption A_3, the message meaning rule has been applied

Step 14: $S_u \mid\equiv U_r \mid\sim (SID_u, U_r \overset{F_u}{\leftrightarrow} S_u, N_k)_{(U_r \overset{SK_{us}}{\leftrightarrow} S_u)}$

From Step 14, Assumption A_2 and freshness rule, we get Step 15 as follows

Step 15: $S_u \mid\equiv \#(SID_u, U_r \overset{F_u}{\leftrightarrow} S_u, N_k)_{(U_r \overset{SK_{us}}{\leftrightarrow} S_u)}$

According to Step 15, we apply the belief rule

Step 16: $S_u \mid\equiv U_r \mid\equiv (U_r \overset{SK_{us}}{\leftrightarrow} S_u)$ **(Goal-4)**

According to Step 16 and assumption A_4, jurisdiction rule has been applied to get Step 17

Step 17: $S_u \mid\equiv (U_r \overset{SK_{us}}{\leftrightarrow} S_u)$ **(Goal-3)**

We proved that the proposed scheme successfully accomplish all the goals. Both the server and user believe that they share the secure session key.

5.1 Security Analysis

In this section, we discuss the security analysis of the proposed scheme. The proposed scheme not only provides mutual authentication, user anonymity but also resist various attacks.

1. Mutual authentication
The proposed scheme achieves mutual authentication between user and server. Both the user and server are authenticated by each other. The login message $(CID_u, P_{us}, M_1, N_i)$ is sent from the user to server. Upon receiving the message, server extracts F_u, R_u and verifies the validity of M_1. If M_1 is true, then the user is a valid user. Then the server computes $M_2 = h(P_{us}\|SK_{us}\|F_u\|N_k))$ and sends it to the user. After receiving the message M_2 from the server, user checks for the legitimacy of the server. If it is successfully validated, then the user computes $M_3 = h(SID_u\|F_u\|SK_{us}\|N_k)$. Then sends it to the server to prove its legitimacy and to complete the mutual authentication process. When both user and server successfully authenticate each other, the session key $SK_f = h(SID_j\|B_u\|SK_{us}\|F_u\|N_i\|N_k)$ will be generated for secure communication.
2. User anonymity
In this scheme, the secrecy of user U_r is maintained by transmitting a session variant user identity ID_u. The information stored in smart card includes ID_u with secret key R_x and password PW_n . Moreover, the login message $(CID_u, P_{us}, M_1, N_i)$ does not contain any ID_u in plain text. However, to verify the guessed identity ID_u^* from the expression $C_u = h(h(R_x\|R_y)\|h(R_y)) \oplus B_u$, the secret key R_x and R_y is needed, which is impossible to extract. To extract the ID_u from the E_u, the adversary has to know the PW_n, in which the password PW_u is concatenated with the random number b. Hence, the proposed scheme achieves user anonymity.
3. Insider attack
It may happens that, the U_r can use same ID_u and password for different applications. If the password is revealed by RS or any privileged insider, then it leads to various security flaws. In the proposed scheme, U_r registers himself by sending $h(PW_u\|b)$ instead of PW_u in plain text. So RS or any privileged insider cannot extract password. Thus, the proposed scheme can withstand the insider attack.
4. Replay attack
To ensure the freshness of the message during the authentication phase, two methods are used, one is the time stamp and the other is random number. In the proposed scheme, two random numbers N_i and N_k has been generated to make the login message dynamic. Suppose the adversary (A_k) got hold of the login message $(CID_u, P_{us}, M_1, N_i)$ and attempts to respond the login message by sending (Z_1, M_2). However, he will not succeeded to generate a valid login message as $Z_1 = N_k \oplus R_u$, $R_u = h(F_u\|C_u\|h(R_y))$, $F_u = h(h(R_x\|R_y)\|N_i)$, and $C_u = h(SID_u\|h(R_x\|R_y)\|h(R_y)\|N_i) \oplus P_{us}$. Although random number N_i is known, still he can not compute F_u and C_u as master key $h(R_x\|R_y)$ is used. In addition, to compute $M_2 = h(P_{us}\|SK_{us}\|F_u\|N_k)$, he needs $SK_{us} = h(B_u\|SID_j\|R_u\|N_k)$. The authentication will definitely fail as A_k could not able to compute M_2. This proves that the proposed scheme can resist the replay attack.
5. Known-key security
The proposed scheme achieves known key security. Even if the session key (SK_f) is compromised, it would not reveal any information about other session keys. The SK_f is computed as $SK_f = h(SID_u\|SK_{us}\|B_u\|F_u\|N_i\|N_k)$, where F_u is not known to adversary. Also, if the smart card information C_u is leaked, he cannot extract B_u as $B_u = h(h(R_x\|R_y)\|h(R_y)) \oplus C_u$. And also N_i and N_k are two random numbers

generated by user and server respectively which are different for each session. So, the proposed scheme achieves known key secrecy.

6. Smart card loss attack
In case the smart card is lost or stolen, the attacker tries to extract the smart card information $(C_u, D_u, E_u, P_u, h(R_y), h(.))$, where $C_u = h(h(R_x \| R_y) \| h(R_y)) \oplus B_u$, $D_u = h(ID_u \| PW_n \| C_u) \oplus h(R_x \| R_y)$, $E_u = h(ID_u \| PW_n \| h(R_y))$. But, he can not succeeded to get ID_u and PW_u to login the system as they are protected through a one way hash function. Moreover, the adversary can not retrieve the master secret key R_x and secret number R_y as the ID_u and PW_u are not known. Therefore, the proposed scheme can resist smart card loss attack.
7. Offline password guessing attack
Suppose, the smart card has been lost or stolen, and the adversary can retrieve the information $(C_u, D_u, E_u, P_u, h(R_y), h(.))$ from the smart card. In addition, another assumption is an A_k has eavesdropped the login message $(CID_u, P_{us}, M_1, N_i)$ transmitted between the U_r and S_u. But the adversary cannot compute PW_n as it is protected by one-way hash function. To reveal PW_n, the adversary has to know PW_u and b which are provided only by the user. Therefore, the proposed scheme withstands offline password guessing attack.
8. User impersonation attack
The proposed scheme secure against the user impersonation attack due to the following reason.
Adversary A_k tries to impersonation the user by generating login message. However, it is impossible because, to compute CID_u, A_k has to know $h(R_x \| R_y)$, which is the secret key created by the server. To retrieve $h(R_x \| R_y)$, an adversary has the knowledge of ID_u, PW_u and random number b. So, A_k cannot impersonate the user during the login phase. In authentication phase, A_k may try to impersonate the message M_3. But, it is impossible without the knowledge of $SK_{us} = h(B_u \| SID_u \| R_u \| N_k)$. Again to calculate SK_{us} adversary needs B_u, R_u and random number N_k. So, the proposed scheme withstands user impersonation attack.
9. Server impersonation attack
For server impersonation attack, adversary has to generate (Z_1, M_2), where $M_2 = h(P_{us} \| SK_{us} \| F_u \| N_k)$. Even though attacker knows P_{us}, he needs B_u, R_u, F_u, N_k to calculate Z_1. In the proposed scheme, B_u is computed by using master key R_x and F_u calculated by using secret key $h(R_x \| R_y)$ generated by RS. Thus, server impersonation attack is not possible in the proposed protocol.
10. Forgery attack
The assumption is adversary knows the login message $(CID_u, P_{us}, M_1, N_i)$. He tries to forge the message as a legitimate user. But, to compute CID_u, he needs F_u and R_u, where $F_u = h(h(R_x \| R_y) \| N_i)$, and $R_u = h(F_u \| C_u \| h(R_y))$. He cannot compute the login request without knowing master key R_x, secret key $h(R_x \| R_y)$ and random number b. So, A_k cannot generate a valid login message. Hence, the proposed scheme can withstand the forgery attack.
11. Denial of service attack
To execute Denial of service attack, A_k has to submit correct user ID_u and PW_u to the smart card. But extraction of ID_u and PW_u is infeasible for the adversary, as these parameters are not sent in encrypted form. Also, these parameters are not sent

between the user and server during login and authentication phase. Therefore, the proposed scheme can resist denial of service attack.

12. Reparability
Reparability means the *RS* can issues a new smart card to a user in case of lost or stolen. The proposed scheme can revoke the smart card of a legal user if it is lost or stolen. The user has to request to *RS* with the same user ID_u and PW_n. The adversary can not get the user ID_u and PW_n as we have discussed in stolen smart card attack. Hence, the proposed scheme provides good reparability.
13. Password update
In this scheme, a user can freely choose his password and change it as needed. One can change the password only if he knows the old password. The intruder cannot change the password without knowing the valid login message and the old password.

6 Simulation Result Using AVISPA Tool

In this section, we provide the simulation of the proposed scheme using widely accepted AVISPA (Automated Validation of Internet Security Protocols and Applications) tool [38]. To validate internet security protocol, AVISPA is one of the prominent tools out of the all existing tools [39]. It is a modular and expressive formal language for defining the protocols and security properties.

All interaction with the simultaneous tool is made by passing the security problem (one that is any security property that the protocol is expected to achieve) in High-Level Protocol Specification Language (HLPSL) [40]. The user-defined security problem is automatically translated (via the HLPSL2IF Translator)into an analogous specification written in an Intermediate Format (IF) as shown in Fig. 1. IF specifications are input to the back-ends which implement different techniques to search the corresponding infinite-state transition. There are four back-ends that are On-the-fly Model-Checker(OFMC), CL-based Attack Searcher(CL-AtSe), SAT-based Model-checker(SATMC), and Tree-Automata-based Protocol-Analyser(TA4SP). Then, the back-end analyzed the protocol and generated the Output Format(OF).

6.1 Specifying the Scheme

In this section, we describe the specification of the proposed scheme using HLPSL language for the roles of user (U_r), server (S_u) and registration center (*RS*). Each role has some initial information by parameters, and then they can transmit parameters to others using a secure and public channel. The channel(dy) denotes the Dolev-Yao threat model [41] in which an intruder can eavesdrop and modify the message during communication. The user U_r (initiator of the proposed scheme), first receives the start signal and changes its state from 0 to 1. During the registration phase, the U_r will send the message (ID_u, PW_n) to the *RS* through a secure channel using snd() operation and symmetric key SK_{us}. Upon receiving the message, *RS* will issues a *SC* to the U_r containing the parameters $(C_u, D_u, E_u, h(R_y), h(.))$. The user will receive the *SC* by using rcv() operation and symmetric key SK_{us} in a secure channel. In login phase, U_r generates a random number N_i by using new() operation and send login message $(CID_u, P_{us}, M_1, N_1)$ to the server S_u through public channel. U_r receives the message (Z_1, M_2) from the server and finally send the message M_3 through the public channel.

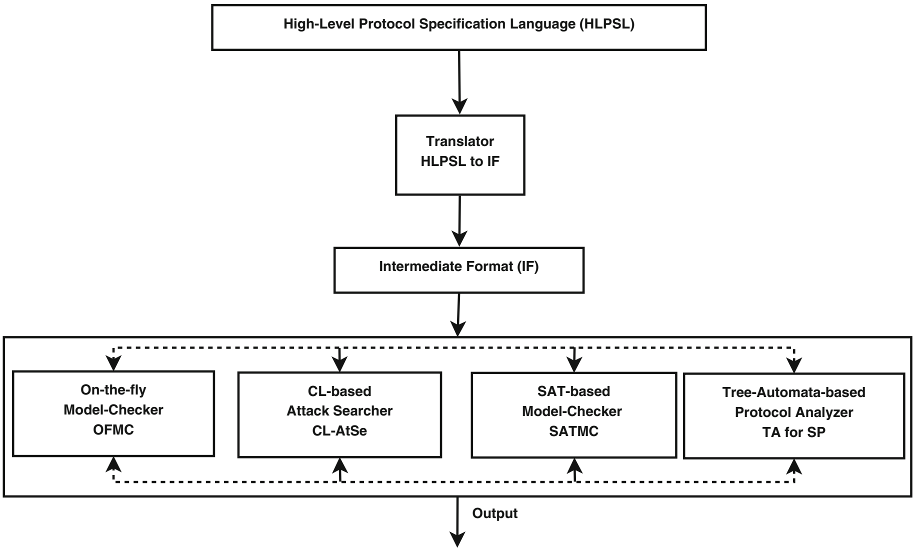


Fig. 1 The architecture of the AVISPA Tool

The role specification of the U_r, S_u, and RS of the proposed protocol is shown in Table 6, Table 7 and Table 8 respectively. During login phase, S_u will receive login message $(CID_u, P_{us}, M_1, N_i)$ and changes the state and response with the authentication message (Z_1, M_2). The specification of environment, session, and goal are also defined in Table 9 and Table 10. The basic role of the user, registration center, and server are described in session role. In goal section, various goals of the proposed protocol are defined. The authentication on user-server-n1 supposes that the S_u successfully authenticates random number N_i which generated by the U_r. The other authentication goals are defined in the same way.

We simulated the proposed algorithm under both, OFMC and AtSe back-ends. The simulation results are shown in Figs. 2a, b respectively. The output demonstrates that proposed protocol is secure.

7 Performance Evaluation

In this section, we have presented a comparison of computational cost, communication cost and security features between the proposed scheme and some other competent schemes.

We have used computational cost and communication cost as the evaluation metrics. The comparison of the computational cost of the proposed algorithm with other existing algorithms is presented in Table 11. To compare the results, we have used the notations $T_{EX}, T_{HS}, T_{MUL}, T_{EN}$ which denotes XOR operation, one-way hash function, multiplication function, and encryption function respectively. As evident from the table, most of the existing algorithms have total cost more than the proposed approach. Compared with Shunmuganathan et al. scheme [34], the computational overhead of proposed scheme costs

Table 6 Specification of user

% User U_r role specification in HLPSL
role user(U_r, RS, S_u : agent,
%symmetric key between U_r and RS
SK_{urc} : symmetric–key,
H : hash–func,
snd, rcv: channel(dy))
played_by U_r
def=
local State : nat,
ID_u, SID_j, PW_u, B, S_X, S_Y : text,
$A_u, B_u, C_u, D_u, E_u, F_u, N_1, N_2$: text,
CID_u, P_{us}, M_1, M_2, M_3 : text,
SK_{us} : text
const usr_ser_n1, ser_usr_n2,
sb1, sb2, sb3 : protocol _ id
init State := 0
transition
% User registration phase % U_r sends (IDi, PW_n) to RS via a secure channel
1. State = 0 /\ rcv(start) =|>
State' : = 2 /\ $snd(ID_u.H(B.PW_u)\_SK_{urc})$
/\ secret(PW_u, B, $sb1$, U_r)
/\ secret(ID_u, SID_u, $sb2$, U_r, RS, S_u)
2. State = 2 /\ rcv ($\{xor(H(H(S_X.S_Y).H(Y)),(ID_u.S_X))$.
$xor(H(ID_u.H(B.PW_u).xor(H(H(S_X.S_Y).H(S_Y)),(ID_u.S_X))),H(S_X.S_Y))$.
$H(ID_u.H(B.PW_u).H(S_Y)).H(S_Y).H\}\ \_SK_{urc}) =| >$
% Login phase
% U_r sends $(CID_u, P_{us}, M_1, N_1)$ to S_u via a public channel
State' := 4 /\ secret($S_X, S_Y, sb3, RS$)
/\ N_1':= new()
/\ $CID_u' := xor(H(((H(S_X.S_Y).N_1').xor(H(H(S_X.S_Y).H(S_Y)),(ID_u.S_X)).H(S_Y))$.
$xor(H(ID_u.H(B.PW_u).xor(H(H(S_X.S_Y).H(S_Y))$,
$(ID_u.S_X)),H(S_X.S_Y)).H(S_Y)),(H(S_X.S_Y).N_1')))$
/\ $P_{us}' := xor(H(SID_j.H(S_X.S_Y).H(S_Y).N_1),xor(H(H(S_X.S_Y).H(S_Y)),(ID_u.S_X)))$
/\ $F_u' := H(H(S_X.S_Y).N_1')$
/\ $M_1' := H(P_{us}'.CID_u'.F_u')$
/\ $snd(CID_u'.P_{us}'.M_1'.N_1')$
% U_r has freshly generated the value N_1' for S_u
/\ witness($U_r, S_u, usr\_ser\_n1, N_1'$)
% Verification phase
% U_r receives (Z_1, M_2) from S_u via a public channel
3.$State = 4$/\$rcv$ $(xor(N_2,(H(H(S_X.S_Y).N_1').xor(H(H(S_X.S_Y).H(S_Y)),(ID_u.S_X)).H(S_Y)))$.
$xor(H(SIDj.H(S_X.S_Y).H(S_Y).Ni).xor(H(H(S_X.S_Y).H(S_Y)),(IDi.S_X)))$.
$H(H(IDi.S_X).SIDj.((H(S_X.S_Y).Ni').xor(H(H(S_X.S_Y).H(S_Y)),(IDi.S_X)).H(S_Y)).Nj')$.
$(H(S_X.S_Y).Ni').Nj') =| >$
% U_r sends (M_3) to S_u via a public channel
$State' := 6$/\$M_3' := H(SID_j.H(H(S_X.S_Y).N_1').H(H(ID_u.S_X).SID_j.((H(S_X.S_Y).N_1')$.
$xor(H(H(S_X.S_Y).H(S_Y)),(ID_u.S_X)).H(S_Y)).N_2').N_2)$
/\ snd(M_3')
% U_r's acceptance of the value N_2 generated for U_r by S_u
/\ request($S_u, U_r, ser\_usr\_n2, N_2'$)
end role

Table 7 Specification of server

```
%Role specification in HLPSL for the Server
role user(U_r, RS, S_u : agent,
%symmetric key between U_r and RS
SK_urc : symmetric–key,
% H is hash function
H : hash–func,
snd, rcv: channel(dy))
played_by S_u
def=
local State : nat,
ID_u, SID_u, PW_u, B, S_X, S_Y : text,
A_u, B_u, C_u, D_u, E_u, F_u, N_1, N_2 : text,
CID_u, P_us, M_1, M_2, M_3 : text,
SK_us : text
const usr_ser_n1, ser_usr_n2,
sb1, sb2, sb3 : protocol _ id
init State := 0
transition
% Login phase
% S_u receives (CID_u, P_us, M_1, N_1) from U_r via a public channel
 1.State = 0/\rcv (xor(H(((H(S_X.S_Y).N1').xor(H(H(S_X.S_Y).H(S_Y)), (ID_u.S_X)).H(S_Y)).
                  xor(H(ID_u.H(B.PW_u).xor(H(H(S_X.S_Y).H(S_Y)), (ID_u.S_X)), H(S_X.S_Y)).
                  H(S_Y)), (H(S_X.S_Y).N1'))).xor(H(SID_u.H(S_X.S_Y).H(S_Y).N_1),
                  xor(H(H(S_X.S_Y).H(S_Y)), (ID_u.S_X))).H(P'_us.CID'_u.F'_u).N1') =|>
State' := 3 /\ secret(PW_u, B, sb1, U_r)
/\secret(ID_u, SID_u, sb2, U_r, RS, S_u)
/\secret(S_X, S_Y, sb3, RS)
/\N2' := new()
 /\SKus' := H(H(ID_u.S_X).SID_j.((H(S_X.S_Y).N1').
              xor(H(H(S_X.S_Y).H(S_Y)), (ID_u.S_X)).H(S_Y)).N2')
 /\M2' := H(xor(H(SID_u.H(S_X.S_Y).H(S_Y).N_1),
          xor(H(H(S_X.S_Y).H(S_Y)), (ID_u.S_X))).
          SK'_us.H(H(S_X.S_Y).N1').N2')
/\Z1' := xor(N_2, (H(H(S_X.S_Y).N1').xor(H(H(S_X.S_Y).H(S_Y)), (ID_u.S_X)).H(S_Y)))
% S_u sends (Z_1, M_2) to U_r via a public channel
/\snd(Z1', M2')
% S_u has freshly generated the value N_2' for U_r
/\witness(S_u, U_r, ser_usr_n2, N2')
% S_u receives (M_3) from U_r via a public channel
 2.State = 3/\rcv (H(SID_j.H(H(S_X.S_Y).N1').H(H(ID_u.S_X).
                  SID_j.((H(S_X.S_Y).N1').xor(H(H(S_X.S_Y).H(S_Y)),
                  (ID_u.S_X)).H(S_Y)).N2').N_2)) =|>
% S_u's acceptance of the value N_1 generated for S_u by U_r
State' := 5 /\request(U_r, S_u, usr_ser_n1, N1')
end role
```

Table 8 Specification of RS

% Registration Center(RS) role specification in HLPSL
role user(U_r, RS, S_u : agent,
%symmetric key between U_r and RS
SK_{urc} : symmetric– key,
% H is hash function
H : hash– func,
snd, rcv: channel(dy))
played_by RS
def=
local State : nat,
$ID_u, SID_u, PW_u, B, S_X, S_Y$: text,
A_u, B_u, C_u, D_u, E_u, F_u, N_1, N_2 : text,
$CID_u, P_{us}, M_1, M_2, M_3$: text,
SK_{us} : text
const usr_ser_n1, ser_usr_n2,
sb1, sb2, sb3 : protocol _ id
init State := 0
transition
% RS receives (ID_u, PW_n) from U_r via a secure channel
1. State = 0 /\ $rcv(ID_u.H(xor(ID_u, xor(B, PW_u)))_S K_{urc}) =\|\ >$
State' := 1 /\ secret$(PW_u, B, sb1, U_r)$
$/\backslash secret(ID_u, SID_u, sb2, U_r, RS, S_u)$
% RS sends ($smartcard$) to U_r via a secure channel
$/\backslash PW'_u := H(B.PW_u)$
$/\backslash B'_u := H(ID'_u.S_X)$
$/\backslash C'_u := xor(H(H(S_X.S_Y).H(S_Y)), (ID_u.S_X))$
$/\backslash D'_u := xor(H(ID_u.H(B.PW_u).xor(H(H(S_X.S_Y).H(S_Y)), (ID_u.S_X)), H(S_X.S_Y)))$
$/\backslash E'_u := H(ID_u.H(B.PW_u).H(S_Y))$
$/\backslash snd(C'_u.D'_u.E'_u.H(Y).H\_SK_{urc})$
$/\backslash secret(S_X, S_Y, sb3, RS)$
end role

a little more to offer more security, such as replay attack, known-key security, smart card lost attack, user impersonation attack, forgery attack, and denial of service attack.

In Table 12, we have compared the communicational cost of the proposed scheme with the existing competent schemes. The total number of message exchange in Sood et al. scheme, Li et al. scheme, Zhao et al. scheme, and Xue et al. scheme is four. Lee et al. scheme, Li et al. scheme, Shunmuganathan et al. scheme, and Jingarala et al. scheme requires less number of message exchange. The proposed scheme also requires 3 message exchanges during the communication.

Table 13 shows the security comparison of the proposed scheme with other related schemes. Our scheme provides protection against insider attack, replay attack, smart card loss attack, user impersonation attack, server impersonation attack, forgery attack, and

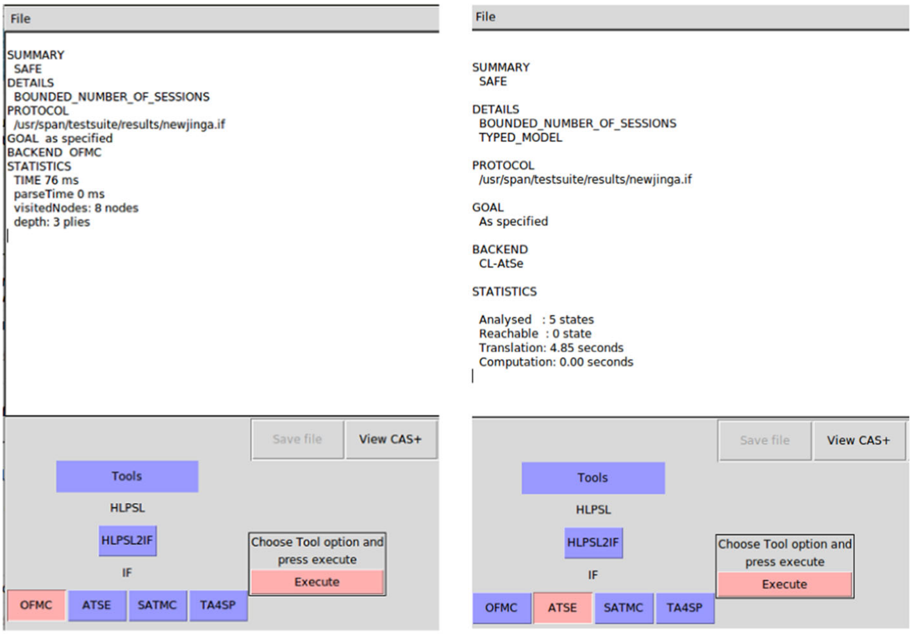
Table 9 Specification in HLPSL for the environment

```
%Role specification in HLPSL for role environment
const usr_ser_n1, ser_usr_n2,
sb1, sb2, sb3 : protocol_id
init State := 0
transition
role environment()
def=
const u_j, rs, s_i: agent,
sk_urc : symmetric_key,
h : hash_func,
usr_ser_n1, ser_usr_n2,
sb1, sb2, sb3 : protocol_id
intruder_knowledge = u_j, rs, s_i, h
composition
session (u_j, rs, s_i, sk_urc, h)
/\ session (i, rs, s_i, sk_urc, h)
/\ session (u_j, i, s_i, sk_urc, h)
/\ session (u_j, rs, i, sk_urc, h)
end role
goal
secrecy_of sb1
secrecy_of sb2
secrecy_of sb3
authentication_on usr_ser_n1
authentication_on ser_usr_n2
end goal
environment()
```

Table 10 Specification in HLPSL for the session, and goal

```
%Role specification in HLPSL for the session, and goal
role session (U_r, RS, S_u : agent,
% symmetric key between U_r and RS
SK_urc : symmetric_key,
% H is one-way hash function
H : hash_func)
def= local SN_1, SN_2, SN_3, RV_1, RV_2, RV_3 : channel (dy)
composition user(U_r, RS, S_u, SK_urc, H, SN_1, RV_1)
/\rs(U_r, RS, S_u, SK_urc, H, SN_2, RV_2)
/\server(U_r, RS, S_u, SK_urc, H, SN_3, RV_3)
end role
```

denial of service attack. Also, it provides security features such as user anonymity and mutual authentication. The proposed scheme is considerably more secure as compared to the existing state of art approaches.



(a) (b)

Fig. 2 **a** Result using OFMC backend. **b** Result using ATSE backend

Table 11 Computational cost analysis of scheme

Scheme	Log in phase	Authentication phase	Total cost
Lee et al. [25]	$7T_{HS}+4T_{EX}$	$8T_{HS}+4T_{EX}$	$15T_{HS}+8T_{EX}$
Sood et al. [26]	$5T_{HS}+9T_{EX}$	$25T_{HS}+18T_{EX}$	$30T_{HS}+27T_{EX}$
Li et al. [27]	$6T_{HS}+4T_{EX}$	$25T_{HS}+23T_{EX}$	$31T_{HS}+27T_{EX}$
Li et al. [28]	$7T_{HS}+4T_{EX}$	$11T_{HS}+8T_{EX}$	$18T_{HS}+12T_{EX}$
Zhao et al. [29]	$5T_{HS}+1T_{EX}+8T_{MUL}$	$5T_{HS}+4T_{MUL}$	$10T_{HS}+1T_{EX}+12T_{MUL}$
Xue et al. [30]	$6T_{HS}+5T_{EX}$	$19T_{HS}+19T_{EX}$	$25T_{HS}+24T_{EX}$
Shunmuganathan et al. [34]	$7T_{HS}+3T_{EX}$	$10T_{HS}+7T_{EX}$	$17T_{HS}+10T_{EX}$
Jingarala et al. [35]	$8T_{HS}+6T_{EX}$	$14T_{HS}+6T_{EX}$	$22T_{HS}+12T_{EX}$
Proposed scheme	$8T_{HS}+6T_{EX}$	$12T_{HS}+4T_{EX}$	$20T_{HS}+9T_{EX}$

Note: $S1$—user anonymity, $S2$—insider attack, $S3$—replay attack, $S4$—known-key security, $S5$—smart card loss attack, $S6$—offline password guessing attack, $S7$—user impersonation attack, $S8$—server impersonation attack, $S9$—forgery attack, $S10$—denial of service attack, $S11$—mutual authentication, $S12$—repairability, $S13$—password update.

Yes = Prevent the attack, *No* = Not prevent the attack

Table 12 Communicational cost analysis of the proposed scheme

SCHEMES	Message transferred during login and authentication phase
Lee et al. [25]	3
Sood et al. [26]	4
Li et al. [27]	4
Li et al. [28]	3
Zhao et al. [29]	4
Xue et al. [30]	4
Shummuganathat et al. [34]	3
Jingarala et al. [35]	3
Proposed scheme	3

Table 13 Security comparison of schemes

Scheme	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13
Lee et al. [25]	Yes	Yes	Yes	No	No	No	No	No	No	No	No	No	No
Sood et al. [26]	Yes	No	No	No	No	Yes	No	No	No	No	Yes	No	Yes
Li et al. [27]	Yes	No	No	No	Yes	No	Yes	Yes	No	No	Yes	No	Yes
Li et al. [28]	Yes	Yes	Yes	Yes	No	No	No	No	No	No	No	No	Yes
Zhao et al. [29]	Yes	Yes	Yes	No	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
Xue et al. [30]	Yes	Yes	Yes	No	Yes	No	No	No	Yes	Yes	Yes	No	Yes
Shummuganathat et al. [34]	Yes	Yes	No	No	No	Yes	No	Yes	No	No	Yes	No	Yes
Jingarala et al. [35]	Yes	Yes	No	Yes	Yes	Yes	No	Yes	No	Yes	No	Yes	Yes
Proposed scheme	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

8 Conclusion

In this paper, we have discussed Jangirala1 et al.'s scheme and pointed out some security issues, such as forgery attack, replay attack, user impersonation attack, and lack of proper mutual authentication. To compensate these security issues, we have proposed an enhanced scheme for the multi-server environment. Our proposed protocol satisfies all the essential security requirements along with mutual authentication, and session key agreement. Also, the comparison of proposed scheme with other schemes has been done which demonstrate that the proposed protocol has less computational and communicational cost. Also, we have simulated the proposed scheme using widely accepted AVISPA tool and proves mutual authentication through BAN logic. Moreover, the proposed protocol is user friendly and offers good repairability.

References

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

2. Hwang, T., Chen, Y., & Laih, C. J. (1990) Non-interactive password authentications without password tables. In *1990 IEEE region 10 conference on computer and communication systems*, 1990, IEEE TENCON'90 (pp. 429–431). IEEE.
3. Yang, W.-H., & Shieh, S.-P. (1999). Password authentication schemes with smart cards. *Computers & Security*, *18*(8), 727–733.
4. Hwang, M.-S., & Li, L.-H. (2000). A new remote user authentication scheme using smart cards. *IEEE Transactions on Consumer Electronics*, *46*(1), 28–30.
5. Chan, C.-K., & Cheng, L.-M. (2000). Cryptanalysis of a remote user authentication scheme using smart cards. *IEEE Transactions on Consumer Electronics*, *46*(4), 992–993.
6. Sun, H.-M. (2000). An efficient remote use authentication scheme using smart cards. *IEEE Transactions on Consumer Electronics*, *46*(4), 958–961.
7. Chien, H.-Y., Jan, J.-K., & Tseng, Y.-M. (2002). An efficient and practical solution to remote authentication: Smart card. *Computers & Security*, *21*(4), 372–375.
8. Wu, S.-T., & Chieu, B.-C. (2003). A user friendly remote authentication scheme with smart cards. *Computers & Security*, *22*(6), 547–550.
9. Ku, W.-C., & Chen, S.-M. (2004). Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards. *IEEE Transactions on Consumer Electronics*, *50*(1), 204–207.
10. Yoon, E.-J., Ryu, E.-K., & Yoo, K.-Y. (2004). Further improvement of an efficient password based remote user authentication scheme using smart cards. *IEEE Transactions on Consumer Electronics*, *50*(2), 612–614.
11. Lu, R., & Cao, Z. (2005). Efficient remote user authentication scheme using smart card. *Computer Networks*, *49*(4), 535–540.
12. Lee, S.-W., Kim, H.-S., & Yoo, K.-Y. (2005). Improvement of chien et al'.s remote user authentication scheme using smart cards. *Computer Standards & Interfaces*, *27*(2), 181–183.
13. Lee, N.-Y., & Chiu, Y.-C. (2005). Improved remote authentication scheme with smart card. *Computer Standards & Interfaces*, *27*(2), 177–180.
14. Xu, J., Zhu, W.-T., & Feng, D.-G. (2009). An improved smart card based password authentication scheme with provable security. *Computer Standards & Interfaces*, *31*(4), 723–728.
15. Amin, R., & Biswas, G. P. (2015). Cryptanalysis and design of a three-party authenticated key exchange protocol using smart card. *Arabian Journal for Science and Engineering*, *40*(11), 3135–3149.
16. Li, L.-H., Lin, L.-C., & Hwang, M.-S. (2001). A remote password authentication scheme for multiserver architecture using neural networks. *IEEE Transactions on Neural Networks*, *12*(6), 1498–1504.
17. Lin, I.-C., Hwang, M.-S., & Li, L.-H. (2003). A new remote user authentication scheme for multi-server architecture. *Future Generation Computer Systems*, *19*(1), 13–22.
18. Juang, W.-S. (2004). Efficient multi-server password authenticated key agreement using smart cards. *IEEE Transactions on Consumer Electronics*, *50*(1), 251–255.
19. Chang, C.-C., & Lee, J.-S. (2004). An efficient and secure multi-server password authentication scheme using smart cards. In *2004 international conference on cyberworlds* (pp. 417–422). IEEE.
20. Tsaur, W.-J., Chia-Chun, W., & Lee, W.-B. (2004). A smart card-based remote scheme for password authentication in multi-server internet services. *Computer Standards & Interfaces*, *27*(1), 39–51.
21. Yang, Y., Deng, R. H., & Bao, F. (2006). A practical password-based two-server authentication and key exchange system. *IEEE Transactions on Dependable and Secure Computing*, *3*(2), 105–114.
22. Tsai, J.-L. (2008). Efficient multi-server authentication scheme based on one-way hash function without verification table. *Computers & Security*, *27*(3), 115–121.
23. Liao, Y.-P., & Wang, S.-S. (2009). A secure dynamic id based remote user authentication scheme for multi-server environment. *Computer Standards & Interfaces*, *31*(1), 24–29.
24. Hsiang, H.-C., & Shih, W.-K. (2009). Improvement of the secure dynamic id based remote user authentication scheme for multi-server environment. *Computer Standards & Interfaces*, *31*(6), 1118–1123.
25. Lee, C.-C., Lin, T.-H., & Chang, R.-X. (2011). A secure dynamic id based remote user authentication scheme for multi-server environment using smart cards. *Expert Systems with Applications*, *38*(11), 13863–13870.
26. Sood, S. K., Sarje, A. K., & Singh, K. (2011). A secure dynamic identity based authentication protocol for multi-server architecture. *Journal of Network and Computer Applications*, *34*(2), 609–618.
27. Li, X., Xiong, Y., Ma, J., & Wang, W. (2012). An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards. *Journal of Network and Computer Applications*, *35*(2), 763–769.

28. Li, X., Ma, J., Wang, W., Xiong, Y., & Zhang, J. (2013). A novel smart card and dynamic id based remote user authentication scheme for multi-server environments. *Mathematical and Computer Modelling*, *58*(1), 85–95.
29. Zhao, D., Peng, H., Li, S., & Yang, Y. (2013). An efficient dynamic id based remote user authentication scheme using self-certified public keys for multi-server environment. arXiv preprint arXiv:1305.6350.
30. Xue, K., Hong, P., & Ma, C. (2014). A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture. *Journal of Computer and System Sciences*, *80*(1), 195–206.
31. Das, A. K. (2015). A secure and efficient user anonymity-preserving three-factor authentication protocol for large-scale distributed wireless sensor networks. *Wireless Personal Communications*, *82*(3), 1377–1404.
32. Li, X., Niu, J., Kumari, S., Liao, J., & Liang, W. (2015). An enhancement of a smart card authentication scheme for multi-server architecture. *Wireless Personal Communications*, *80*(1), 175–192.
33. Odelu, V., Das, A. K., & Goswami, A. (2015). An effective and robust secure remote user authenticated key agreement scheme using smart cards in wireless communication systems. *Wireless Personal Communications*, *84*(4), 2571–2598.
34. Shunmuganathan, S., Saravanan, R. D., & Palanichamy, Y. (2015). Secure and efficient smart-card-based remote user authentication scheme for multiserver environment. *Canadian Journal of Electrical and Computer Engineering*, *38*(1), 20–30.
35. Jangirala, S., Mukhopadhyay, S., & Das, A. K. (2017). A multi-server environment with secure and efficient remote user authentication scheme based on dynamic ID using smart cards. *Wireless Personal Communications*, *95*(3), 2735–2767.
36. Burrows, M., Abadi, M., & Needham, R. M. (1989). A logic of authentication. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering*, *426*, 233–271.
37. Ali, R., & Pal, A. K. (2017). Three-factor-based confidentiality-preserving remote user authentication scheme in multi-server environment. *Arabian Journal for Science and Engineering*, *42*(8), 3655–3672.
38. AVISPA Automated Validation of Internet Security Protocols and Applications. http://www.avispa-project.org/ (2015).
39. Viganò, L. (2006). Automated security protocol analysis with the AVISPA tool. *Electronic Notes in Theoretical Computer Science*, *155*, 61–86.
40. Armando, A., Basin, D., Boichut, Y., Chevalier, Y., Compagna, L., Cuéllar, J., et al. (2005). The AVISPA tool for the automated validation of internet security protocols and applications. In *International conference on computer aided verification* (pp. 281–285). Springer.
41. Dolev, D., & Yao, A. (1983). On the security of public key protocols. *IEEE Transactions on Information Theory*, *29*(2), 198–208.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Shreeya Swagatika Sahoo received M.Tech. degree in Computer Science and Engineering from Veer Surendra Sai University, Burla in 2015. Currently, she is pursuing Ph.D. in the Department of CSE at NIT, Rourkela, India. Her current research interests include cryptography, security protocol and network security.

Sujata Mohanty received Ph.D. degree in Computer Science and Engineering from NIT Rourkela, India. She is currently working as an Assistant Professor in the Department of Computer Science and Engineering at National Institute of Technology, Rourkela, India. Her research interests include information security, cryptography, and network security.

Banshidhar Majhi received his Ph.D. degree from Sambalpur University, Odisha, India, in 2001. He is currently working as a Professor in the Department of Computer Science and Engineering at National Institute of Technology, Rourkela, India. His field of interests include image processing, data compression, cryptography and security, parallel computing, soft computing, and biometrics. He is a professional member of MIEEE, FIETE, LMCSI, IUPRAI, and FIE. He has authored more than hundred papers in journals and conferences of international repute.