Sādhanā (2019)44:192
https://doi.org/10.1007/s12046-019-1163-4

Unified and integrated authentication and key agreement scheme for e-governance system without verification table

DARPAN ANAND[1,2,*] and VINEETA KHEMCHANDANI[3]

[1]Department of Computer Science and Engineering, University Institute of Engineering, Chandigarh University, Chandigarh, India
[2]Dr. A.P.J. Abdul Kalam University, Lucknow, India
[3]J.S.S. Academy of Technical Education, Noida, India
e-mail: darpan.e8545@cumail.in; darpan.anand.agra@gmail.com; vkhemchandani@jssaten.ac.in

MS received 14 November 2017; revised 26 March 2019; accepted 21 May 2019

Abstract. E-governance or electronic governance is an application of Information and Communication Technology (ICT) for delivering cost-effective government services by any country to its citizens with reliability, transparency and efficiency. Majority of authentication schemes for e-governance in India are based on single-server environment. To access the services, users need to register themselves at the authentication server for every e-governance service. Various e-governance services work through different servers, and therefore users get registered on each server separately. These services and servers require a unified and integrated authentication scheme to overcome the problem of multiple registrations and login processes. This paper proposes a dynamic authentication protocol based on the identity of a user for multi-server architecture without using verification tables. It is also capable of integrating all the existing e-governance projects. The proposed protocol fulfills the security requirements such as mutual authentication, traceability and identity protection along with the facility to share a session key among all the servers for secure communication.

Keywords. Communication; computer security; network security; authentication; authentication protocol; e-governance.

1. Introduction

Today, internet has become an integral part of our life. We are surrounded by Information and Communication Technology (ICT). The expeditious burgeoning of technology and internet is leading to Internet of Things (IoT). Most of the users' services are now available online. Authentication plays an important role to provides accessibility to these internet-based services only to the legitimate users.

Many researchers presented different authentication protocols both for two-layer as well as for multi-layer architecture-based systems. The authentication schemes for multi-server architecture are available in the literature [1–4]. It has been observed that the hash-based authentication schemes are most efficient techniques [1, 5–8]. In 2014, Xue *et al* proposed a technique [9], which claims the anonymity and traceability with all necessary security properties as in the Li *et al* [3] protocol. Gaharana and Anand [10] presented a security analysis of various multi-server authentication techniques. These techniques are based on two-way as well as on three-way factor-based authentication [11–16]. Generally, authentication schemes are dependent on a central server that stores the verification data. Because of centrally stored verification data, these schemes are vulnerable. Therefore, a new authentication scheme is required to overcome this weakness.

Authentication of multi-layer systems requires registration of all layers at a central layer, which acts as an authentication server. This type of schemes should be dynamic in nature, in which the parameters are calculated instantaneously at various layers and should not be based on central verification table. The authentication and session key will be calculated using the data stored at each layer at the time of registration and also depend on the instantaneous calculation of shared parameters based on user's identity.

This paper proposes a dynamic authentication scheme for multi-server environment. The proposed scheme is based on user's ID and does not require verification table. In this scheme, session key is deduced from the parameters derived from user's identity. These parameters are shared during authentication/login process. This scheme is corroborated with the case study of an e-governance system where all the services are available in different servers and requires separate registration of user at each server for its service. Therefore, the proposed scheme is able to integrate these isolated systems and gives an unified view to it as an

*For correspondence
Published online: 08 August 2019

integrated system. The proposed schema is also resistive for known security attacks such as leak-of-verifier attack, eavesdropping attack, stolen smart card attack, denial-of-service (DoS) attack, replay attack, forgery attack, etc.

2. Motivation of the work: unified and integrated authentication and key agreement scheme for multi-server systems

The current Indian e-governance system users can access various government services through an ICT-based system. These services are accessed by citizens, business organizations, government agencies and nonprofit organisations in different cases as Government to Citizen [17], Government to Business [18], Government to Government [19] and Government to Non-Profit [20], respectively. However, these services are deployed on various servers and users need to register on each server separately to access these services. Therefore, there is a requirement of a strong, integrated and unified authentication scheme to verify the lawfulness of the citizens to access various government services using a single registration process. To establish the need of this proposed authentication scheme, e-Pramaan Framework developed by Department of Electronics and Information Technology, Ministry of Communications and Information Technology, Government of India, has been studied and analysed [21–24]. It has been identified that the current system requires multiple registrations and authentications to access various government services and most of these authentication schemes are dependent on verifiable password or credentials stored at a central server [25–28]. These schemes are generally implemented for single-server environment or central-server environment. The password-based authentication schemes are required to store verifiable data in the form of a database table containing user identities and passwords. The multi-server-based system like e-governance is different because it is based on multi-server architecture; hence, password-based authentication schemes are not suitable. Therefore, there is a requirement of a strong and secure authentication scheme that will overcome the gaps and weaknesses of the existing authentication schemes.

3. Related work

This section gives details of some authentication techniques based on multi-server environment. To explain the working of these schemes, some notations used are listed in table 1.

3.1 *Lee's authentication protocol [8]*

The working of this scheme is shown in figure 1. There are three layers involved in this scheme: user (U_s), service providing server (S_j) and registration server (RS). Users

Table 1. Notations used to explain various protocols.

Symbol	Description
UID	User ID
PWD	User password
$h()$	One-way hash function
$\oplus$	Bitwise XOR computation
$\|\|$	Concatenation operation
y	Secret value of server(to be stored on smart card)
x	Secret value of server

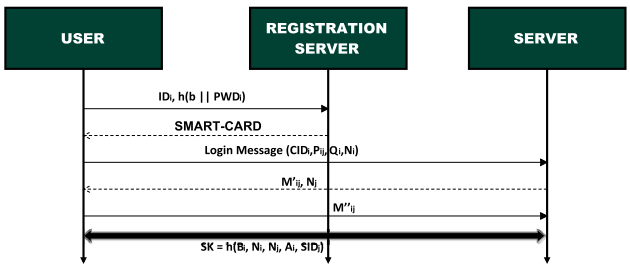

Figure 1. Sequence flow of Lee *et al*'s authentication protocol.

choose a random number b and password PWD_i. They send the message $(ID_i \oplus h(b, PWD_i))$ to RS. RS chooses the master key x and a secret number y to compute $h(x \| y)$ and $h(y)$. Thereafter, RS shares these parameters with S_j through a secure channel. Only RS knows the master secret key x and secret number y. The smart card parameters CID_i, P_{ij}, B_i and Q_{ij} are calculated at user's end with its password PWD_i, random number b, time-stamp T_i and nonce N_i:

$$A_i = h(b \oplus PWD_i), \tag{1}$$

$$CID_i = h(b \oplus PWD_i) \oplus h(T_i \| A_i \| N_i)), \tag{2}$$

$$P_{ij} = T_i \oplus h(h(y) \| N_i \| SID_j), \tag{3}$$

$$B_i = h(h(b \oplus PWD_i) \| h(x \| y)), \tag{4}$$

$$Q_{ij} = h(B_i \| A_i \| N_i). \tag{5}$$

Now, at RS's end, the parameter $M_{ij}^{'}$ is calculated on the basis of received parameters CID_i, P_{ij}, B_i and Q_{ij}:

$$M_{ij}^{'} = h(B_i \| N_i \| A_i \| SID_j). \tag{6}$$

$M_{ij}^{'}$ is sent to user's end and the parameter $M_{ij}^{''}$ is calculated and shared with the server:

$$M_{ij}^{''} = h(B_i \| N_j \| A_i \| SID_j). \tag{7}$$

CID_i is the parameter that will be saved on smart card along with others, T_i is the time-stamp and SID_j is server's ID;

$P_{ij}, Q_{ij}, M'_{ij}, M''_{ij}$ and A_i are the intermediate parameters used for authentication; N_i , N_j are nonces generated at user (U_s) and server (RS) side, respectively [8]. Finally, the session key, i.e. SK, will be generated at both the ends:

$$SK = h(B_i, N_i, N_j, A_i, SID_j). \tag{8}$$

3.2 *Scheme of Li et al [29]*

The detailed information flow of this scheme is illustrated in figure 2. This scheme works for three participants: the user (U_i), the service providing server (S_j) and registration server (RS). RS chooses the master key x and a secret number y to compute $h(x \| y)$ and $h(SID_j \| h(y))$ (where the SID_j is server's ID, requested by user to get access to the services from it). Later, RS shares these parameters with server S_j through a secure channel. The login message consists of the parameters CID_i, P_{ij}, M_1 and M_2 and the calculation of these parameters is as follows:

$$P_{ij} = E_i \oplus h(h(SID_j \| h(y)) \| N_i), \tag{9}$$

$$CID_i = A_i \oplus h(D_i \| SID_j \| N_i), \tag{10}$$

$$M_1 = h(P_{ij} \| CID_i \| D_i \| N_i), \tag{11}$$

$$M_2 = h(SID_j \| h(y)) \oplus N_i. \tag{12}$$

On the basis of the received parameters, server calculates M_3 and M_4 and later sends back to user. The calculation of these parameters is as follows:

$$M_3 = h(D_i \| A_i \| N_j \| SID_j), \tag{13}$$

$$M_4 = A_i \oplus N_i \oplus N_j. \tag{14}$$

User calculates M_5 on the basis of received and existing parameters:

$$M_5 = h(D_i \| N_i \| A_i \| SID_j). \tag{15}$$

CID_i is saved on smart card along with other required parameters. SID_j is server's ID; $P_{ij}, Q_{ij}, M_1, M_2, M_3, M_4, M_5, D_i$ and A_i are the intermediate parameters used for authentication. N_i and N_j are nonces generated at user and server side, respectively [29].

Finally, the session key, i.e. SK, will be generated at both the ends.

$$SK = h(D_i, A_i, N_j, N_i, SID_j). \tag{16}$$

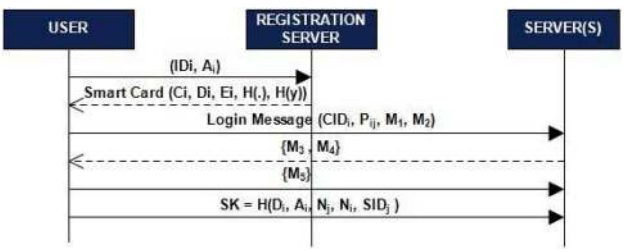

Figure 2. Sequence flow of Li *et al*'s authentication protocol.

3.3 *Scheme of Xue et al [30]*

This protocol also uses three participants, which are involved in authentication process as user (U_i), the service providing server (S_j) and control server (CS). The working and flow of information is shown in figure 3. User chooses a random number b and it's ID, ID_i. PID_i is the protected pseudonym identity of the user, which is calculated as $h(ID_i \| b)$ and B_i is calculated as $h(PID_i \| x)$, where x is the secret number known only to CS. TS_i is the current time-stamp value at U_i. N_{i1} is nonce at user's layer, N_{i2} is nonce at S_j layer and N_{i3} is nonce at CS server's end. SID_j is server's ID of service providing server and rest are the intermediate parameters used for calculations.

For login, user calculates F_i, P_{ij}, CID_i and G_i on the basis of parameters stored in their own smart card:

$$F_i = B_i \oplus N_{i1}, \tag{17}$$

$$P_{ij} = h(B_i \oplus h(N_i 1 \| SID_j \| PID_i \| TS_i), \tag{18}$$

$$CID_i = ID_i \oplus h(B_i \| N_{i1} \| TS_i \| \text{``00''}), \tag{19}$$

$$G_i = b \oplus h(B_i \| N_{i1} \| TS_i \| \text{``11''}). \tag{20}$$

Now, service provider server at S_j layer receives the login parameters and calculates J_i, K_i, L_i, M_i and sends to control server along with the received parameters from user's layer:

$$J_i = BS_j \oplus N_{i2}, \tag{21}$$

$$K_i = h(N_{i2} \| BS_j \| P_{ij} \| TS_i), \tag{22}$$

$$L_i = SID_j \oplus h(BS_j \| N_{i2} \| TS_i \| \text{``00''}), \tag{23}$$

$$M_i = d \oplus h(BS_j \| N_{i2} \| TS_i \| \text{``11''}). \tag{24}$$

Control server verifies the user and sends parameters P_i, R_i, Q_i, and V_i to the service provider server:

$$P_i = N_{i1} \oplus Ni3 \oplus h(SID_j \| N_{i2} \| BS_j), \tag{25}$$

$$R_i = N_{i2} \oplus N_{i3} \oplus h(ID_i \| N_{i1} \| B_i), \tag{26}$$

$$Q_i = h(N_{i1} \oplus N_{i3}), \tag{27}$$

$$V_i = h(N_{i2} \oplus N_{i3}). \tag{28}$$

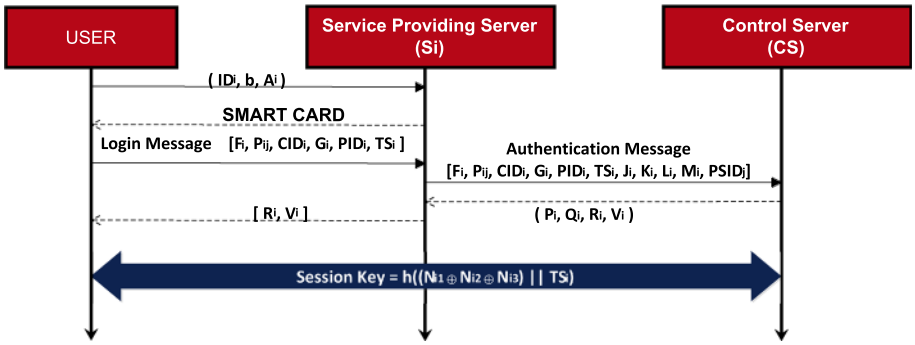

Figure 3. Sequence flow of Xue *et al*'s authentication protocol.

CID_i and PID_i are the parameters saved on smart card along with others. SID_j is server's ID; $P_{ij}, G_i, J_i, K_i, L_i, M_i, R_i, Q_i, V_i$ and BS_j are the intermediate parameters used for authentication. N_{i1} , N_{i2} and N_{i3} are nonces generated at user, service providing server and central server sides, respectively. TS_i is the time-stamp generated at the end of user at the time of login [30].

Finally, the session key, i.e. SK, will be generated at both the ends as follows:

$$SK = h((N_{i1} \oplus N_{i2} \oplus N_{i3}) \parallel TS_i). \tag{29}$$

3.4 *Scheme of Leu and Hsieh [31]*

This scheme uses a random number, which makes it difficult for illegitimate user to access the system. This random number is used to verify the identity. Three participants, the user (U_i), the service providing server (S) and registration server (RS), are involved in this scheme. The RS chooses x as a master key and a secret number y. Further, it computes $h(x \parallel y)$ and $h(y)$. Then, these parameters are shared with server RS through a secure channel. The parameters x and y are known only to RS [31].

The sequence diagram is shown in figure 4, which provides details of this scheme. User directly sends the parameters CID_i, P_{ij}, Q_i and nonce N_i on the basis of parameter A_i. The calculation of these parameters is as follows:

$$A_i = h(b \parallel PWD_i), \tag{30}$$

$$CID_i = h(b \oplus PWD_i \oplus R_i) \oplus h(T_i \parallel A_i \parallel N_i), \tag{31}$$

$$P_{ij} = T_i \oplus h(h(y) \parallel N_i \parallel SID_j), \tag{32}$$

$$Q_i = h(O_i \parallel A_i \parallel N_i). \tag{33}$$

Service providing server S calculates the parameter M_{ij} and sends this back to user's end. The calculation of M_{ij} is as follows:

$$M_{ij} = h(O_i \parallel A_i \parallel N_i \parallel SID_j). \tag{34}$$

After receiving M_{ij}, user calculates another parameter M''_{ij} and sends back to service providing server:

$$M''_{ij} = h(O_i \parallel A_i \parallel N_i \parallel SID_j). \tag{35}$$

CID_i is the parameter saved on smart card along with other parameters. SID_j is server's ID; $P_{ij}, Q_i, O_i, M_{ij}, M''_{ij}, T_i, M_i, R_i$ and A_i are the intermediate parameters used for authentication. N_i is nonce generated at user side [31].

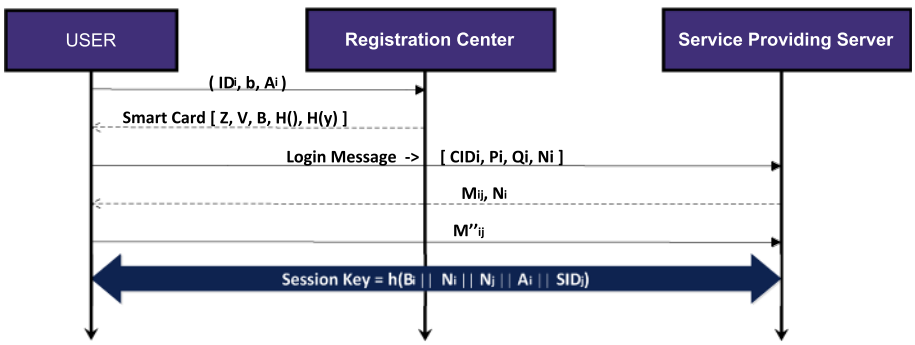

Figure 4. Sequence flow of Leu and Hsieh authentication protocol.

Finally, the session key, i.e. SK, will be generated at both the ends as follows:

$$SK = h(B_i \parallel N_i \parallel N_j \parallel A_i \parallel SID_j). \quad (36)$$

All the afore-mentioned schemes provide identity protection, session key agreement and a separate phase for password update or change. Masquerade attack cannot be performed on all the schemes except on the Leu and Hsieh scheme. Replay attack is possible only on the Lee *et al* scheme but rest of the schemes are safe from this attack. According to Gahrana and Anand [10], there is no scheme that meets all the security requirements and attains all the security goals. These schemes give an idea about the authentication for the afore-mentioned proposed problem of this paper. Therefore, the proposed scheme is motivated from the analysis of these schemes.

4. Proposed authentication scheme

The proposed scheme will work for distributed, multi-server-based environment where various services are provided by different servers. These services work in isolation, but the proposed authentication scheme provides an integrated view to the existing e-governance services as a unified e-governance system in which registration/login credentials of the user can work for all the servers and services. The symbols used in this scheme are illustrated in table 2.

The proposed integrated and unified authentication system contains three basic layers of the operations.

1. *User layer (U)*: It is the client layer from where the user can access the services. The subscript 'i' relates the used parameters to i^{th} user.
2. *Control and authentication server layer (CAS)*: It is the layer responsible for completing the authentication process of the user and helps deduce the session key.
3. *Department Service layer (DS)*: This layer is responsible for providing services (related to their department) to legitimate users.

The details of this architecture for proposed scheme are illustrated in figure 5, in which there are three algorithms/processes.

1. *Registration process*: Users present and upload required documents, and they are provided a smart card. User layer and the department layer must get registered at the CAS layer. For this purpose, two registration processes are designed.
   (a) *User registration (at user layer)*: This process is responsible for registering a user for authentication.
   (b) *Department server registration (at DS layer)*: This process is used to register the server of *DSlayer*, which is responsible for providing specific government service.
2. *Login process*: Various services will be provided to legitimate users along with session key, which will be shared after verification of the legitimacy of the user.
3. *Password change process*: Allow users to change their passwords either in case of loss or a routine change.

A detailed description and calculations are as follows.

Table 2. Notations used in our proposed protocol.

Symbol	Description
CSC	Common service centre
DS	Department server under government ownership
DSO	One of the DS where user initiates interaction with government
CAS	Central authentication server, responsible for authentication
U_i	i^{th} user from set U
$h()$	A one-way hash function (() denotes the function/process)
E	Encryption process
$\parallel$	The bitwise concatenation operation
$\oplus$	The bitwise XOR operation
UID_i	User Identity for user U_i
$r1$, $r2$, $r3$	Random numbers generated at CSC, DSO & CAS
key1, key2	Encryption keys between CSC & DSO and DSO & CAS
ID_{DSO}	Legitimate ID of DSO
ID_{DS}	Legitimate ID of DS
$TS1_i, TS2_i, TS3_i$	Time-stamps at CSC, DS and CAS, respectively
$N1_i, N2_i, N3_i$	Random nonces generated at CSC, DS and CAS, respectively
PIN	User has this key used for encryption on smart card data
$\Delta TS_{DS_{TV}}$	Acceptable difference between time-stamps at DS with TS1
$\Delta TS_{CAS_{TV}}$	Acceptable difference between time-stamps at CAS with TS1
$\Delta TS_{DSO_{TV}}$	Acceptable difference between time-stamps at DSO with TS1
$SessKey$	Session key used for communication after authentication

4.1 *Registration process*

This process consists of the following two processes.

4.1a *Registration process for user (UR)*: For the registration, user calculates the parameter UID_i in Eq. (37), using an arbitrarily selected id U_i, password PWD_i and a

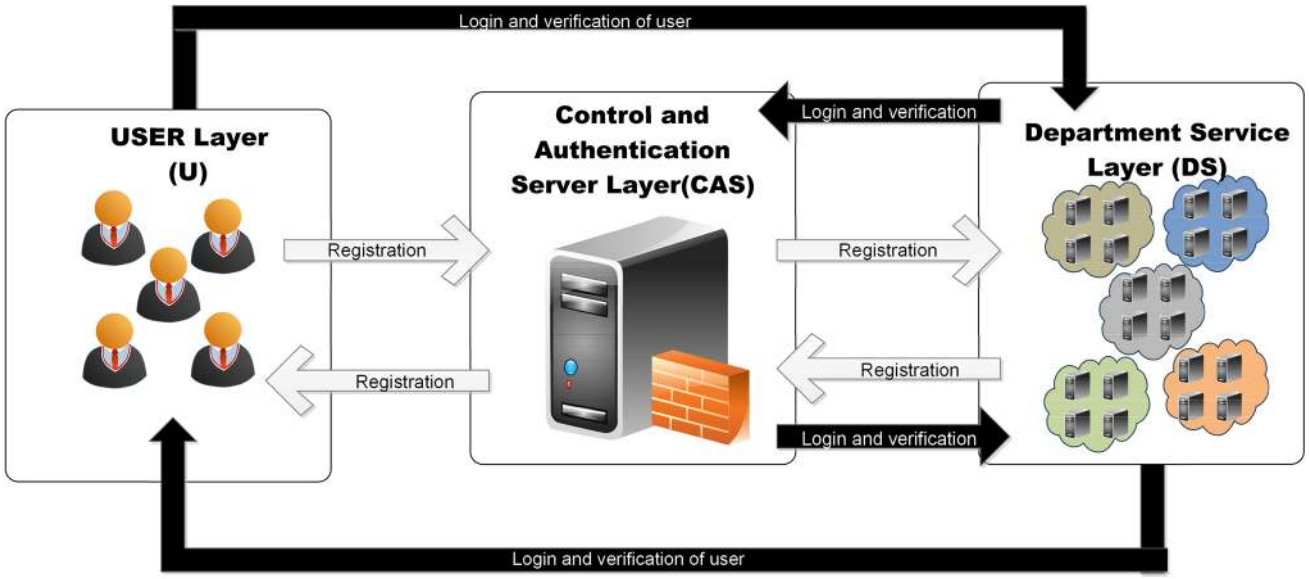

Figure 5. Integrated authentication framework for multi-server-based system.

random number r_{U_i}. Later, the user sends this message to CAS. The process is illustrated in algorithm 1. The steps of this process are as follows.

UR Step-1. In this step, users select a random number r_{U_i} and calculate UID_i:

$$UID_i = h(U_i \parallel PWD_i \parallel r_{U_i}). \tag{37}$$

Now the message $MSG_{USER_{to}CAS}$ is sent to the CAS server for registration. The calculation of $MSG_{USER_{to}CAS}$ is as follows:

$$MSG_{USER_{to}CAS} \leftarrow E(U_i \parallel UID_i \parallel r_{U_i}). \tag{38}$$

Here, E denotes the encryption process; this implies that the message should be transferred through a secure communication channel. E may be based on public key or secret key encryption, which totally depends on the government policies. For the generic purpose, E represents the process for secure communication; $r_{i_{CAS'}}$ and $r_{i_{CAS''}}$ are two random numbers, generated at CAS.

UR Step-2. Message is successfully received by the CAS server. CAS decrypts the message and computes $\alpha 1_i$, $\alpha 2_i$, $\alpha 3_i$ and $\alpha 4_i$:

$$\alpha 1_i = h(U_i \parallel UID_i \parallel r_{U_i}), \tag{39}$$

$$\alpha 2_i = h(\alpha 1_i \parallel r_{i_{CAS'}}), \tag{40}$$

$$\alpha 3_i = h(\alpha 2_i \parallel r_{i_{CAS''}}), \tag{41}$$

$$\alpha 4_i = \alpha 2_i \oplus \alpha 3_i. \tag{42}$$

These parameters are further used for authentication. CAS stores $\alpha 2_i$ and creates a message $MSG_{CAS_{to}USER}$ in Eq. (43). Then CAS sends $MSG_{CAS_{to}USER}$ to $USER_i$:

$$MSG_{CAS_{to}USER} \leftarrow E(\alpha 1_i \parallel \alpha 3_i \parallel \alpha 4_i \parallel h() \parallel r_{U_i}). \tag{43}$$

UR Step-3. At the user layer, parameters $\alpha 1_i$, $\alpha 3_i$, $\alpha 4_i$ and r_{U_i} are calculated from the message $MSG_{CAS_{to}USER}$ and written on the smart card provided to the user.

4.1b *Registration process for department server (DR)*: This process must be used to register the department server at CAS layer. The DS server is denoted by suffix 'j' for j^{th} server in this scheme. This registration process is illustrated in algorithm 2. The steps of this process are as follows.

DR Step-1. DS server sends its ID, i.e. $DSID_j$, to CAS.

DR Step-2. CAS server calculates SID_j, where CAS selects two random numbers ($r_{j_{CAS}}$ and $r_{j_{DS}}$) for registration of server DS_j:

$$SID_j = h(DSID_j \parallel r_{j_{CAS}}). \tag{44}$$

DR Step-3. Now, CAS sends SID_j and $r_{j_{DS}}$ to DS_j.

DR Step-4. After receiving the parameter SID_j and random number $r_{j_{DS}}$, DS_j calculates the parameter CID_j:

$$CID_j = h(SID_j \parallel r_{j_{DS}}). \tag{45}$$

Algorithm 1 User registration algorithm.

procedure USER-REGISTRATION(U_i,PWD_i,r_{U_i})
At USER level:
  $UID_i = h(U_i \parallel PWD_i \parallel r_{U_i})$
  $MSG_{USER_{to}CAS} \leftarrow E(U_i \parallel UID_i \parallel r_{U_i})$
At CAS level:
  $\alpha 1_i = h(U_i \parallel UID_i \parallel r_{U_i})$
  $\alpha 2_i = h(\alpha 1_i \parallel r_{i_{CAS'}})$ ▷ $r_{i_{CAS'}}$ is random number at CAS
  $\alpha 3_i = h(\alpha 2_i \parallel r_{i_{CAS''}})$ ▷ $r_{i_{CAS''}}$ is random number at CAS
  $\alpha 4_i = \alpha 2_i \oplus \alpha 3_i$
  $MSG_{CAS_{to}USER} \leftarrow E(\alpha 1_i \parallel \alpha 3_i \parallel \alpha 4_i \parallel h(.) \parallel r_{U_i})$
  at user layer, separate the parameters ($\alpha 1_i$, $\alpha 3_i$, $\alpha 4_i$, $h()$, r_{U_i})
  write parameters ($\alpha 1_i$, $\alpha 3_i$, $\alpha 4_i$, h(), r_{U_i}) to Smart Card.
  Deliver/hand over the smart card with $\alpha 1_i$, $\alpha 3_i$, $\alpha 4_i$, h(.), r_{U_i} to User

4.2 *Login process (LP)*

Legitimate users prove their identity using this login process to access the authorized services using their registered smart card. The process is divided into five steps; a detailed description of these steps is as follows.

LP Step-1. In this step, registered users swipe their smart card and input their password PWD_i. Following calculations for UID_i and $\alpha 1_i^{'}$ are done at the user's layer using Eqs. (37) and (42). Now, the algorithm compares calculated $\alpha 1_i^{'}$ with stored $\alpha 1_i$; session terminates if both $\alpha 1_i$ and $\alpha 1_i^{'}$ are not the same. This step helps check the legitimacy of the user because password is known only to the user. Further, the parameters $\alpha 2_i$, T_i, λ_{ij}, $\lambda_{ij}^{'}$ and M_i are calculated.

$$T_i = \alpha 2_i \oplus N1_i \tag{46}$$

where $N1_i$ is the nonce at user layer. The term $TS1_i$ represents the time-stamp and $DSID_j$ is the ID of requested department server (when users select a particular e-governance service, the request is redirected from CSC to the respective department server, i.e. $DSID_j$).

$$\lambda_{ij} = h(\alpha 2_i) \oplus h(DSID_j \parallel TS1_i) \tag{47}$$

$$\lambda_{ij}^{'} = h(\alpha 2_i \parallel (\alpha 1_i \oplus N1_i) \parallel append0(N1_i) \parallel append0(U_i) \tag{48}$$

where *append*0 is the process to append two bits '0'.

$$M_i = U_i \oplus h(\alpha 1_i \parallel \lambda_{ij}). \tag{49}$$

Now the parameters $M_i, \alpha 1_i, \alpha 2_i, \lambda_{ij}^{'}, TS1_i, T_i$ are sent to department server, where the service will be provided to the requested user at DS layer.

LP Step-2. At this step, DS layer checks the time-stamp and prevents the timing attack and replay attack by comparing the received time-stamp ($TS1_i$) with its time-stamp ($TS2_i$). If $TS1_i$ is found to be greater than $TS2_i$ then process terminates, else it further calculates the other parameters $\beta 1_i$, $\beta 2_i$, $\beta 3_i$:

$$\beta 1_i = CID_j \oplus N2_i. \tag{50}$$

$N2_i$ and $TS2_i$ are nonce and time-stamp at DS layer.

$$\beta 2_i = h(N2_i \parallel CID_j \parallel \lambda_{ij}^{'}), \tag{51}$$

$$\beta 3_i = DSID_j \oplus h(CID_j \oplus N2_i). \tag{52}$$

LP Step-3. Now, the parameters $M_i, \alpha 1_i, \alpha 2_i, \beta 1_i, \beta 2_i, \beta 3_i, \lambda_{ij}^{'}, DSID_j, TS1_i, T_i$ shall be sent to CAS server, to finalize the authentication. Now, CAS performs the calculation of SID_j and CID_j for verification in Eqs. (44) and (45). Using the calculated parameters SID_j, CID_j and received parameter $\beta 1_i$, the algorithm calculates $N2_i$ given in Eq. (50). Further, the process checks the integrity through the calculation and comparison of calculated values of $\beta 2_i^*$ and $\alpha 2_i^*$ with received values of $\beta 2_i$ and $\alpha 2_i$ in Eqs. (51) and (43), respectively.

Algorithm 2 Department Server DS Registration algorithm.

procedure DS-REGISTRATION($DSID_j$) ▷ DS are sent its ID $DSID_j$ to CAS
At CAS layer:
  $SID_j = h(DSID_j \parallel r_{jCAS})$ ▷ r_{jCAS} & r_{jDS} are random numbers selected by CAS
  SID_j & r_{jDS} are sent to DS layer
At DS layer:
  $CID_j = h(SID_j \parallel r_{jDS})$
These two parameters, i.e. SID_j & CID_j, will be used for login process :

If the comparison is successful, then the system concludes that the user is authentic and further calculations of parameters are used for sharing the session key; λ_{ij}, U_i and $N1_i$ are calculated using (47), (49) and (46), respectively. After deducing the nonce $N1_i$, which is exclusively generated at CSC, CAS calculates $\delta1_{ij}$, $\delta2_{ij}$, $\delta3_{ij}$ and $\delta4_{ij}$; $\delta1_{ij}$ and $\delta2_{ij}$ are for DS, and $\delta3_{ij}$ and $\delta4_{ij}$ are for user. Mutual authentication between CAS and DS as well as with CSC takes place using these parameters. Calculation of these parameters is given in Eqs. (53)–(56):

$$\delta1_{ij} = (N1_i \oplus N3_i) \oplus h(DSID_j \parallel N2_i \parallel CID_j), \tag{53}$$

$$\delta2_{ij} = h(N1_i \oplus N3_i). \tag{54}$$

$N3_i$ is the random nonce generated at CAS.

$$\delta3_{ij} = (N2_i \oplus N3_i) \oplus h(U_i \parallel N1_i \parallel \alpha1_{ij}), \tag{55}$$

$$\delta4_{ij} = h(N2_i \oplus N3_i). \tag{56}$$

Now the parameters $\delta1_{ij}$, $\delta2_{ij}$, $\delta3_{ij}$ and $\delta4_{ij}$ are sent to DS server, to acknowledge the authentication process and deduce the session key.

$$Key_{session} = N1_i \oplus N2_i \oplus N3_i. \tag{57}$$

LP Step-4. Now, DS performs the following calculation to generate session key. From Eq. (53), if $\delta1_{ij}$ is *XOR* with $h(DSID_j \parallel N2_i \parallel CID_j)$, the term $(N1_i \oplus N3_i)$ is received. Initially, DS calculates $(N1_i \oplus N3_i)$ through the received parameters from CAS:

$$(N1_i \oplus N3_i) = \delta1_{ij} \oplus h(DSID_j \parallel N2_i \parallel CID_j). \tag{58}$$

Now calculate $\delta2^*_{ij}$ through $(N1_i \oplus N3_i)$. If the calculated parameter $\delta2^*_{ij}$ is the same as received $\delta2_{ij}$, then the process will go forward; else the process terminates the session with failed status. If both the received and calculated, i.e. $\delta2_{ij}$ and $\delta2^*_{ij}$, are the same, then calculate the session key.

$$Key_{session} = N2_i \oplus (N1_i \oplus N3_i). \tag{59}$$

Now, DS server sends the parameters $\delta3_{ij}$ and $\delta4_{ij}$ to the requested user layer, which waits for acknowledgement of authentication and also for session key.

LP Step-5. User layer calculates $(N2_i \oplus N3_i)$ by Eq. (60):

$$(N2_i \oplus N3_i) = \delta3_{ij} \oplus h(U_i \parallel N1_i \parallel \alpha1_i). \tag{60}$$

Further, users calculate $\delta4^*_{ij}$ by Eq. (56). If this calculated parameter $\delta4^*_{ij}$ is the same as received $\delta4_{ij}$ then they go forward, else they terminate the session with failed status. If both the received and calculated, i.e. $\delta4_{ij}$ and $\delta4^*_{ij}$, are the same then they calculate the session key as illustrated in Eq. (61):

$$key_{session} = N1_i \oplus (N2_i \oplus N3_i). \tag{61}$$

This is a way to share the session key among all the three layers for further communication. The sequence diagram of this flow of login process is illustrated in figure 6.

4.3 *Password change process*

Users can use this process to change their password. In this process, users send their ID and password, i.e. U_i and PWD_i, to CAS server with calculated parameters UID^*_i and $MSG^*_{USER_{to}CAS}$ using Eqs. (37) and (38), respectively. If CAS finds that $MSG^*_{USER_{to}CAS}$ is the same as $MSG_{USER_{to}CAS}$ then CAS allows users to send a new password, and later password-related steps are further performed as in the registration process, which is already discussed.

5. Security analysis and discussion

This section examines and analyses the proposed scheme in comparison with other multi-server-based authentication schemes. A threat model is developed to assess various security attacks and available mitigation. Further, performance of the proposed scheme is analysed, both in terms of required computation and communication efforts.

5.1 *Threat model*

A threat model is designed to identify various vulnerabilities and protection against these vulnerabilities. Figure 7 shows the threat model for the proposed authentication scheme. Applying the threat analysis, totally six attacks and three preventive actions are identified to strengthen the security of the scheme. At the user layer, three attacks identified are threat to user's identity, user anonymity attack and stolen smart card attack. Between user layer and DS layer, the three attacks identified are masquerade attack, replay attack and DOS attack and and three security functionalities provided are session key agreement, traceability and mutual authentication. Between DS layer and CAS

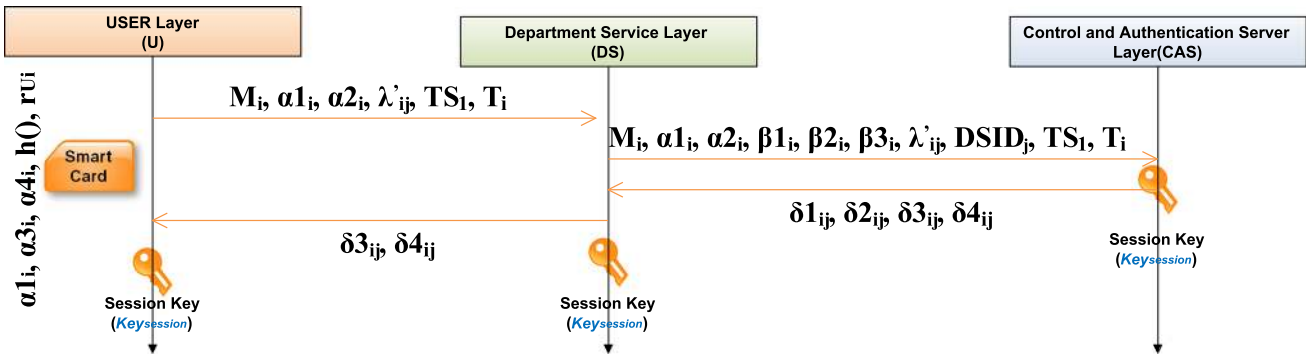

Figure 6. Login process.

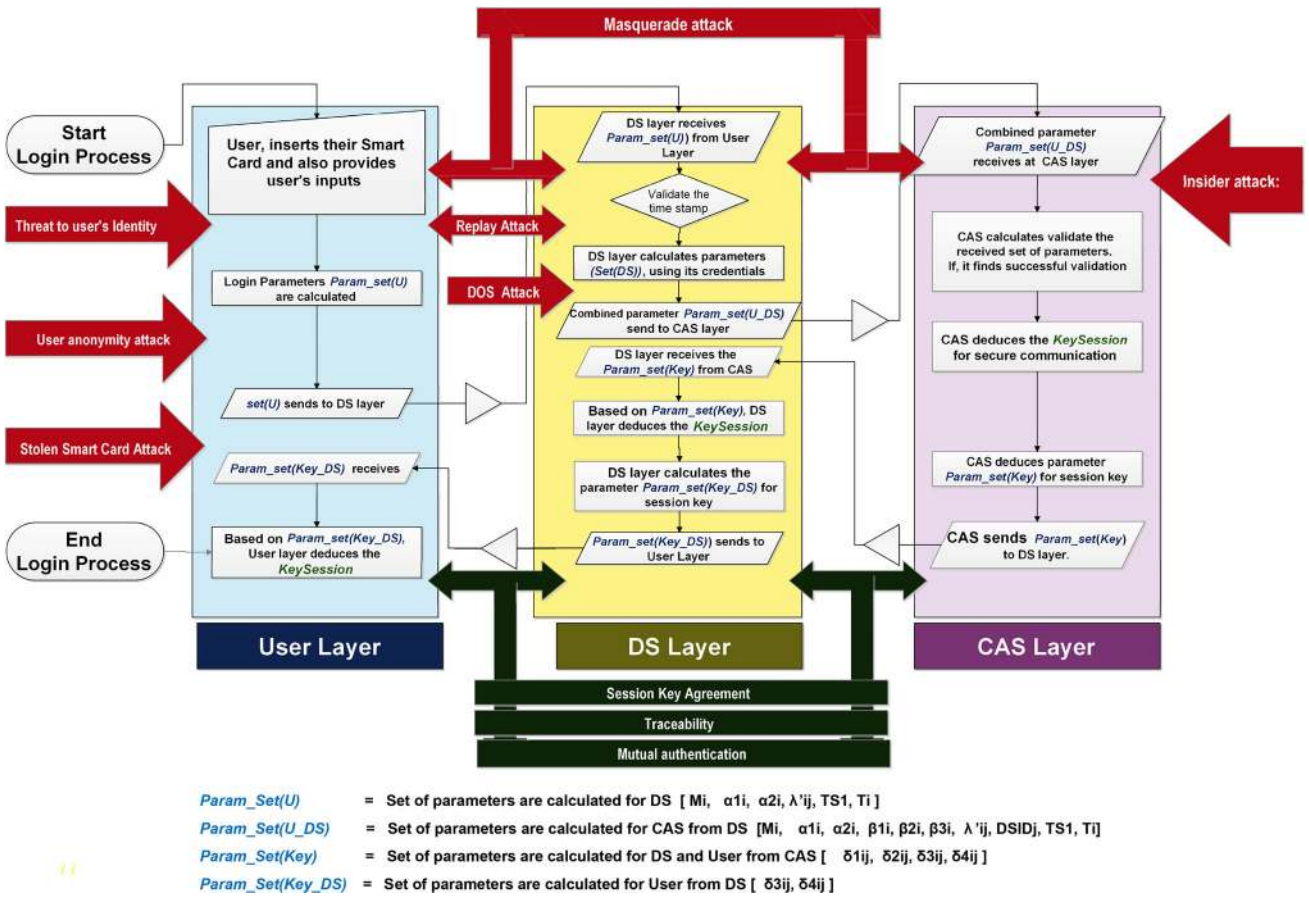

Figure 7. Threat model of proposed scheme.

layer, one attack and three security functionalities identified are masquerade attack and session key agreement, traceability and mutual authentication, respectively. Finally, at CAS layer, one attack may be possible that is identified by this threat model, i.e., insider attack. All the attacks and security functionalities are illustrated in figure 7. How this scheme protects itself from these attacks and provides other security functionalities are explained in the next section.

5.2 *Security analysis of scheme*

The security testing and analysis of the proposed scheme are discussed here.

5.2a *Security attacks*

Identity protection and user anonymity: There are two basic and necessary requirements for any authentication algorithm: first,

protection of identity and second, anonymity preservation of user [32]. The proposed scheme prevents identity theft and preserves the anonymity of users as their identity is not available on communication channel in its original form. Rather, it is processed by Eqs. (37) and (38) during the registration process as given in algorithm 1.

Masquerade attack: Another security attack is man in middle or masquerade attack, in which an attacker can monitor or/and forge the communicated messages [33]. In the proposed scheme, it is impossible for a man in the middle to identify or calculate session key because it is based on parameters distributed in disjoint manner at three layers of the proposed architecture. It is difficult to extract the part of session key from received message as Eqs. (57), (59) and (61) are the combination of nonces of each layer. Nonces ($N1_i$, $N2_i$ and $N3_i$) at each layer (user's layer, DS and CAS, respectively) are known only by the respective layer exclusively, i.e. $N1_i$ is available only at user's layer, $N2_i$ is available only at the DS layer and $N3_i$ is available only at the CAS layer. Therefore, the session key is calculated only when all the three nonces are available at any end. It is impossible for an attacker to work between user's layer and DS layer, and at the same time also be between DS layer and CAS layer.

Replay attack: This attack includes blocking the past messages and later replaying them to the server or receiver, and it seems that the whole process is initiated as a legitimate user [34]. It is an attack in which users can easily impersonate a legitimate user. Due to the use of time-stamp $TS1_i$, our proposed scheme resists this attack.

Password updating/changing: To provide controlled access for an application/service, it is highly desirable to change passwords frequently. In the proposed scheme, users are allowed to change the password on providing their credentials stored on the smart card to CAS layer, which calculates parameters UID_i and $\alpha 1_i^{'}$ using Eqs. (37) and (42), respectively, and matches these with already stored parameters. If the matching is successful then users can regenerate their password using algorithm 1 with a new password.

DoS attack: To prevent DoS attack [35] the proposed scheme maintains time-stamps at DS and CAS layers. When the login process is initiated at user's layer, it sends a $TS1_i$ to DS. On receiving the request, DS layer recalculates time-stamp that is not later than the set threshold value. The same procedure is repeated at CAS layer. To prevent flooding from user's layer, the password PWD_i is entered along with user's ID U_i. Later parameter $\alpha 1_i^{'}$ is calculated (Eq. (42)) to compare it to its stored value in smart card. Due to this comparison, the DoS attack is resistant for this scheme.

Insider attack: An insider (like administrator, etc.) is a legitimate person to access the services and data of the computer or network system, and can misuse this information or system [36]. In the proposed scheme, the password is not transmitted to the server in plain (Eq. (37)). In fact, the proposed scheme uses various operations, including hashing, to protect the system from insider attack. A malicious insider user possessing a legitimate smart card cannot extract any secret information even after applying brute force attack on the elements ($\alpha 1_i$, $\alpha 3_i$, $\alpha 4_i$, r_{U_i}), which are stored in the smart card.

Stolen smart card attack: The security algorithm should be designed in such a manner so as to resist the attack, in which an attacker steals the smart card and tries to access the system. Physical protection methods cannot resist malicious attackers getting the stored elements, but our scheme is designed such that finding stored elements will not help

an attacker in obtaining credentials because to deduce the first element for verification process, UID_i needs user's ID U_i and password PWD_i as given in Eq. (37).

Session key agreement: A session key is computed at all the layers, involved in communication, to shield the communication among them to provide secrecy. In our scheme, a session key $key_{session}$ is generated to secure the correspondence between the imparting parties. The calculation of $key_{session}$ is given in Eqs. (57), (59) and (61).

5.2b *Security functionalities*

Mutual authentication: Mutual authentication is an essential property for any authentication technique, in which all the involved parties authenticate each other [37]. In the proposed scheme all the layers share their masked IDs (when plain ID is processed with cryptographic operations like XOR, etc., to protect it from attackers is called as masking ID) during the registration process (as shown in algorithms 2 and 1) and later on these IDs are used for mutual authentication during the login process as in Eqs. (37), (48), (50)–(52), (44), (45), (53) and (55).

Session key agreement: A session key is computed at all the layers, involved in communication, to shield the communication among them to provide secrecy. In our scheme, a session key $key_{session}$ is generated to secure the correspondence between the imparting parties. The calculation of $key_{session}$ is given in Eqs. (57), (59) and (61).

Traceability: During the login phase, the control server can compute user's original ID and this feature is called traceability, which makes a user traceable on the control server. In our scheme, original ID, i.e. U_i, can be computed as in Eq. (49), and hence the scheme has the unique feature to support traceability.

5.3 *Performance analysis*

The performance of any security algorithm depends on two factors.

Computation overhead: Comparison of computation overhead of our proposed scheme with other existing techniques is given in table 3. The implementation delay of the proposed scheme is due to login phase, authentication and key agreement phases; therefore, while calculating computation overhead, we consider all the three phases (login, authentication and key agreement). The predominant reason of overhead is hashing; due to this, we assume that T_{hash} is the time for one hash. Computation overhead of the proposed technique is as follows.

Login and authentication: Totally 15 used hashes are reported in this phase, up to full authentication of the user. Therefore, computation overhead for this phase is $15T_{hash}$.

Key agreement: When the CAS verifies the user and completes the process of key agreement, totally used hashes are 8; hence, the computation cost for key agreement phase of our proposed scheme is $8T_{hash}$.

Therefore, the total computation overhead is $15T_{hash} + 8T_{hash} = 23T_{hash}$. The same process has been applied to calculate the computation overhead for considered techniques, as shown in table 3. The protocol of Xue *et al* [30] has a cost of $24T_{hash}$; thus, if we include all the steps to resist all the attacks as given in table 4, the proposed scheme is better.

Communication overhead: The communication overhead is calculated in terms of total number of bits transferred during login, authentication and session key generation processes. Lengths of messages

Table 3. Computation overhead of various techniques.

Sl. no.	Protocols	Computation overhead of the authentication and key agreement phase
1	Lee *et al*'s	$25T_{hash}$
2	Li *et al*'s	$27T_{hash}$
3	Leu *et al*'s	$28T_{hash}$
4	Sood *et al*'s	$24T_{hash}$
5	Xue *et al*'s	$24T_{hash}$
6	Our proposed scheme	$23T_{hash}$

Table 4. Comparative analysis of various techniques with the proposed technique ((1) Lee *et al* [8], (2) Li *et al* [29], (3) Leu *a*nd Hsieh [31], (4) Sood *et al* [2], (5) Xue *et al* [30] and (6) our proposed scheme).

Sl. no.	Security requirements	(1)	(2)	(3)	(4)	(5)	(6)
1	Anonymity of user & identity protection	✓	✓	✓	✓	✓	✓
2	Traceability	×	×	×	×	✓	✓
3	Mutual authentication	✓	✓	✓	✓	✓	✓
4	Session key agreement	✓	✓	✓	✓	✓	✓
5	Password updating and changing	✓	✓	✓	✓	✓	✓
6	Resistance to insider attack	✓	×	✓	×	✓	✓
7	Resistance to stolen smart card	✓	✓	×	×	✓	✓
8	Resistance to replay attack	×	×	✓	×	✓	✓
9	Resistance to denial-of-service attack	×	×	×	×	✓	✓
10	Masquerade attack	×	×	✓	×	✓	✓

Table 5. Message length comparison of various techniques with our proposed scheme.

Sl. no.	Protocols	Message $U_i \rightarrow S_j$	Length $S_j \rightarrow CS$	(bytes)$CS \rightarrow S_j$	$S_j \rightarrow U_i$	Total
1	Lee *et al*'s	72	–	–	24	96
2	Sood *et al*'s	64	80	64	32	240
3	Li *et al*'s	64	112	64	32	272
4	Leu *et al*'s	72	–	–	32	104
5	Xue *et al*'s	83	163	64	32	342
6	Proposed scheme	88	152	64	32	336

including $U_i \rightarrow S_j$, $S_j \rightarrow CS$, $CS \rightarrow S_j$ and $S_j \rightarrow U_i$, transferred between two of the user, service provider and CS layers, have been calculated to determine total number of bits. Assuming that the length of each hash value is 128 bits, the length of the time-stamp value is 24 bits, and that of each of the other transmitted elements is also 128 bits. Table 5 shows a total message length of 336 in comparison with Xue *et al* [30], whose length is 342, involving all the three layers. Communication overhead is mainly compared with Xue *et al* protocol because it is the only protocol resistive to all the attacks involved in the proposed scheme. In comparison with Xue *et al* protocol, communication overhead of the proposed scheme is lesser by 6 bytes.

6. Conclusion

In this paper, a strong authentication scheme is proposed, which is not only useful to prove the legitimacy of the user in multi-server environment, but also able to share the session key between all the stakeholders engaged in this process. The proposed scheme has been compared to other techniques on the basis of various security attacks like leak-of-verifier attack, eavesdropping attack, stolen smart card attack, DoS attack, replay attack and forgery attack and found to be resistive for all. The proposed scheme utilizes time-stamp to oppose replay attack. Further, the proposed scheme compared on the basis of overhead cost (computation and communication) gives better results. This scheme has features like being dynamic in nature, ID-based and multi-server oriented, not needing to store verification table and also provides session key at all involved layers for secure communication. The users of this scheme become confident and feel comfortable, because they are not require to remember a number of login credentials for multiple services and servers. This scheme works for distributed environment and integrates various servers and services through a single registration and authentication. Therefore, this proposed scheme is best suited for current e-governance system of India. This scheme can not only integrate the e-governance projects in distributed environment but also provide unified authentication platform to facilitate the user.

References

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

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

[3] Li C T, Lee C C Weng C Y and Fan C I 2015 A secure dynamic identity based authentication protocol with smart cards for multi-server architecture. *Journal of Information Science and Engineering* 31(6): 1975–1992

[4] Li X, Ma J, Wang W, Xiong Y and Zhang J 2013 A novel smart card and dynamic ID based remote user authentication scheme for multi-server environments. *Journal of Network and Computer Applications* 58(1): 85–95

[5] Li X, Qiu W, Zheng D, Chen K and Li J 2010 Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards. *IEEE Transactions on Industrial Electronics* 57(2): 793–800

[6] Hsiang H C and 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

[7] Hwang T, Chen Y and Laih C J 1990 Non-interactive password authentications without password tables. In: *Proceedings of IEEE TENCON'90: 1990 IEEE Region 10 Conference on Computer and Communication Systems*, pp. 429–431

[8] Lee C C, Lin T H and Chang R X 2011 A secure dynamic ID based remote user authentication scheme for multi-server environment using smart cards. *Journal of Network and Computer Applications* 38(11): 13863–13870

[9] Huang X, Xiang Y, Chonka A, Zhou J and Deng R H 2011 ATCS: a novel anonymous and traceable communication scheme for vehicular ad hoc networks. *International Journal of Network Security* 13(2): 71–78

[10] Gaharana S and Anand D 2015 Dynamic ID based remote user authentication in multi server environment using smart cards: a review. In: *Proceedings of the IEEE International Conference on Computational Intelligence and Communication Networks (CICN)*, pp. 1081–1084

[11] Huang X, Xiang Y, Chonka A, Zhou J and Deng R H 2011 A generic framework for three-factor authentication: preserving security and privacy in distributed systems. *IEEE Transactions on Parallel and Distributed Systems* 22(8): 1390–1397

[12] Jiang Q, Ma J, Li G and Li X 2015 Improvement of robust smart-card-based password authentication scheme. *International Journal of Communication Systems* 28(2): 383–393

[13] Wang D and Wang P 2014 Understanding security failures of two-factor authentication schemes for real-time applications in hierarchical wireless sensor networks. *Ad Hoc Networks* 20: 1–15

[14] Wang D, He D, Wang P and Chu C H 2015 Anonymous two-factor authentication in distributed systems: certain goals are beyond attainment. *IEEE Transactions on Dependable and Secure Computing* 12(4): 428–442

[15] Wang D, Wang N, Wang P and Qing S 2015 Preserving privacy for free: efficient and provably secure two-factor authentication scheme with user anonymity. *Information Sciences* 321: 162–178

[16] Wang D, Gu Q, Cheng H and Wang P 2016 The request for better measurement: a comparative evaluation of two-factor authentication schemes. In: *Proceedings of the 11th ACM Asia Conference on Computer and Communications Security*, pp. 475–486

[17] Li X, Xiong Y, Ma J and Wang W 2007 G2C e-government: modernisation or transformation? *Electronic Government, an International Journal* 4(1): 68–75

[18] Sambasivan M, Patrick Wemyss G and Che Rose R 2010 User acceptance of a G2B system: a case of electronic procurement system in Malaysia. *Internet Research* 20(2): 169–187

[19] Karacapilidis N, Loukis E and Dimopoulos S 2005 Computer-supported G2G collaboration for public policy and decision-making. *Journal of Enterprise Information Management* 18(5): 602–624

[20] Coston J M 1998 Nonprofit and voluntary sector quarterly. *Journal of Network and Computer Applications* 27(3): 358–382

[21] Anand D and Khemchandani V 2016 The challenges for authentication in Indian e-Governance System (a survey on Indian administrative staff). *International Journal of Control Theory and Applications* 40(9): 335–346

[22] Bhatt R and Kumar S 2015 E-authentication framework for secure e-governance services. *International Journal of Current Innovation Research* 1(2): 30–36

[23] National Information Center (NIC) 2015 *e-pramaan: Framework for e-authentication*. https://egovstandards.gov.in/faq/e-pramaan-framework-e-authentication

[24] Kumar M and Vaisla K S 2014 e-Authentication framework for e-governance review paper. In: *Proceedings of the International Conference on Advances in Computing and Communication (ICACCE-2014)*

[25] Camenisch J L, Lehmann A and Neven G 2015 *Password-based authentication*. US Patent App. 14/745,086

[26] Chang C C and Wu T C 1999 Remote password authentication with smart cards. *IEE Proceedings E-Computers and Digital Techniques* 138(3): 165–168

[27] Nugent B 1987 Password-based authentication. *ACM SIGSAC Review* 5(4): 10–13

[28] Abdalla M and Pointcheval D 2005 Interactive Diffie–Hellman assumptions with applications to password-based authentication. In: *Proceedings of the International Conference on Financial Cryptography and Data Security*, pp. 341–356

[29] Li X, Ma J, Wang W, Xiong Y and 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

[30] Xue K, Hong P and Ma C 2014 An efficient and security dynamic identity based authentication protocol for multi-server architecture using smart cards. *Journal of Computer and System Sciences* 80(1): 195–206

[31] Leu J S and Hsieh W B 2014 Efficient and secure dynamic ID-based remote user authentication scheme for distributed systems using smart cards. *IET Information Security* 8(2): 104–113

[32] Chien H Y and Chen C H 2005 A remote authentication scheme preserving user anonymity. In: *Proceedings of the 19th IEEE International Conference on Advanced Information Networking and Applications (AINA'05)*, vol. 2, pp. 245–248

[33] Awasthi A K and Lal S 2012 A remote user authentication scheme using smart cards with forward secrecy. *IEEE Transactions on Consumer Electronics* 49(4): 1246–1248

[34] Albert R, Edgett J and Sunder S 2002 *Method and system for identifying a replay attack by an access device to a computer system*. US Patent App. 10/118,406, Google Patents

[35] Schuba C L, Krsul I V, Kuhn M G, Spafford E H, Sundaram A and Zamboni D 1997 Analysis of a denial of service attack on TCP. In: *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 208–223

[36] Salem M Ben, Hershkop S and Stolfo S J 2008 Insider attack and cyber security. In: *A survey of insider attack detection research*, pp. 69–90

[37] Otway D and Rees O 1987 Efficient and timely mutual authentication. *ACM SIGOPS Operating Systems Review* 21(1): 8–10

[38] Li X, Xiong Y, Ma J and 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

[39] Li X, Xiong Y, Ma J and Wang W 1990 Non-interactive password authentications without password tables. In: *Proceedings of IEEE TENCON'90: IEEE Region 10 Conference on Computer and Communication Systems*, pp. 429–431