A Novel Verification Protocol to Restrict Unconstitutional Access of Information From Smart Card

Ajay Kumar Sahu, Raj Kumar Goel Institute of Technology and Management, India

Ashish Kumar, ITS Engineering College, India

ABSTRACT

The services of the internet play an essential part in the daily life of the users. So, safety and confidentiality of the information are to be maintained to preserve user conviction in various services offered by network. The two-factor-based password verification techniques are used between remote server and legitimate users for verification over insecure channel. Several protocols have been suggested previously claiming their simplicity, privacy, safety, and robustness. The authors proved that their enhanced protocols are vulnerable to different attacks on the network and permit only authenticated users to update their password preserving traceability and identity. Analysis shows that no scheme has fulfilled all the security requirements and achieved entire goals. Therefore, in this article, a scheme has been presented to overcome these issues in the previous schemes to resist illegal access leading to misuse and achieve all the security requirements and goals. The safety analysis of the presented scheme has confirmed its performance in terms of reliability and safety.

KEYWORDS

Hash Function, Identity, Information Retrieval, Key Agreement, Mutual Authentication, Password, Security, Smart Card

INTRODUCTION

As time grows day by day, dependency of user in various technology increases which constituted a challenge regarding validity of the remote user. There are various types of attacks possible in the network which causes significant financial loss. Therefore, there is a requirement of some techniques to validate the legitimate users to an unsafe media such as Internet. The most commonly used technique is two factor based password verification. This protocol is susceptible to numerous attacks caused by human intellectual capacity of scheming and memorizing typical passwords.

Chip card based technique can be efficiently implemented in various password-based verification protocols (Lamport, 1991), (Gamal, 1985), (Kocher & Jaffe, 1999), (Messerges, Dabbish, & Sloan, 2002), (Chang C. C & Wu T. C, 1993), (Hwang M. S & Lee, 2000), (Kumar & Gupta, 2011), (Xiong & Niu, 2014) and (Kumari & Khan, 2013) easily. These have several applications like financial

DOI: 10.4018/IJDCF.2021010104

transactions, identity approval and accessing of remote services. To improve their feasibility, cards are confined in to limited size and cost. Various protocols has been reported (Tang, Hwang, & Lee, 2002), (Chang & Chang, 2005) and (Srivastva & Sharma, 2012) in which user may update password without interacting with the server, however user's identity must be same in every login attempt. Moreover, the schemes based on variable identity (Das, Saxena, & Gulati, 2004), (Wang, Liu, Xiao, & Dan, 2009), (Chang and Chang, 2009), (Madhusudhan & Mittal, 2012), (Chang, Tai & Chang, 2013), (Khan et al., 2014), (Devgan & Awasthi, 2016), (Chaudhary et al., 2015), (Wang et al., 2015), (Kharu et al., 2018), (Lu et al., 2016) and (Jung, Lee & Kim, 2016) are less prone to attacks and promising.

Literature review of the various schemes shows that till date most of the presented schemes are unsafe to different attacks like pose attack, online password guessing attack, chip card misplaced attack, repetition attack and man in middle attack. Many of the above schemes need a lot of storage cost and computational cost which decreases the performance of the scheme. Various schemes are fail to achieve all the security parameters and goals; therefore a need arises to develop a protocol that fulfils the entire above criterion. Therefore, in this paper, the authors proposed a scheme as A Novel Verification Protocol to Restrict Unconstitutional Access of Information from Smart Card.

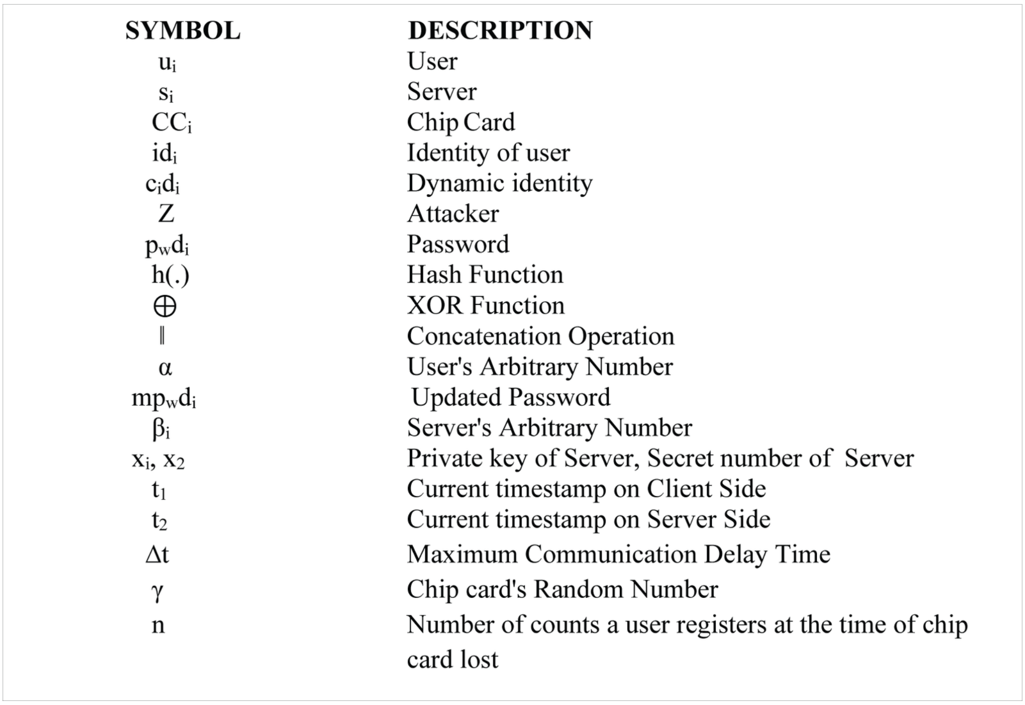
NOTATIONS AND DESCRIPTION

The following symbols/notations are preferred in this paper as described in Table 1.

Scheme Design

Initially, user enters his personal information to the terminal and sends towards the server for registration. Then user obtains chip card delivered by the server with security parameters. The

Table 1. Symbols/Notations



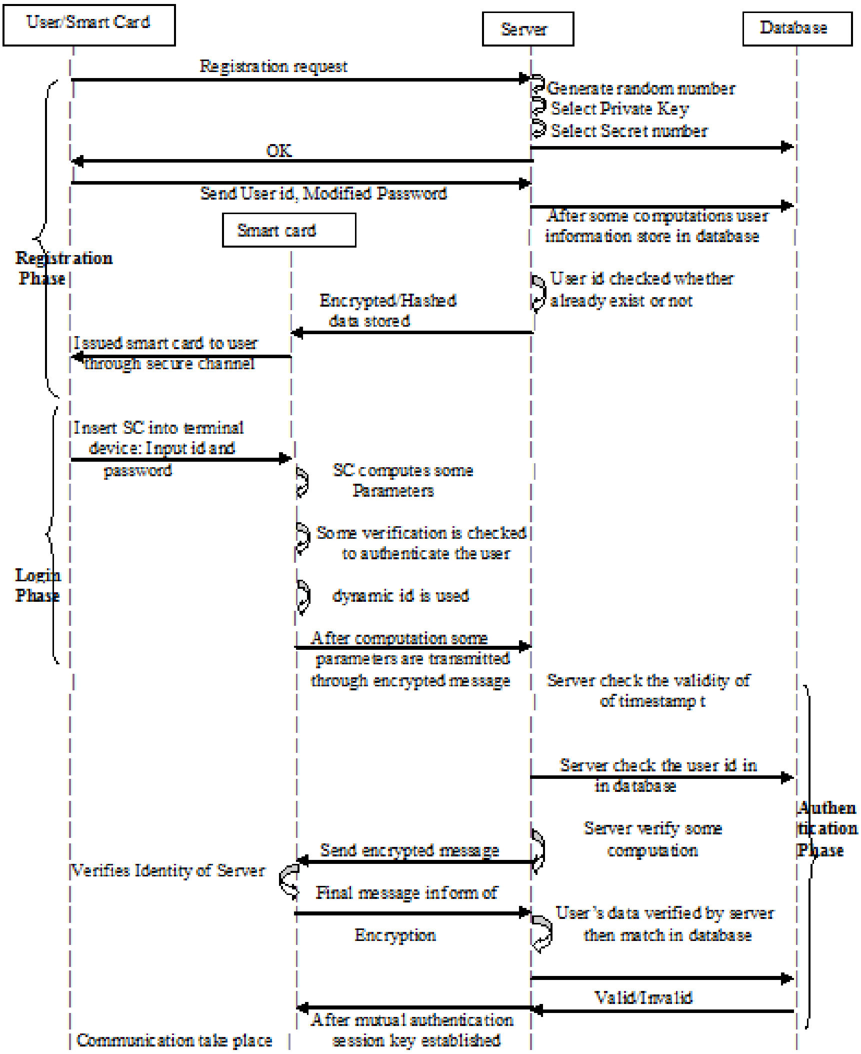
SYMBOL	DESCRIPTION
u_i	User
s_i	Server
CC_i	Chip Card
id_i	Identity of user
c_id_i	Dynamic identity
Z	Attacker
p_wd_i	Password
h(.)	Hash Function
$\oplus$	XOR Function
$\|$	Concatenation Operation
α	User's Arbitrary Number
mp_wd_i	Updated Password
β_i	Server's Arbitrary Number
x_i, x_2	Private key of Server, Secret number of Server
t_1	Current timestamp on Client Side
t_2	Current timestamp on Server Side
Δt	Maximum Communication Delay Time
γ	Chip card's Random Number
n	Number of counts a user registers at the time of chip card lost

registration phase is required only once in this protocol unless user re-registers for unavoidable conditions. Login stage requires credentials given by a user and transmits this request towards server for accessing the resources. The transmission takes place only after both, server and user validates each other. The login and authentication process usually will be carried out several times. Password update stage and misplaced chip card re-registration stage provides the service to update its own password and after re-registration, resume these services offered by the server. The process flow diagram of system design is presented in figure 1.

PROPOSED TWO FACTOR VERIFICATION SCHEME

The proposed verification scheme consists of five different phases as: registration, login, verification, password update and revocation phase. It is simple, adequate and most secure variable identity based verification scheme to resist those attacks that exist in previous schemes. To reduce memorizing cost,

Figure 1. Process flow diagram of system design


User/Smart Card
Server
Database
Registration request
Generate random number
Select Private Key
Select Secret number
OK
Send User id, Modified Password
After some computations user information store in database
Smart card
Registration Phase
User id checked whether already exist or not
Encrypted/Hashed data stored
Issued smart card to user through secure channel
Insert SC into terminal device: Input id and password
SC computes some Parameters
Some verification is checked to authenticate the user
Login Phase
dynamic id is used
After computation some parameters are transmitted through encrypted message
Server check the validity of of timestamp t
Server check the user id in in database
Authen tication Phase
Server verify some computation
Send encrypted message
Verifies Identity of Server
Final message inform of Encryption
User's data verified by server then match in database
Valid/Invalid
After mutual authentication session key established
Communication take place

computational cost as well as to maintain its high performance and efficiency along with verified safety, scheme requires OR, Ex-OR, and elementary hash functions.

Within registration stage, chip card contains data which depends on the information produces by user and several credentials inserted through server. After the registration phase, both of user and server validate to each another then after successful mutual verification both server and user communicate to each other through session key agreement protocol. In the registration or authentication stage, initial validity is checked by smart card then transmits user's information towards server for more verification. In the presented scheme, user and server together validate to each other then after session key settled among both of them and transmission takes place. If the authentication of the server or user fails, this is recommended that the login stage is denied.

Registration Stage

In the beginning of this phase u_i registers/re-registers with s_i whenever wants some services from it. Assume x_1 and x_2 are private key and secret number of the server. Here n specifies how many times a user registers by authentication server s_i. In some cases like chip card loss, theft or snatched, chip card can revoke through applying the value of n. This is saved in database of user's history on the server. The authentication server stores these secret key x_1 and number x_2 securely. The entire registration phase having a number of computation steps as follows:

Step 1: User u_i select its own id_i along with $p_w d_i$ then choose an arbitrary no. α. Compute $mp_w d_i = h(\alpha \| p_w d_i)$ then transfers $\{id_i, mp_w d_i\}$ towards s_i through a protected medium.

Step 2: These registration credentials of user are verified by server and checks the database that selected identity is earlier present or not. If selected identity is match with another in database, s_i warn u_i to select another id_i. Server also investigates registration detail of u_i then fixes the amount of n accordingly. Specify value n= 0 by s_i for unique u_i, moreover specify n=1 by s_i for re-registering user's into the server. In this way, each time of re-registration the value of n is raised by 1 after that value of n and id_i will stored in the database.

Step 3: After getting $\{id_i, mp_w d_i\}$, server selects a random number β_i, which is different for each user.

Step 4: Server estimates the values of $A_i = h(id_u \| \beta_i \| mp_w d_i)$ where $id_u = (id_i \| n)$.

Step 5: Then s_i compute $B_i = h(h(id_i) \| x_1) \oplus mp_w d_i$, $C_i = \beta_i \oplus h(h(id_i) \| x_1) \oplus mp_w d_i$ and $D_i = \beta_i \oplus h(x_2 \| x_1)$. After that stores $\{C_i, A_i, D_i, h(.)\}$ into chip card then deliver {chip card, B_i} to u_i through a protected channel.

Step 6: Subsequently obtaining {chip card, B_i} from s_i, then u_i computes $E_i = h(id_i \| pwd_i) \oplus \alpha$, $F_i = B_i \oplus \alpha$ and stores $\{E_i, F_i\}$ into chip card.

Login Stage

For obtaining services from s_i, user must login into the server by inserting its personal chip card into terminal then input its own id_i as well as pwd_i. After that chip card compute successive steps as:

Step 1: Compute $\alpha = E_i \oplus h(id_i \| p_w d_i)$, $mp_w d_i = h(\alpha \| p_w d_i)$, $h(h(id_i) \| x_1) = F_i \oplus mp_w d_i \oplus \alpha$, $\beta_i = C_i \oplus h(h(id_i) \| x_1) \oplus mp_w d_i$.

Step 2: The validity of the following equation is $A_i ? = h(id_i \| \beta_i \| mp_w d_i)$ is checked by chip card i.e it correct or not. If equation is not correct, chip card drops this session. If correct, chip card computes $h(x_2 \| x_1) = \beta_i \oplus D_i$, $B_i = F_i \oplus \alpha$.

Step 3: Chip card acquires current time-stamp t_1 then evaluate $c_i d_i = h(id_i) \oplus h(B_i \| \beta_i \| t_1)$, $B_i^{'} = B_i \oplus h(\beta_i \| t_1)$, $G_i = B_i \oplus mp_w d_i$. Further take an arbitrary no. γ then compute $H_i = h(h(id_i) \| \gamma)$, $I_i = G_i \oplus H_i$, $J_i = h(B_i \| \beta_i \| H_i \| t_1)$, $K_i = \beta_i \oplus (h(x_2 \| x_1) \| t_1)$ then transmits $\{c_i d_i, B_i^{'}, J_i, K_i, t_1, I_i\}$ to server via public channel.

Verification Stage

After receiving login request message $\{c_i d_i, B_i', J_i, K_i, t_1, I_i\}$ from u_i, server authenticates user and after proper mutual authentication, session key will be established as follows:

Step 1: Server acquires current time-stamp t_2 then justify whether t_1 is authentic means $t_2 - t_1 \leq \Delta t$. if t_1 is not correct, server deny all the login request and drop this session.
Step 2: Analyze the database for obtaining the value of n then determine $id_u = (id_i \| n)$.
Step 3: If timestamp t_1 is valid, server continues to calculate $\beta_i = K_i \oplus (h(x_2 \| x_1) \| t_1)$, $B_i = B_i' \oplus h(\beta_i \| t_1)$, $h(id_i) = c_i d_i \oplus h(B_i \| \beta_i \| t_1)$, $G_i^* = h(h(id_i) \| x_1)$, $H_i^* = G_i^* \oplus I_i$. Then verify the equation $J_i ? = h(B_i \| \beta_i \| H_i^* \| t_1)$ holds or not.
Step 4: If above equation is verified, server acquires current time-stamp t_3 then determine $a = h(G_i^* \| \beta_i \| t_3)$, further server transmits $\{a, t_3\}$ towards user.
Step 5: After receiving $\{a, t_3\}$ from server, chip card confirm the accuracy of t_3. If timestamp t_3 is correct, it will check the equation $a? = h(G_i \| \beta_i \| t_3\}$ holds or not. If this equation is correct, both u_i and s_i mutual validate to one another otherwise, this phase will be aborted by the server.
Step 6: Finally, server and user agreed upon a similar session key. The computed session key for the user is $s_k = h(G_i \| \beta_i \| t_1 \| t_3 \| h(x_2 \| x_1) \| H_i)$ and server $s_k^* = h(G_i^* \| \beta_i \| t_1 \| t_3 \| h(x_2 \| x_1) \| H_i^*)$ respectively.

This session key is used by user for secret transmission through encrypting and decrypting these messages via a secure channel.

Password Update Stage

Considering safety parameters if any user desire to modify its own password $p_w d_i$ with new password $p_w d_{new}$ in the system, user insert its own chip card into terminal of card reader further input its own id_i and $p_w d_i$. The following computation has been performed by the chip card without involvement of remote server S.

Step 1: Chip card computes the value of $\alpha = E_i \oplus h(id_i \| p_w d_i)$, $mp_w d_i = h(\alpha \| p_w d_i)$, $h(h(id_i) \| x_1) = F_i \oplus mp_w d_i \oplus \alpha$, $\beta_i = C_i \oplus h(id_i \| x_1) \oplus mp_w d_i$ then test this equation $A_i ?= h(id_i \| \beta_i \| mp_w d_i)$ is correct or not.
Step 2: In case that the condition is true, then remote user is granted to modify his pwd_i otherwise session is aborted.
Step 3: Then chip card computes $mp_w d_i^{new} = h(\alpha \| p_w d_i^{new})$, $E_i^{new} = h(id_i \| p_w d_i^{new}) \oplus \alpha$, $F_i^{new} = F_i \oplus mp_w d_i \oplus mp_w d_i^{new}$, $C_i^{new} = C_i \oplus mp_w d_i \oplus mp_w d_i^{new}$, $A_i^{new} = h(id_u \| \beta_i \| mp_w d_i^{new})$ and replaces the old $\{E_i, A_i, F_i, C_i\}$ with $\{E_i^{new}, A_i^{new}, F_i^{new}, C_i^{new}\}$ respectively. Now, the modified password has changed successfully and this phase is terminated.

Revocation Stage

This stage is initialised in the case of chip card lost, damage or misplaced situation. For revocation of chip card user forwards a request towards server. Moreover, after receiving the request server ask several credentials from user to check the authenticity, like Adhaar number, Mobile OTP, Birth date, Mother's maiden name or any other user's known value. Furthermore, after checking the validity of this revocation request created by end user, server modifies current value of n for re-registering the chip card. In each time of misplaced or lost chip card case, n is incremented by 1. Afterwards, user re-register's himself to server without updating its own identity. For the revocation of this card, it is desired from user that don't avail any earlier values like earlier password, arbitrary number otherwise by availing same values which was already stored within misplaced or stolen smart card anybody may pose as a server's legitimate user.

IMPLEMENTATION AND CODING

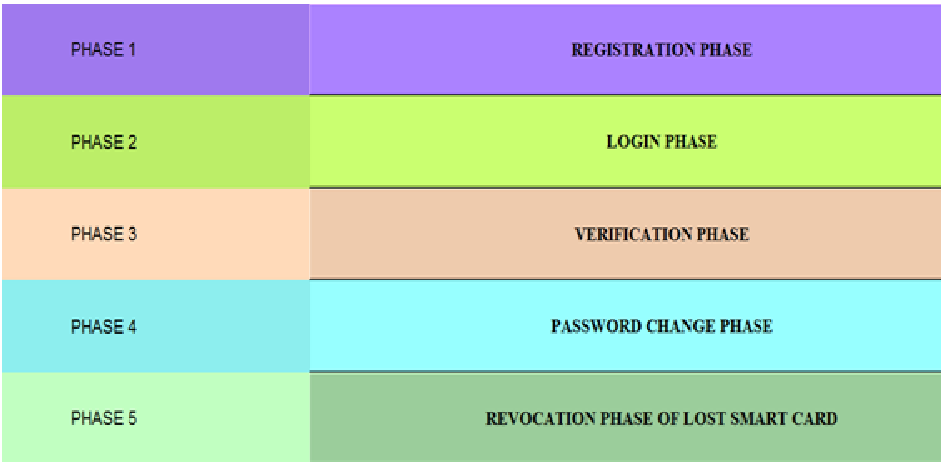
The presented scheme has been coded in PYTHON language. The screen shots of the following phases are described as shown in Figures 2-5.

The figure 2 represents all the five phases in single screen. Here phase 1 represents registration phase, phase 2 represents login phase, phase 3 represents verification phase, phase 4 represents password change phase and in the last phase 5 represents revocation phase of lost smart card.

FORMAL SAFETY VALIDATION WITH AVISPA TOOL

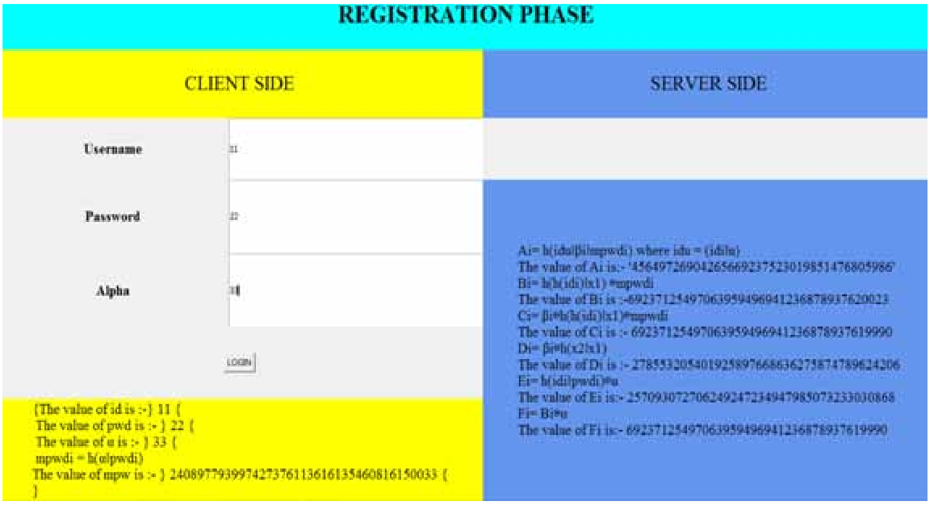
AVISPA is described as a pushdown software tool used to validate the internet security protocols. It supports High Level Protocol Specification Languages (HLPSL) and offers the formal safety

Figure 2. Different Phases of Proposed Protocol



PHASE 1	REGISTRATION PHASE
PHASE 2	LOGIN PHASE
PHASE 3	VERIFICATION PHASE
PHASE 4	PASSWORD CHANGE PHASE
PHASE 5	REVOCATION PHASE OF LOST SMART CARD

Figure 3. Registration Phase of Proposed Protocol



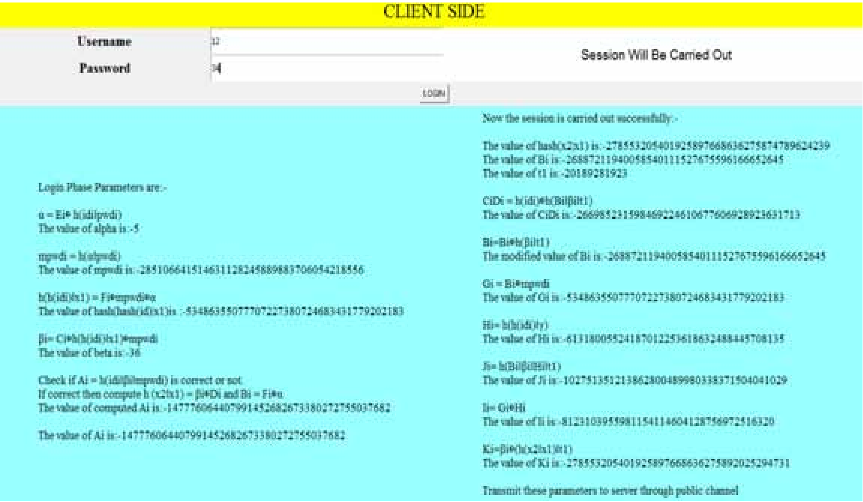
REGISTRATION PHASE

CLIENT SIDE		SERVER SIDE
Username	11	
Password	22	
Alpha	33	

```
{The value of id is :-} 11 {
 The value of pwd is :- } 22 {
 The value of α is :- } 33 {
 mpwdi = h(α||pwdi)
The value of mpw is :- } 240897793997427376113616135460816150033 {
}
```

```
Ai= h(idu||βi||mpwdi) where idu = (idi||u)
The value of Ai is:- '45649726904265669237523019851476805986'
Bi= h(h(idi)||x1) *mpwdi
The value of Bi is :-69237125497063959496941236878937620023
Ci= βi*h(h(idi)||x1)*mpwdi
The value of Ci is :- 69237125497063959496941236878937619990
Di= βi*h(x2||x1)
The value of Di is :- 278553205401925897668636275874789624206
Ei= h(idi||pwdi)*u
The value of Ei is:- 257093072706249247234947985073233030868
Fi= Bi*u
The value of Fi is:- 69237125497063959496941236878937619990
```

Figure 4. Login Phase of Proposed Protocol



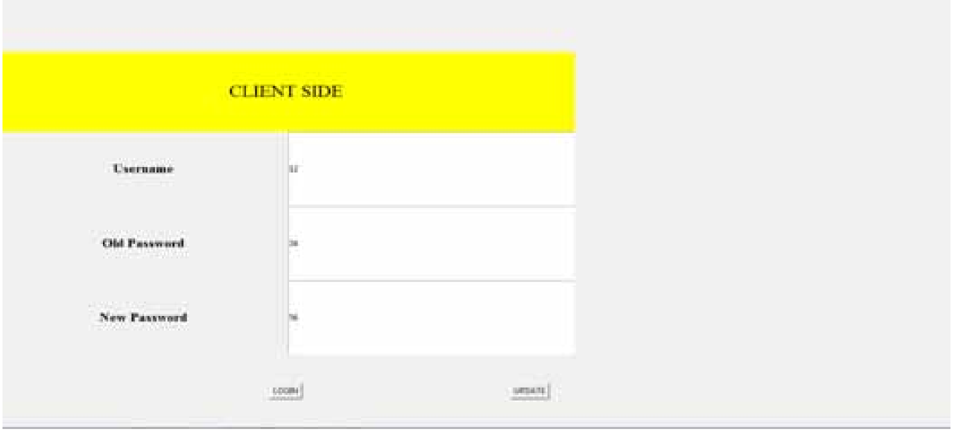
Username: 12
Password: 34

Session Will Be Carried Out

```
Login Phase Parameters are:-

α = Ei⊕ h(idi||pwdi)
The value of alpha is:-5

mpwdi = h(α||pwdi)
The value of mpwdi is:-285106641514631128245889883706054218556

h(h(idi)||x1) = Fi⊕mpwdi⊕α
The value of hash(hash(idi)|x1)is :-53486355077707227380724683431779202183

βi= Ci⊕h(h(idi)||x1)⊕mpwdi
The value of beta is:-36

Check if Ai = h(idi||βi||mpwdi) is correct or not.
If correct then compute h (x2||x1) = βi⊕Di and Bi = Fi⊕α
The value of computed Ai is:-147776064407991452682673380272755037682

The value of Ai is:-147776064407991452682673380272755037682
```

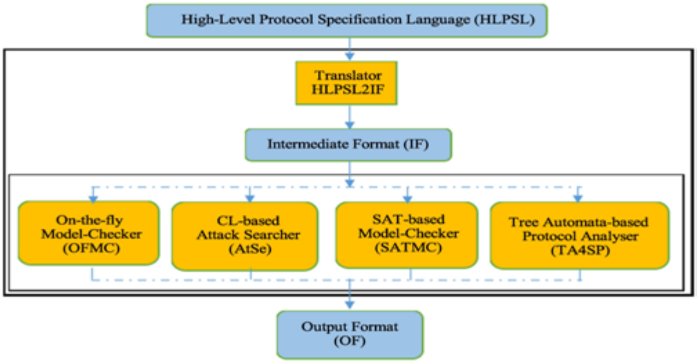
```
Now the session is carried out successfully:-

The value of hash(x2||x1) is:-278553205401925897668636275874789624239
The value of Bi is:-26887211940058540111527675596166652645
The value of t1 is:-20189281923

CiDi = h(idi)⊕h(Bi||βi||t1)
The value of CiDi is:-266985231598469224610677606928923631713

Bi=Bi⊕h(βi||t1)
The modified value of Bi is:-26887211940058540111527675596166652645

Gi = Bi⊕mpwdi
The value of Gi is:-53486355077707227380724683431779202183

Hi= h(h(idi)||y)
The value of Hi is:-61318005524187012253618632488445708135

Ji= h(Bi||βi||Hi||t1)
The value of Ji is:-102751351213862800489980338371504041029

Ii= Gi⊕Hi
The value of Ii is:-8123103955981154114604128756972516320

Ki=βi⊕(h(x2||x1)||t1)
The value of Ki is:-278553205401925897668636275892025294731

Transmit these parameters to server through public channel
```

Figure 5. Password Change Phase of Proposed Protocol



Username: 12
Old Password: 34
New Password: 56

Figure 6. Components of AVISPA tool



verification with simulated protocol. Using AVISPA, their simulation results shows that whether presented protocol is secure against the different attacks.

In order to verify the safety of presented protocol, AVISPA software tool is apply to simulate results. Therefore, presented protocol translates into HLPSL code. In this paper, the role of user, server, the session, goal and environment coded in HLPSL specification is presented below. After execution this HLPSL code into AVISPA tool, simulation results confirm that presented protocol is SAFE with respect to different active as well as passive attacks.

```
role user(
        U,S: agent,
        K1: symmetric_key,
        H,F: hash_func,
        SND,RCV: channel(dy))
played_by U
def=        local
                 State:nat,Ra,Pwd,Ai,Bi,Ci,Di,Ei,Fi,Hi,Ii,Ji,Ki,Ci
di,FiT2,Bii,Rndc,A,Tt1,Id,Tt3,Rb:text,
                T1,T3,T2,Gi:message
                 const subs1,subs4,subs5,password:protocol_id
        init State:=0
        transition
        %registration phase
        1. State=0      /\ RCV(start) =|>
           State’:=1    /\ Ra’:=new()
                       /\ Pwd’:=H(Ra’.Pwd)
                        /\ SND ({Id.Pwd’}_K1)
                        /\secret(Pwd,password,{U,S})
        2.State=1 /\ RCV({Ai’.Bi’.Ci’.Di’}_K1) =|>
         State’:=4 /\ Ei’:=xor(H(Id.Pwd’),Ra)
                    /\ Fi’:= xor(Bi,Ra)
                   /\Ra’:=xor(Ei’,H(Id.Pwd))
                  /\Pwd’:=H(Ra’.Pwd)
                  /\T1’:=xor(xor(Fi’,Pwd’),Ra’)
                  /\ Rb’:=xor(xor(Ci’,T1’),Pwd’)
                    /\ Ai’:=H(Id.Rb’.Pwd’)
                 /\ request(U,S,subs1,Ai’)
                 /\T2’:=xor(Rb’,Di’)
                 /\Bi’:=xor(Fi’,Ra)
                  /\Tt1’:=new()
                 /\Cidi’:=xor(H(Id),H(Bi’.Rb’.Tt1’))
                    /\Bii’:=xor(Bi’,H(Rb’.Tt1’))
                   /\Gi’:=xor(Bi’,Pwd’)
                    /\Rndc’:=new()
                    /\Hi’:=H(H(Id).Rndc)
                    /\Ii’:=xor(Gi’,Hi’)
                     /\Ji’:=H(Bi’.Rb’.Hi’.Tt1’)
                    /\Ki’:=xor(Rb’,(H(T2’).Tt1’))
                 /\ SND(Id.Tt1’,Cidi’.Bii’.Ji’.Ki’.Ii’)
                 /\witness(U,S,subs4,Ji’)
        3.State=4/\RCV(Id.Tt3’,A’)=|>
        State’:=6 /\ A’:=H(Gi.Rb.Tt3’)
                 /\ request(U,S,subs5,A’)
end role

role server(
        U,S: agent,
        K1: symmetric_key,
        H,F: hash_func,
        SND,RCV: channel(dy))
played_by S
def=
        local
                State:nat,Ra,Pwd,Ai,Bi,Ci,Di,Ei,Hi,Ii,Ji,Ki,Cidi,
Fi,Rb,Rndc,Hid,A,X1,X2,Tt1,Bii,Tt3,Id:text,
                T1,T3,T2,Gi:message
               const subs1,subs4,subs5,password:protocol_id
        init State:=2
        transition
        1.State=2 /\ RCV({Id.Pwd’}_K1)=|>
          State’:=3   /\ Rb’:=new()
                       /\Ai’:=H(Id.Rb’.Pwd’)
                      /\Bi’:=xor((H(H(Id).X1)),Pwd’)
                     /\Ci’:= xor(xor(Rb’,H(H(Id).X1)),Pwd’)
                     /\Di’:=xor(Rb’,H(X2.X1))
                     /\ SND ({Ai’.Bi’.Ci’.Di’}_K1)
                     /\ witness(S,U,subs1,Ai’)
                     /\secret(Pwd,password,{U,S})
2.State=3 /\RCV(Id.Tt1’,Cidi’.Bii’.Ji’.Ki’.Ii’)=|>
        State’:=5 /\ Rb’:=xor(Ki’,H(X2.X1).Tt1’)
                   /\Bi’:=xor(Bii’,H(Rb’.Tt1’))
                   /\Hid’:=xor(Cidi’,H(Bi’.Rb’.Tt1’))
                   /\Gi’:=H(H(Id).X1)
                   /\Hi’:=xor(Gi’,Ii’)
                    /\Ji’:=H(Bi’.Rb’.Hi’.Tt1’)
                  /\request(S,U,subs4,Ji’)
                   /\Tt3’:=new()
                  /\A’:=H(Gi’.Rb’.Tt3’)
                 /\SND(Id.Tt3’,A’)
                 /\witness(S,U,subs5,A’)
end role
role session(
    U,S: agent,
    K: symmetric_key,
    MD1,MD2:hash_func)
def=
        local SENDU,SENDS,RECS,RECU: channel(dy)
        composition
        user(U,S,K,MD1,MD2,SENDU,RECS) /\ server(U,S,K,MD1,MD2,SE
NDS,RECU)
end role
```

```
role environment()
        def=
        const
        subs1,subs4,subs5,password:protocol_id,
        k1,k2,k3:symmetric_key,
        u,s: agent,
        h,f: hash_func
        intruder_knowledge ={u,s,h,f,k2,k3}
        composition
        session(u,s,k1,h,f) /\ session(s,u,k1,h,f)
end role
goal
secrecy_of password
authentication_on subs1
authentication_on subs4
authentication_on subs5
end goal
environment()
HLPSL Specification of presented protocol
```

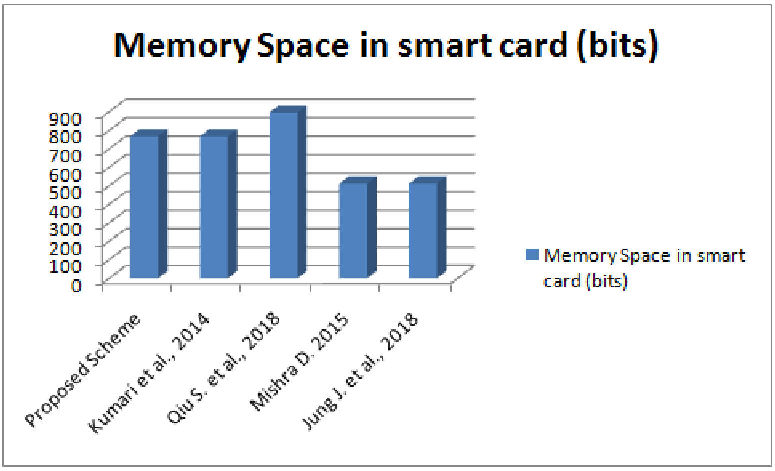
PERFORMANCE EVALUATION

The following section determines and estimates different performance criterion of presented protocol in terms of memory space, transmission cost and computation cost with other protocols suggested by various researchers. In this paper, time complexity of hash operation is represented as t_h and XOR operation as $t_\oplus$. Here, the authors suppose that some parameters as arbitrary numbers, secret numbers, password, time-stamps and identity are 128 bits. The performance evaluation regarding different schemes is shown in table 2.

Table 2. Efficiency Comparison related with Memory Space in Smart Card requirement (in bits), transmission cost (in bits) and Computational complexity cost (in bits)

Protocols	Presented Protocol	Kumari et al., (2014)	Qiu S. et al., (2018)	Mishra D. (2015)	Jung J. et al., (2018)
Memory Space in smart card (bits)	6*128=768 bits	6*128=768 bits	7*128=896 bits	4*128=512 bits	4*128=512 bits
Transmission Cost in (bits)	8*128=1024 bits	7*128=896 bits	3*128=384 bits	4*128=512 bits	4*128=512 bits
Computational Complexity Cost					
Registration Phase (User Side)	$2t_h+2t_\oplus$	$1t_h+2\ t_\oplus$	0	$1t_h$	$1t_h$
Registration Phase (Server Side)	$6t_h+4t_\oplus$	$4t_h+3\ t_\oplus$	$4t_h+\ 2t_\oplus$	$2t_h+2t_\oplus$	$5t_h+\ 1t_\oplus$
Login Phase	$12t_h+11t_\oplus$	$8t_h+10\ t_\oplus$	$4t_h+3t_\oplus$	$2t_h+2t_\oplus$	$5t_h+3t_\oplus$
Authentication Phase	$8t_h+4\ t_\oplus$	$6t_h+3\ t_\oplus$	$6t_h+1t_\oplus$	$6t_h+2t_\oplus$	$6t_h+9t_\oplus$
Password Change Phase	$9t_h+10\ t_\oplus$	$6t_h+7\ t_\oplus$	$4t_h+3t_\oplus$	$2t_h+2t_\oplus$	$7t_h+2t_\oplus$
Sum of Computational Cost	$37t_h+31t_\oplus$	$25t_h+25\ t_\oplus$	$18t_h+9t_\oplus$	$24t_h+25t_\oplus$	$24t_h+15t_\oplus$

Figure 7. Memory Space in Smart Card (bits) Comparison Graph



The memory space is described as total number of parameters stored in chip card. In this paper, total 6 variables like $\{A_i, C_i, D_i, E_i, F_i, \text{hash}\}$ are stored in chip card's memory. Therefore, memory space required by chip card to store these parameters is 6*128=768 bits.

Figure 8 shows the correlation graph of memory space needed in smart card (in bits) of presented scheme along with different other similar schemes.

In this scheme, to evaluate transmission cost, total 6 parameters $\{c_i d_i, \beta_i, J_i, K_i, t_1, I_i\}$ are required in login request, hence total bits computed as 6*128=768 bits. Furthermore, in mutual verification total number of parameters used are $\{a, t_3\}$, requires 2*128=256 bits. Hence, overhead for transmission becomes = 6*128+2*128=1024 bits. Figure 9 presents the comparison graph of transmission cost in bits of presented scheme along with various other similar schemes.

In this scheme, registration phase requires two hash functions along with two XOR function in user side. Therefore, $2t_{h(.)}+2t_{\oplus}$ is the computational complexity in user's side. Correspondingly, server requires six hash functions along with four XOR function during the registration phase, therefore, at server side, computational complexity is $6t_{h\,(.)}+4t_{\oplus}$.

Figure 8. Transmission Cost (in bits) Comparison Graph

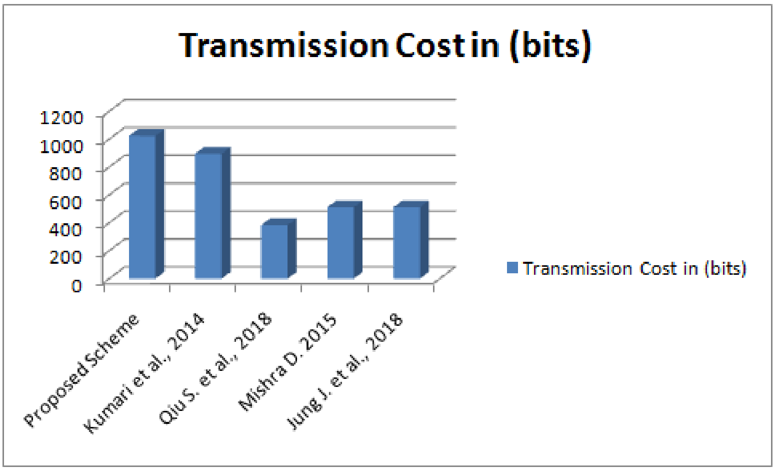
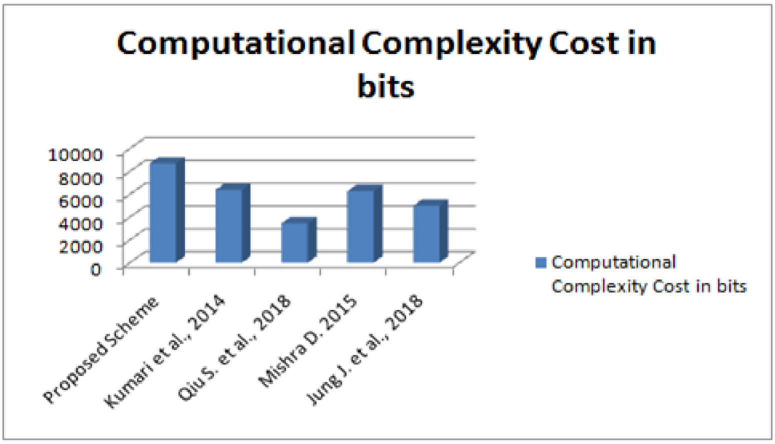


Figure 9. Computational Complexity Cost (in bits) Comparison Graph



In this scheme, login stage requires 12 hash functions along with 11 XOR functions for login request. Hence, $12t_{h\,(.)}+11t_{\oplus}$ is the computational cost for login message.

In this scheme, mutual authentication phase requires 1 hash function and server requires 7 hash functions along with 4 XOR functions. Therefore, total computational cost for authentication phase requires $8h_{\,(.)}+ 4t_{\oplus}$ operations.

Figure 9 represents comparison graph of computational complexity evaluation cost in (bits) of this scheme corresponding with various other similar schemes.

CONCLUSION

In this paper, the presented scheme ensures security, privacy and confidentiality of a user. This scheme is an improvement over all the schemes presented in the literature review. After analysis, it is observed that earlier work is unsafe for practical applications because all security parameters can be easily obtained by the challenger and vulnerable to smart card misplaces violation as well as user un-traceability violation attack. Moreover, an adversary can get server's secret key, password of the entire registered user's and also the session key of the server, may also be obtained by an adversary which may lead to destroying the whole system. The presented scheme has been coded in PYTHON language and tested into AVISPA tool. The simulation results concluded that presented protocol is safe against entire active and passive attacks and achieve all the goals. The efficiency comparison of the scheme has confirmed its feasibility and performance to the practical approach. The presented scheme can be applied in such applications which providing privacy protection with low-computation-ability devices. Thus, our idea is practically more acceptable to operate secure remote access over the public environment as well as may be simply integrated into various types of services such as Military, Academics, Aeronautics, Banking, Crime control departments and Business applications.

REFERENCES

Chang, C. C., & Wu, T. C. (1993). Remote Password Authentication with Smartcards. *Proceeding of Computers and Digital Techniques*, *138*(3), 165–168. doi:10.1049/ip-e.1991.0022

Chang, Y. F., & Chang, C. C. (2005). Authentication Schemes with No Verification Table. *Applied Mathematics and Computation*, *167*(2), 820–832. doi:10.1016/j.amc.2004.06.118

Chang, Y. F., & Chang, H. C. (2009). Security of Dynamic ID-based Remote User Authentication Scheme. *Fifth International Joint Conference on INC, IMS and IDC*, 2108–2110.

Chang, Y. F., Tai, W. L., & Chang, H. C. (2013). Untraceable Dynamic-Identity-based Remote User Authentication Scheme with Verifiable Password Update. *International Journal of Communication Systems*, *27*(11), 3430–3440. doi:10.1002/dac.2552

Chaudhary, S. A., Farash, M. S., N., Kumari, S., & Khan, M. K. (2015). An Enhanced Privacy Preserving Remote User Authentication Scheme with Provable Security. *The Journal of Security Communication Networks*.

Das, M. L., Saxena, A., & Gulati, V. P. (2004). A Dynamic ID-based Remote User Authentication Scheme. *IEEE Transactions on Consumer Electronics*, *50*(2), 639–931. doi:10.1109/TCE.2004.1309441

Devgan, S., & Awasthi, A. K. (2016). Security Enhancement of an Improved Remote User Authentication Scheme with Key Agreement. *Wireless Personal Communications*.

El Gamal, T. (1985). A Public-key Cryptosystem and a Signature Scheme based on Discrete Algorithms. *IEEE Transactions on Information Theory*, *31*(4), 469–472. doi:10.1109/TIT.1985.1057074

Hwang, M. S., & Li, L. H. (2000). New Remote User Authentication Scheme using Smart Cards. *IEEE Transactions on Consumer Electronics*, *46*(1), 28–30. doi:10.1109/30.826377

Jung, J., Lee, D., & Kim, J. (2016), *Cryptanalysis and Improvement of Efficient Password-based User Authentication Scheme using Hash Function*. ACM. doi:.<ALIGNMENT.qj></ALIGNMENT>10.1145/2857546.2857570

Jung, J., Lee, D., Lee, H., & Won, D. (2018). Security Enhanced Anonymous User Authenticated Key Agreement Scheme using Smart Card. *Journal of Electronic Science and Technology*, *16*(1), 45–49.

Khan, M. K., Kumari, S., Wang, X. M., & Kumar, R. (2014). Dynamic ID-based Authentication Scheme. *Proceeding of 12th International Conference on Dependable, Autonomic and Secure Computing (DASC)*, 347-361.

Khari, M., Shrivastava, G., Gupta, S., & Gupta, R. (2018). Role of Cyber Security in Today's Scenario. In *Cyber Security and Threats: Concepts, Methodologies, Tools, and Applications* (pp. 1–15). IGI Global. doi:10.4018/978-1-5225-5634-3.ch001

Kocher, P., Jaffe, J., & Jun, B. (1999). Differential Power Analysis. *Advances in Cryptology–CRYPTO*, 388–397.

Kumar, M., Gupta, M. K., & Kumari, S. (2011). An Improved Efficient Remote Password Authentication Scheme with Smart Card over Insecure Networks. *International Journal of Network Security*, *13*(3), 167–177.

Kumari, S., & Khan, M. K. (2013). Cryptanalysis and Improvement of a Robust Smart-Card based Remote User Password Authentication Scheme. *International Journal of Communication Systems*, 3939–3955.

Kumari, S., Khan, M. K., & Li, X. (2014). An Improved Remote User Authentication Scheme with Key Agreement. *Computers & Electrical Engineering*, *40*(6), 1997–2012. doi:10.1016/j.compeleceng.2014.05.007

Lamport, L. (1991). Password Authentication with Insecure Communication. *Communications of the ACM*, *24*(11), 770–772. doi:10.1145/358790.358797

Lu, Y., Li, L., Peng, H., & Yang, Y. (2016). A Secure and Efficient Mutual Authentication Scheme for Session Initiation Protocol, Peer-to-Peer Net. *App.*, *9*(2), 449–459.

Madhusudhan, R., & Mittal, R. C. (2012). Dynamic ID-based Remote User Password Authentication Schemes using Smart Cards. A Review. *Journal of Network and Computer Applications*, *35*(4), 1235–1248. doi:10.1016/j.jnca.2012.01.007

Messerges, T. S., Dabbish, E. A., & Sloan, R. H. (2002). Examining Smart-card Security under the Threat of Power Analysis Attacks. *IEEE Transactions on Computers*, *51*(5), 541–552. doi:10.1109/TC.2002.1004593

Mishra, D. (2017). Efficient and Secure two-factor Dynamic ID-based Password Authentication Scheme with Provable Security. *Cryptologia*. Advance online publication. doi:10.1080/01611194.2017.1325787

Qiu, S., Xu, G., Ahmad, H., & Guo, Y. (2018). An Enhanced Password Authentication Scheme for Session Initiation Protocol with Perfect Forward Secrecy. *PLoS One*, *13*(3), e0194072. doi:10.1371/journal.pone.0194072 PMID:29547619

Shrivastava, G., Sharma, K., & Bawankan, A. (2012). A New Framework Semantic Web Technology based e-learning. *Environment and Electrical Engineering (EEEIC), 11th International Conference on IEEE*, 1017-1021. doi:10.1109/EEEIC.2012.6221527

Tang, Y. L., Hwang, M. S., & Lee, C. C. (2002). A Simple Remote User Authentication Scheme. *Mathematical and Computer Modelling*, *36*(1-2), 103–107. doi:10.1016/S0895-7177(02)00106-1

Tu, H., Kumar, N., Chilamkurti, N., & Rho, S. (2015). An Improved Authentication Protocol for Session Initiation Protocol using Smart Card. Peer-to-Peer Net. *Appl.*, *8*(5), 903–910.

Wang, D., He, D., Wang, P., & Chu, C. (2015). Anonymous two factor Authentications in Distributed Systems: Certain Goals are beyond Attainment. *IEEE Transaction of Security & Computationality*, *12*(4), 428–442. doi:10.1109/TDSC.2014.2355850

Wang, Y. Y., Liu, J. Y., Xiao, F. X., & Dan, J. (2009). A More Efficient and Secure Dynamic ID-based Remote User Authentication Scheme. *Computer Communications*, *32*(4), 583–585. doi:10.1016/j.comcom.2008.11.008

Xiong, L., Niu, J. W., Liu, Y., Liao, J., & Liang, W. (2014). Robust Dynamic ID-based Remote User Authentication Scheme using Smart Cards. *International Journal of Ad Hoc and Ubiquitous Computing*, *17*(4), 254–264. doi:10.1504/IJAHUC.2014.066423

Ajay Kumar Sahu is an Assistant Professor in the Department of Computer Science & Engineering at Raj Kumar Goel Institute of Technology & Management (RKGITM), Ghaziabad, affiliated to Dr. Abdul Kalam Technical University, Luck now (AKTU), Uttar Pradesh, India. His area of research interest includes Global Information Systems, Organizational Impact of IT, Software Development, Cloud Computing, Network Security and Software testing. He received his M. Tech degree from Guru Gobind Singh Indraprstha University (GGSIPU), Delhi in 2009 and B. Tech. (CSE) from G.L.A institute of Technology, Mathura in 2003. He has total Fifteen years of teaching experience in Academics. He has published more than ten research papers in international journals and proceedings.

Ashish Kumar specializes in Mobile AdHoc Network. He completed his B. Tech from Vikram University-Ujjain, and earned M. Tech from IIT-Kharagpur in Computer Science Engineering. He was awarded Ph. D. from UPES with accolades. His area of research focuses upon investigation of energy consumption in MANET. He has over one-and-a-half-decade experience in training in Object Oriented Techniques with UML, Operating Systems, Software Engineering, Mobile Computing and Distributed System. His interest is in providing sustainable solutions in the field of MANET, Reverse Engineering and Object Oriented Design. Dr. Ashish is a life member of IACSIT, IAENG, and ISTE.