Pertanika J. Sci. & Technol. 27 (1): 175 - 192 (2019)

SCIENCE & TECHNOLOGY

Journal homepage: http://www.pertanika.upm.edu.my/

SPAS: An Authentication Scheme to Prevent Unauthorized Access of Information from Smart Card

Ajay Kumar Sahu[1]* and Ashish Kumar[2]

[1]Department of Computer Science and Engineering,
Raj Kumar Goel Institute of Technology and Management, Ghaziabad 201004,
Uttar Pradesh, India
[2]Department of Computer Science and Engineering, ITS, Greater Noida 201306,
Uttar Pradesh, India

ABSTRACT

Nowadays internet has become indispensable part of one’s life. Therefore, security and privacy are of critical concern to retain user’s confidence in network services and applications. Several password verification based schemes/protocols have been used for authentication over insecure channel to protect resources from unauthorized access in networked environment. However, the schemes were not fault tolerant. Also, the feasibility for implementation in some of the applications was questionable. Therefore, we have devised a scheme SPAS (Secure and Provable Authentication Scheme) to overcome the issues prevalent in existing schemes. The objective was to keep the computational and communication cost low. The analysis of the presented scheme SPAS over existing schemes corroborates its effectiveness in tackling various attacks and uniqueness. Further, the performance analysis of the presented scheme is also given to strengthen the proposal.

Keywords: Hash function, identity; security, information retrieval, key agreement, mutual authentication, password, smart card

ARTICLE INFO

Article history:
Received:18 April 2018
Accepted:12 September 2018
Published: 24 January 2019

E-mail addresses:
ajay4989@gmail.com (Ajay Kumar Sahu)
ashishcse29@gmail.com (Ashish Kumar)
* Corresponding author

INTRODUCTION

Growing usage and applications of technology in all possible domains of human lives is posing a challenge. It has become hard to authenticate the user. Many online frauds are reported in the literature accounted for significant financial loss.

ISSN: 0128-7680
e-ISSN: 2231-8526

Consequently, the need arises to authenticate the genuine users over an insecure channel like Internet. The most commonly used technique is password based authentication protocol (ElGamal, 1985; Kocher et al., 1999; Lamport, 1981; Tang et al., 2002). The password based techniques are prone to various attacks due to human cognitive capability of designing and remembering complex passwords. Using a smart card, having various advantages like portability, low cost and availability. By using the smart card we can practically and efficiently implement these password based authentication protocols easily (Chang & Wu, 1991; Chien et al., 2002; Hwang & Li, 2000; Ku & Chen, 2004; Messerges et al., 2002; Yang & Shieh, 1999) easily. Several protocols considering static identity have been proposed in the literature (Karuppiah & Saravanan, 2014; Karuppiah & Saravanan, 2015; Kumar et al., 2011; Song, 2010; Wang et al., 2014). Static identity based protocols are easier targets for the attackers. Recently, researchers have designed schemes (Chang & Chang, 2009; Chang et al., 2013; Das et al., 2004; Khan et al., 2014; Li et al., 2014; Wen & Li, 2011; Wang et al., 2009) based on dynamic identity to manage the static identity problem. However, the schemes suffers with high storage and computation cost. In some cases they were shown to be unsafe. Kumari et al. (2014), stated that Chang et al. (2013) protocol was susceptible to various attacks like impersonation, password guessing and anonymity violation. Later suggested an enhanced protocol ("An improved remote user authentication scheme with key agreement") to overcome existing security and identity concerns. However, analysis of the protocol proposed by Kumari et al. (2014) revealed that it violated on certain aspects i.e. chip/smart card loss and user anonymity and compromised the offered safety against various attacks. In this protocol attacker may leak the privacy and can steal server's secret key as well as password of the authenticate user, which may impact the entire system. Therefore, we have designed and proposed an efficient, Secure and Provable Authentication Scheme (SPAS), to overcome the observed security weaknesses.

The remaining section in this paper is structured in a systematic way as follows: Notations, System design and Capabilities of adversary are shown in as part 2; we proposed SPAS (Secure and Provable Authentication Scheme) in part 3; Security analysis of suggested scheme is illustrated in part 4; Performance evaluation is shown in part 5 and lastly, conclusion of our scheme is discussed.

NOTATIONS, SYSTEM DESIGN AND CAPABILITIES OF ADVERSARY

In this section, initially basic notations are described then demonstrate system design and capabilities of adversary being used in the scheme.

Notations and Description

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

Table 1

Symbols/Notations

Symbol	Description
u_i	Remote user
s_i	System server
sc_i	Smart Card
id_i	User’s identity
c_id_i	User’s variable identity
p_wd_i	User’s password
α	Random number chosen from user
mp_wd_i	Encrypted/modified password
β_i	Random number chosen by Server for user
x_1, x_2	Server’s private key and Server’s secret number
h(.)	Hash operation
‖	Concatenation process
$\oplus$	XOR operation
t_1	User’s current timestamp
t_2	Server’s current timestamp
Δt	maximum transmission delay time
γ	Random Number Chosen by smart card
n	Number of times user registers in case of smart card lost
Z	Adversary /Attacker

System Design

Our scheme includes five fundamental stages as described in section 3. Initially, user’s information is fed to the terminal and forwarded to the server for registration. Thereafter, server delivers the chip card to remote user along-with security parameters. The registration phase is required only once in our scheme unless and otherwise required re-registration in unavoidable conditions. For accessing resources, permission is given by the server once the credentials are verified in Login stage. The communication takes place only after both, the server and user validates each other. The login and authentication process is usually done several times. In the scheme, a two factor authentication process is used to ensure that chip card owner can be allowed the accessing of the server. Password update and chip card loss re-registration stages are the very useful approaches to overcome the security issues in our proposed scheme.

Capabilities of the Adversary

According to various researches (Chang & Chang, 2005; Wang et al., 2014) of two-factor authentication, the adversary Z has the following capabilities:

In the open communication channel, the adversary Z having full control as it can intercept, remove, update or again send the eavesdropped messages over open communication media.

Adversary Z can get the password of an authenticate user by an infected card reader or obtain the smart card parameters but both cannot be achieved simultaneously.

Adversary Z can also obtain the servers secret key in the case of forwarding secrecy.

THE PROPOSED AUTHENTICATION SCHEME: SPAS

The working of proposed SPAS (Secure and Provable Authentication Scheme) is shown in Figure 1. It consists of five stages namely: 1. Registration stage, 2. Login stage, 3. Verification stage, 4. Password update stage and 5. Re - registration stage. The SPAS is a simple, adequate and most secure based on variable identity authentication scheme to prevent those attacks that exist in the scheme (Kumari et al., 2014). To reduce storage cost, computing complexity as well as to maintain its high performance along with certified security, we need OR, Ex-OR, and elementary hash operations in our scheme.

Within registration stage, the smart card contains data which is dependant on the information produces through user and several credentials inserted through server. In the login/verification phase, both the user and server authenticate each other and after successful mutual authentication, a session key is established before communication between them as various schemes (Kumari & Khan, 2013; Tu et al., 2015; Lu et al., 2015). The initial validity is checked by the smart card then sends user's information towards server for more verification. If the authentication of the server or user fails, this is recommended that the login stage is denied.

User's Registration Stage

In the beginning of this phase u_i registers/re-registers with s_i whenever want some services from it. Assume x_1 and x_2 are private keys and a secret number of the s_i. Suppose n specifies how many times a user registers by authentication server s_i. In mishap-penning cases like chip card loss, theft or snatched, the chip card can revoke through applying the value of n and value of n is saved in a database of user's history on s_i. 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:

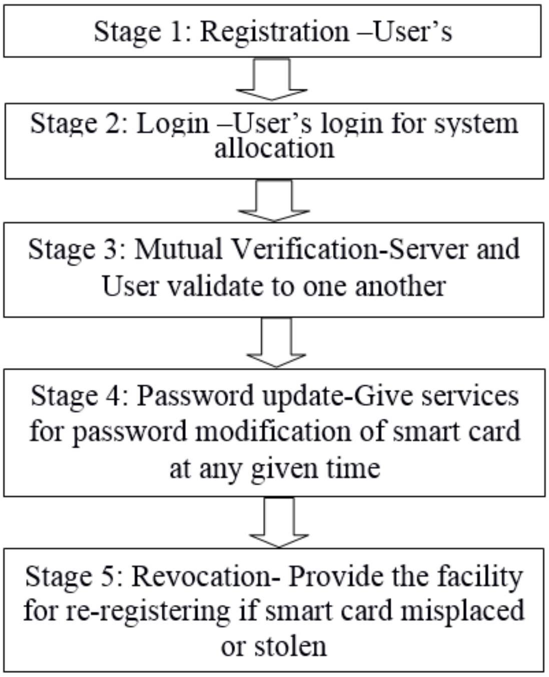
Stage 1: Registration –User's
Stage 2: Login –User's login for system allocation
Stage 3: Mutual Verification-Server and User validate to one another
Stage 4: Password update-Give services for password modification of smart card at any given time
Stage 5: Revocation- Provide the facility for re-registering if smart card misplaced or stolen

Figure 1. Methodology Used

- User u_i select id_i, p_wd_i and an arbitrary number α.
- Compute $mp_wd_i= h(\alpha\|p_wd_i)$ then transfers $\{id_i, mp_wd_i\}$ towards $s_{i.}$
- The registration credentials of user are verified by the server. If selected identity is matched with another in the database, s_i warns u_i to select another id_i.
- Set n =0 by s_i for unique u_i, moreover, specify n =1 by s_i for re-registering user's into the server. Increment n each time of re-registration then id_i will be stored in the database.
- After getting $\{id_i, mp_wd_i\}$, server selects a random number β_i, which is different for each user.
- Server computes $A_i= h(id_u\|\beta_i\|mp_wd_i)$, where $id_u= (id_i\|n)$, $B_i= h(h(id_i)\|x_1) \oplus mp_wd_i$, $C_i= \beta_i\oplus h(h(id_i)\|x_1)\oplus mp_wd_i$ and $D_i= \beta_i\oplus h(x_2\|x_1)$.
- Stores $\{C_i, A_i, D_i, h(.)\}$ into chip card and deliver$\{chip\ card, B_i\}$ to u_i.
- 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.

User's Login Stage

For obtaining services from s_i, a user must login into the server. For this, it must insert its personal chip card into device then insert own id_i as well as pwd_i, further:

- Compute $\alpha = E_i \oplus h(id_i \| p_w d_i)$, $mp_w d_i = h(\alpha \| p_w d_i)$, $h(h(id_i) \| x_1) = F_i \oplus mp_w d_i \oplus \alpha$, $\beta_i = C_i \oplus h(h(id_i) \| x_1) \oplus mp_w d_i$.
- Smart card checks $A_i \,?= h(id_i \| \beta_i \| mp_w d_i)$ i.e it is correct or not.
- If not correct, chip card drops this session. If correct, computes $h\,(x_2 \| x_1) = \beta_i \oplus D_i$, $B_i = F_i \oplus \alpha$.
- Smart card acquires current time-stamp t_1, computes $c_i d_i = h(id_i) \oplus h(B_i \| \beta_i \| t_1)$, $B_i^{'} = B_i \oplus h(\beta_i \| t_1)$, $G_i = B_i \oplus mp_w d_i$.
- Select arbitrary no. γ, compute $H_i = h(h(id_i) \| \gamma)$, $I_i = G_i \oplus H_i$, $J_i = h(B_i \| \beta_i \| H_i \| t_1)$, $K_i = \beta_i \oplus (h(x_2 \| x_1) \| t_1)$.
- Transmits $\{c_i d_i, B_i^{'}, J_i, K_i, t_1, I_i\}$ to server via public channel.

Verification Stage

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

- Server set current time instant t_2, verify t_1 is authentic or not, means $t_2 - t_1 \leq \Delta t$.
- If t_1 is not correct, server denies all the login request and drop this session.
- For obtaining the value of n in database, determine $id_u = (id_i \| n)$.
- If timestamp t_1 is valid, server compute $\beta_i = K_i \oplus (h(x_2 \| x_1) \| t_1)$, $B_i = B_i^{'} \oplus h(\beta_i \| t_1)$, $h(id_i) = c_i d_i \oplus h(B_i \| \beta_i \| t_1)$, $G_i^* = h(h(id_i) \| x_1)$, $H_i^* = G_i^* \oplus I_i$.
- Verify equation $J_i ?= h(B_i \| \beta_i \| H_i^* \| t_1)$ holds or not.
- If true, server acquires current timestamp t_3, compute $a = h(G_i^* \| \beta_i \| t_3)$, transmits $\{a, t_3\}$ towards the user.
- After getting $\{a, t_3\}$ from the server, smart card verifies validity of t_3.
- If timestamp t_3 is true, check the equation $a \,?= h(G_i \| \beta_i \| t_3\}$ holds or not. If correct, both u_i and s_i mutual validate to one another otherwise, will be aborted by the server.
- Server and user agreed upon a common session key. Compute 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 $s_k^* = h(G_i^* \| \beta_i \| t_1 \| t_3 \| h(x_2 \| x_1) \| H_i^*)$.

Password Update Stage

Considering security parameters, if any user desires to modify its own password $p_w d_i$ with new password $p_w d_{new}$ in the system, user inserts its own chip card into a device of chip card reader then input its own id_i and $p_w d_i$. The following computation has been performed by the smart card without the involvement of remote server S.

- Smart card computes $\alpha = E_i \oplus h(id_i \| p_w d_i)$, $mp_w d_i = h(\alpha \| p_w d_i)$, $h(h(id_i) \| x_1) = F_i \oplus mp_w d_i \oplus \alpha$, $\beta_i = C_i \oplus h(id_i \| x_1) \oplus mp_w d_i$.
- Verify equation $A_i \,?= h(id_i \| \beta_i \| mp_w d_i)$ is correct or not.
- If true, a user is permitted to modify his pwd_i otherwise session is aborted.

- Smart card computes $mp_wd_i^{new}=h(\alpha \| p_wd_i^{new})$, $E_i^{new}=h(id_i \| p_wd_i^{new})\oplus\alpha$, $F_i^{new}=F_i\oplus mp_wd_i\oplus mp_wd_i^{new}$, $C_i^{new}=C_i\oplus mp_wd_i\oplus mp_wd_i^{new}$, $A_i^{new}=h(id_u \| \beta_i \| mp_wd_i^{new})$.
- Replaces old $\{E_i, A_i, F_i, C_i\}$ with $\{E_i^{new}, A_i^{new}, F_i^{new}, C_i^{new}\}$.
- Hence, modified password has changed successfully and session is terminated.

Revocation Stage of Lost Smart Card

When any user misplaces smart card then it forwards an application towards server for its revocation. At that moment server asks some credentials from the user to check the authenticity, like *Adhaar number*, *Mobile OTP*, *Birth date*, a *card number* of identity proof, *Voter identity*, Name of *mother's maiden* or any other user's known value. Subsequently after checking the validity of the revocation request generating by user server updates existing value of *n* for re-registering the chip card. In each time of misplaced or lost smart card case, *n* is incremented by 1. Then after, user re-registers with the server without updating its own identity. For the revocation/re-registration of a card, it is expected from a customer that does not avail any earlier values like earlier password, an arbitrary number otherwise by availing same values which are already stored within misplaced or stolen smart card anybody may masquerade as a server's legitimate user.

SECURITY ANALYSIS OF PROPOSED SCHEME

Here we have analyzed SPAS scheme and shown that this scheme is robust and secure against various attacks:

Provide User's Un-traceability and Anonymity

In the scheme, it is hard to track privacy/identity and the claim is corroborated by the analysis of the scheme.

- When attacker *Z* receives $\{E_z, F_z, C_z, A_z, D_z\}$parameters from smart card then calculate $\alpha_z=E_z\oplus h(id_z \| p_wd_z)$, $mp_wd_z=h(\alpha_z \| p_wd_z)$, $h(h(id_z)) \| x_1=F_z\oplus mp_wd_z\oplus\alpha_z$, $\beta_z=h(h(id_z) \| x_1)\oplus C_z\oplus mp_wd_z$, $h(x_2 \| x_1)=\beta_z\oplus D_z$. Hence, attacker can achieve the parameter value of $h(x_2 \| x_1)$.
- Any login message as $\{c_id_i, B_i^{'}, J_i, K_i, t_1, I_i\}$ intercepted by attacker can calculate these parameters like $\beta_i=K_i\oplus(h(x_2 \| x_1) \| t_1)$, $B_i=B_i^{'}\oplus h(\beta_i \| t_1)$, $h(id_i)=c_id_i\oplus h(B_i \| \beta_i \| t_1)$. As a result, an attacker can obtain the $h(id_i)$ in place of original *id* of any legal end user. Hence, confirm that the scheme maintain user's un-traceability and anonymity.

Resist Off-line Password Assumption Attack

In such type of violation, attacker may guess password by applying user's smart card in offline manner. Here, attacker may arrange smart card of any user from either stolen/misplaced, then compute the following:

- An attacker can obtain the value of α_i from above section then compute $E_i \bigoplus \alpha_i = h(id_i \| p_w d_i)$.
- From $h(id_i \| p_w d_i)$, the attacker cannot obtain the original password without knowing id_i of the user.

Prevent Chip Card Loss/Misplaced Attack

Suppose adversary get hold of the chip card of legal user and rival succeeds in obtaining entire information, in such case the scheme will show that Z cannot get favourable information. From the above two sections, anyone can recognize values of $\{h(id_i), B_i, \beta_i, \alpha_i\}$from attacks but not $\{id_i, p_w d_i\}$, so $G_i = h(h(id_i) \| x_1)$ cannot be computed and x_1 (secret key) cannot be known. Further, the value of parameter of $I_i = G_i \bigoplus H_i$ cannot be computed without G_i. Suppose at any time the server receives the message $\{c_i d_i, B_i', J_i^{**}, K_i, t_i, I_i^{**}\}$ from attacker, but the server will not accept this login request as $J_i^{**}? \neq J_i^{*}$ for different G_i. Hence, our scheme can prevent the chip card loss/misplaced attack.

Prevent User Mask and Server Pose Violation

Any attacker Z needs the correct login request $\{c_i d_i, B_i', J_i, K_i, t_1, I_i\}$ if willing to mask an authenticate customer for the server. As described in previous section, no attacker can satisfy the authentication equation of $J_i^{**} = J_i^{*}$ successfully without G_i^{*}. Hence, the suggested scheme avoids user mask violation. Assume that, any adversary computes the valid value of $a = h(G_i^{*} \| \beta_i \| t_3)$, then masquerading attack is possible in server but attacker cannot obtain correct value of a as confidential key x_1 or G_i cannot be achieved. Hence, suggested scheme prevents server posed violations.

Prevent Forward Secrecy

This is used as very helpful tool for providing secure information to users. Here, with the help of following parameters $\{h(id_i \| x_1), \beta_i, h(x_2 \| x_1), h(h(id_i) \| \gamma)\}$, session key is obtained by adversary as: $s_k = h(h(id_i) \| x_1) \| \beta_i \| t_1 \| t_3 \| h(x_2 \| x_1) \| h(h(id_i) \| \gamma))$ that means it is necessary for attacker to know the parameter value of $h(h(id_i) \| \gamma)$. In our scheme, if an attacker can obtain $\{h(id_i \| x_1), \beta_i, h(x_2 \| x_1), h(id_i)\}$, but cannot obtain the number γ anyhow, since it is computed arbitrary in each session. Hence, early session keys cannot be achieved by adversary accurately.

Prevent Replay Violation

To prevent replay attack, we use the concept of current timestamps in scheme. When server receives the request of login message as in the form $\{c_i d_i, B_i', J_i, K_i, t_1, I_i\}$, the validity of timestamp t_1 is checked by server immediately. Likely, the response message is received by user as $\{a, t_3\}$ from server, then verify authenticity of t_3 firstly. If time instant was not appropriate, login request messages will be denied by both user and server. So, the suggested scheme prevents replay violation.

Resist Conspirator Attack

In view of the suggested idea, user transfers modified password in place of plain text. A random number α is used by the user for protecting password against conspirator's attack and compute the value of modified password as $mp_wd_i=h(\alpha\|p_wd_i)$. Here, attacker doesn't know the value of $\{\alpha, p_wd_i\}$ both, and cannot guess both of them simultaneously in polynomial time. Hence, in the scheme, there is no opportunity for prediction of possible password and check its prediction is correct. So, the scheme prevents from insider attack.

Maintain Mutual Verification

The suggested idea describe that validity of legal user verified through server by checking the validity of the equation $J_i = h(B_i\|\beta_i\|H_i^*\|t_1)$? In the same way, the user check the authenticity of the server by verifying the equation $a? = h(G_i\|\beta_i\|t_3)$. In this manner to provide proper and secure communication both server and user validate to each other. Hence, suggested scheme must maintain mutual verification.

Provide Session Key Establishment

Here, session key $s_k=h(G_i\|\beta_i\|t_1\|t_3\|h(x_2\|x_1)\|H_i)$ is computed by user and the session key $s_k^*=h(G_i^*\|\beta_i\|t_1\|t_3\|h(x_2\|x_1)\|H_i^*)$ is computed by server in the last of each session. Hence, both of user and server can exchange their information securely and user will access the desired services from server safely because this scheme also provides forward secrecy, that is why our scheme must provide reasonable along with safe session key.

PERFORMANCE ANALYSIS

Here we measure and evaluate various performance parameters of the proposed SPAS scheme i.e. the storage capacity, communication cost, computational cost and security parameters under various known attacks in contrast to other schemes (Kumari et al., 2014; Kaul & Awasthi, 2016; Chaudhary et al., 2015; Jung et al., 2016). Suppose time complexity of hash function is t_h moreover time complexity of XOR function is $t_{\oplus}$. In our scheme, we made some assumptions on parameters like random numbers, secret numbers, identity, password, and time-stamps i.e. 128-bits. The efficiency comparisons regarding various schemes are described in Table 2. Aforementioned table analyzes and computes storage cost, communication cost along with computational complexity cost over various schemes with our proposed scheme and sum-up in the last.

Table 2

Efficiency Comparison related with Memory requirements (in bits), Transmission cost (in bits) and Computational complexity cost (in bits)

Protocols	Proposed Scheme	Kumari et al. (2014)	Kaul and Awasthi (2016)	Chaudhary et al. (2015)	Jung et al. (2016)
Memory Space in smart card (bits)	6*128=768 bits	6*128=768 bits	5*128=640 bits	7*128=896 bits	4*128=512 bits
Transmission Cost (bits)	8*128=1024 bits	7*128=896 bits	6*128=768 bits	8*128=1024 bits	9*128=1152 bits
Computational Complexity Cost					
Registration Phase (User Side)	$2t_h+2t_{\oplus}$	$1t_h+2\ t_{\oplus}$	$2t_h+2t_{\oplus}$	$1t_h+2t_{\oplus}$	$1t_h$
Registration Phase (Server Side)	$6t_h+4t_{\oplus}$	$4t_h+3\ t_{\oplus}$	$4t_h+\ 5t_{\oplus}$	$3t_h+3t_{\oplus}$	$3t_h+\ 1t_{\oplus}$
Login Phase	$12t_h+11t_{\oplus}$	$8t_h+10\ t_{\oplus}$	$8t_h+12t_{\oplus}$	$8t_h+9t_{\oplus}$	$4t_h+3t_{\oplus}$
Authentication Phase	$8t_h+4\ t_{\oplus}$	$6t_h+3\ t_{\oplus}$	$6t_h+9t_{\oplus}$	$6t_h+4t_{\oplus}$	$7t_h+4t_{\oplus}$
Password Change Phase	$9t_h+10\ t_{\oplus}$	$6t_h+7\ t_{\oplus}$	$10t_h+12t_{\oplus}$	$6t_h+7t_{\oplus}$	$7t_h+3t_{\oplus}$
Sum of Computational Cost	$37t_h+31t_{\oplus}$	$25t_h+25\ t_{\oplus}$	$30t_h+40t_{\oplus}$	$24t_h+25t_{\oplus}$	$22t_h+11t_{\oplus}$

The storage cost is defined as a number of parameters stored in the smart card. Parameters like $\{A_i, C_i, D_i, E_i, F_i, \text{hash}\}$ are saved in smart card's memory in the scheme. Hence, cost of memory space/storage is 6*128=768 bits. Figure 2 shows the comparison graph of storage space estimation cost (in bits) of presented scheme SPAS along with various other relevant schemes.

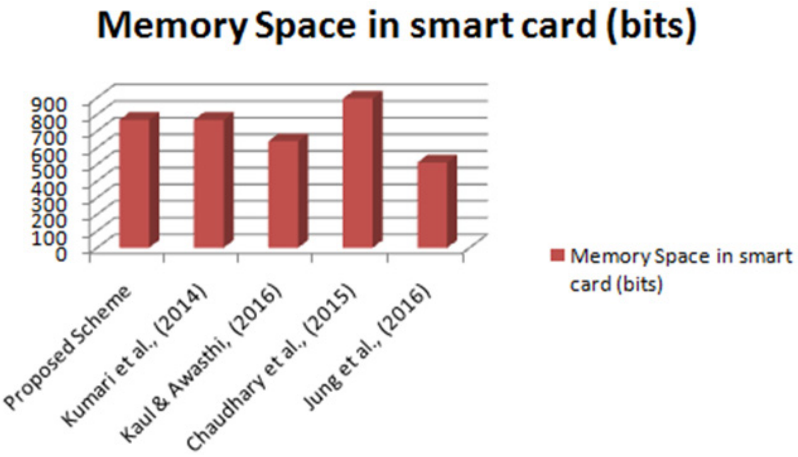

Figure 2. Storage cost comparison

- The total number of bits used in transmitting in login and authentication stage for whole messages are termed as transmission or communication cost.
- The scheme uses 6 parameters in login phase as $\{c_i d_i, \beta_i, J_i, K_i, t_1, I_i\}$ which requires 6*128=768 bits and for mutual authentication the number of parameters are $\{a, t_3\}$, requiring 2*128=256 bits. Hence, overhead for communication becomes = 6*128+2*128 = 1024 bits. Figure 3 shows the comparison graph of communication cost estimation (in bits) of our scheme along with various other relevant schemes.
- At user end, SPAS uses two hash functions along with two XOR function during registration phase. Therefore $2t_{h\,(.)}+ 2t_{\oplus}$ is the computational complexity. Correspondingly, the server uses six hash function along with four XOR function during the registration phase, therefore, computational complexity is $6t_{h(.)}+4t_{\oplus}$ at server end.
- During login stage, the scheme needs twelve hash functions along with eleven XOR functions to produce login request. Hence, $12t_{h\,(.)}+11t_{\oplus}$is the computational cost for login message.

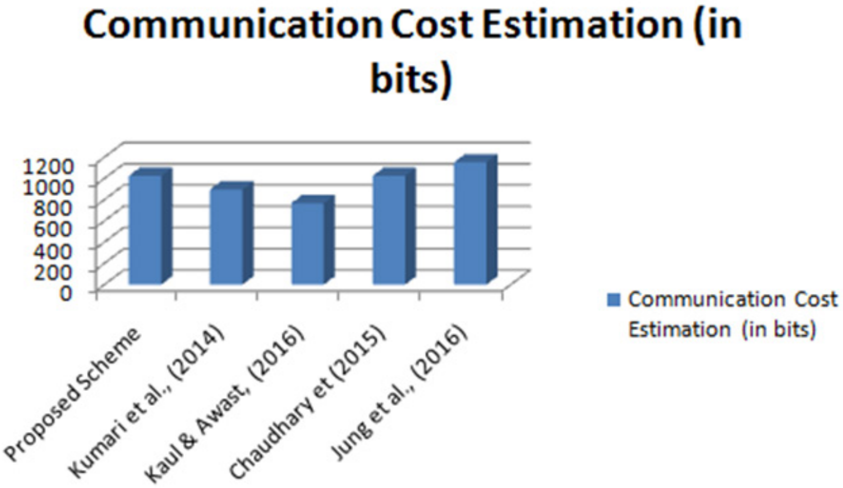

Figure 3. Communication Cost Comparison

- During mutual verification phase, a user needs one hash function, as well as server, needs seven hash along with four XOR functions. Hence, total computational cost for verification/authentication requires $8h_{(.)}+ 4t_{\oplus}$ operations.
- Figure 4 shows the comparison graph of computational complexity estimation cost in (bits) of scheme versus other relevant schemes. In the suggested scheme, computational complexity cost in (bits) is slightly higher than other's scheme (Kumari et al., 2014; Chaudhary et al., 2015; Jung et al., 2016) which is not

baseless as this increased computational cost prevents it from different attacks possible in the network.

- After analysis, we found that schemes as Kaul and Awasthi, (2016); Chaudhary et al. (2015) and Jung et al.(2016) also suffer various attacks like server pose violation, conspirator violation, user impersonation attack, off-line password attack.
- Since the scheme uses low memory (bits) and having low computational cost (bits), and communication cost (bits) as well as secure against various attacks, therefore, we concluded that suggested idea/scheme performed most excellent from others so that we can implement it more practically than others over insecure networks.
- We demonstrated the security analysis of various protocols in Table 3 and achievements/goals in Table 4. Here, Figure 5 shows graph of security characteristics comparison and Figure 6 presents the graph of goal/achievements comparison.

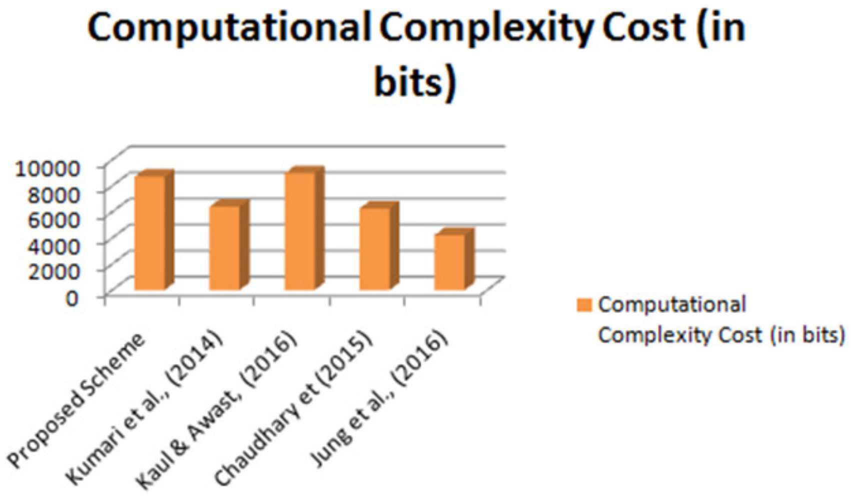

Figure 4. Computational Cost Comparison

Table 3

Security characteristics of our suggested scheme along with various other relevant schemes

S. No.	Security Characteristics	Proposed Scheme	Kumari et al. (2014)	Kaul and Awasthi, (2016)	Chaudhary et al. (2015)	Jung et al. (2016)
1	Conspirator Attack	No	Yes	No	No	No
2	Chip Card Loss attack	No	Yes	No	No	Yes

Table 3 *(Continue)*

S. No.	Security Characteristics	Proposed Scheme	Kumari et al. (2014)	Kaul and Awasthi (2016)	Chaudhary et al. (2015)	Jung et al. (2016)
3	User mask violation/attack	No	No	No	Yes	No
4	Server mask violation/attack	No	No	No	Yes	Yes
5	On-line password assumption violation	No	No	No	No	Yes
6	Off-line password assumption violation	No	Yes	No	No	Yes
7	Replay violation	No	Yes	No	No	No
8	Denial of service violation	No	No	No	No	Yes
9	Man in middle attack	No	Yes	No	Yes	Yes
10	Lifted verifier violation	No	Yes	Yes	No	No

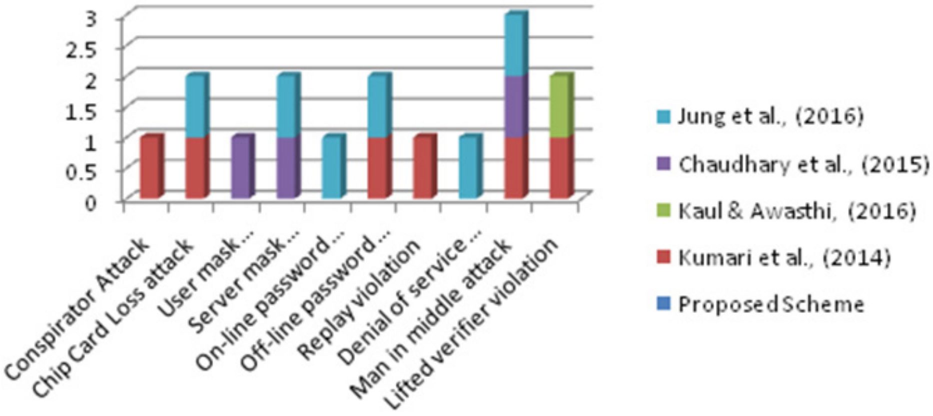

Figure 5. Security Characteristics Comparison Graph

Table 4

Goal/Achievements Comparison

S. No.	Protocols	Proposed Scheme	Kumari et al. (2014)	Kaul and Awasthi (2016)	Chaudhary et al. (2015)	Jung et al. (2016)
1	User's anonymity and un-traceability	Yes	No	No	Yes	No
2	Support forward secrecy	Yes	No	No	Yes	Yes
3	Maintain Mutual Verification	Yes	No	No	Yes	Yes
4	Compromise secret key	No	Yes	No	No	No
5	Compromise session key	No	Yes	No	No	No
6	Single registration	Yes	Yes	Yes	Yes	Yes
7	Freely change password	Yes	Yes	Yes	Yes	Yes
8	No need of verification table	Yes	Yes	Yes	Yes	Yes
9	Provide fast incorrect password checking facility	Yes	Yes	Yes	Yes	No
10	Scope of verification characteristics in chip card	Yes	Yes	No	No	No
11	Storage, functional as well as transmission cost must be low	Yes	Yes	No	Yes	No

We implemented the scheme in Python and the security features are tested/validate on AVISPA (Automated Validation of Internet Security Protocols and Applications*)* tool. AVISPA is having four tools to check the validation of security protocols.

- On-the-fly Model-Checker (OFMC)
- Constraint-Logic-based Attack Searcher (CL-AtSe)
- SAT-based Model-Checker (SATMC)
- Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP)

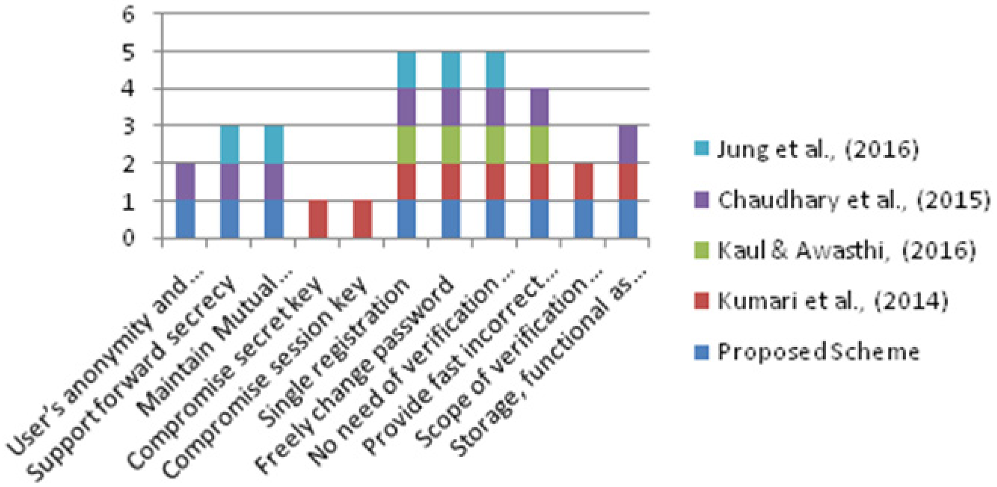

Figure 6. Goal/Achievements Comparison Graph

Figure 7 shows process flow diagram of AVISPA tool. The security analysis of the presented scheme has confirmed its performance in terms of reliability and safety because all the observations, trials, measuring operations return the same outcome on repeated testing. Moreover, the scheme is free from outside attacks, hazard, and insecurity, threat arising from loss of smart card. Analysis of the scheme shows that it is robust against all known attacks and ensures anonymity and privacy as there is no adverse effect of any types of attacks and it can withstands all such types of rigorous conditions. Moreover, its expandability characteristic is having the ability to support extra network users.

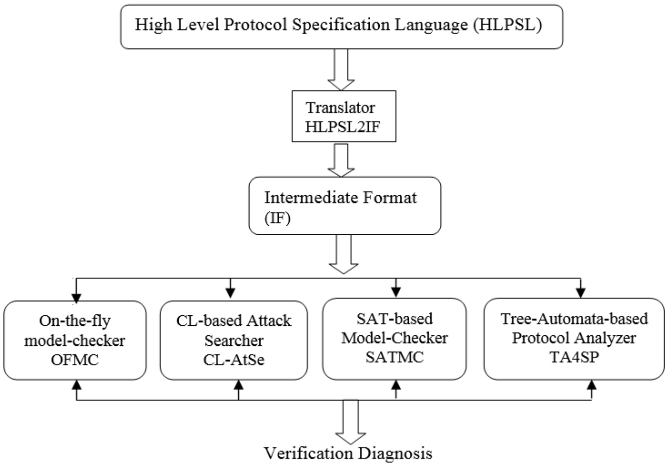
High Level Protocol Specification Language (HLPSL)
Translator HLPSL2IF
Intermediate Format (IF)
On-the-fly model-checker OFMC
CL-based Attack Searcher CL-AtSe
SAT-based Model-Checker SATMC
Tree-Automata-based Protocol Analyzer TA4SP
Verification Diagnosis

Figure 7. AVISPA TOOL

CONCLUSION

The SPAS scheme is introduced in this paper. The scheme ensures security, privacy and confidentiality of a user. It is an improvement over all the existing schemes (Chaudhary et al., 2015; Das et al., 2004; Jung et al., 2016; Kaul & Awasthi, 2016; Kumari et al., 2014). During investigation we found that earlier works were not secure enough for practical applications because all security parameters can be easily obtained by the challenger and are vulnerable to chip card loss violation as well as user un-traceability violation attack. Moreover, an adversary can get server's secret key, as well as password of the entire registered user's and also the session key for server, which may lead to destroying the whole system. During performance analysis of the scheme it is found that it incurs some extra bits of memory, and increased computational and communication cost but it helps to prevent smart card loss and user anonymity violation attack. The analysis of the scheme has confirmed its feasibility and performance in practical approach. The proposed scheme may be used 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 academics, banking, and business applications. After performance and efficiency comparison, we demonstrate that suggested idea is safer as well as relevant to practical approach.

REFERENCES

Chang, C. C., & Wu, T. C. (1991). Remote password authentication with smart cards. *IEEE Proceedings-E 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*, 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. In *Fifth International Joint Conference on INC, IMS and IDC* (pp. 2108-2110). Seoul, South Korea. doi: 10.1109/NCM.2009.101

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., Naqvi, H., Kumari, S., & Khan, M. K. (2015). An enhanced privacy preserving remote user authentication scheme with provable security. *Security and Communication Networks, 8*, 3782-3795. doi:10.1002/sec.1299

Chien, H. Y., Jan, J. K., & Tseng, Y. M. (2002). An efficient and practical solution to remote authentication: smart card. *Computers and Security, 21*(4), 372-375. doi:10.1016/S0167-4048(02)00415-7

Das, M. L., Saxena, A., & Gulati, V. P. (2004). A Dynamic ID-based Remote User Authentication Scheme. *IEEE Transactions on Consumer Electronics, 50*(2), 629-631. doi:10.1109/TCE.2004.1309441

ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory, 31*(4), 469-472. doi:0018-9448/85/0700-0469

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. doi:10.1109/30.826377

Jung, J., Lee, D., Kim, J., Lee, Y., Kang, D., & Won, D. (2016). Cryptanalysis and improvement of efficient password-based user authentication scheme using hash function. In *Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication* (pp. 23). New York, NY, USA. doi:10.1145/2857546.2857570

Karuppiah, M., & Saravanan, R. (2014). A secure remote user mutual authentication scheme using smart cards. *Journal of Information Security and Applications, 19*(4), 1-13. doi:10.1016/j.jisa.2014.09.006

Karuppiah, M., & Saravanan, R. (2015). Cryptanalysis and an improvement of new remote mutual authentication scheme using Smart Cards. *Journal of Discrete Mathematical Sciences and Cryptography, 18*(5), 623-649. doi:10.1080/09720529.2015.1013693

Kaul, S. D., & Awasthi, A. K. (2016). Security enhancement of an improved remote user authentication scheme with key agreement. *Wireless Personal Communications, 89*(2), 621-637. doi:10.1007/s11277-016-3297-6

Khan, M. K., Kumari, S., Wang, X. M., & Kumar, R. (2014, August 24-27). Security Issues of Chen et al.'s dynamic ID-based authentication scheme. In *12th International Conference on Dependable, Autonomic and Secure Computing (DASC)* (pp. 125-128). Dalian, China. doi:10.1109/DASC.2014.31

Kocher, P., Jaffe, J., & Jun, B. (1999). Differential Power Analysis. In M. Wiener (Eds.), *Advances in Cryptology-CRYPTO' 99* (pp. 388-397). Berlin, Heidelberg: Springer. doi:10.1007/3-540-48405-1_25

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. doi:10.1109/TCE.2004.1277863

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. doi:10.1007/s00607-013-0308-2

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, 27*(12), 3939-3955. doi:10.1002/dac.2590

Kumari, S., Khan, M. K., & Li, X. (2014). An improved remote user authentication scheme with key agreement. *Computers and Electrical Engineering, 40*(6), 1997-2012. doi:10.1016/j.compeleceng.2014.05.007

Lamport, L. (1981). Password authentication with insecure communication. *Communications of the ACM, 24*(11), 770-772. doi:0001-0782/81/1100-772

Li, X., Niu, J., 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

Lu, Y., Li, L., Peng, H., & Yang, Y. (2015). A secure and efficient mutual authentication scheme for session initiation protocol. *Peer-to-Peer Networking and Applications, 9*(2), 449-459. doi: 10.1007/s12083-015-0363-x

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*, 1235-1248.

Messerges, T. S., Dabbish, E. A., & Sloan, R. H. (2002). Examining smart-card security under the threat of power analysis attacks. *IEEE Transaction on Computers, 51*(5), 541-552. doi: 10.1109/TC.2002.1004593

Song, R. (2010). Advanced smart card based password authentication protocol. *Computer Standards & Interfaces, 32*, 321–325. doi:10.1016/j.csi.2010.03.008

Tang, Y. L., Hwang, M. S., & Lee, C. C. (2002). A simple remote user authentication scheme. *Mathematical and Computer Modelling, 36*, 103-107. doi:SO895-7177(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 Networking and Applications, 8*(5), 903-910. doi:10.1007/s12083-014-0248-4

Wang, D., He, D., Wang, P., & Chu, C. H. (2014). Anonymous two-factor authentication in distributed systems: Certain goals are beyond attainment. *IEEE Transactions on Dependable and Secure Computing, 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

Wen, F., & Li, X. (2011). An improved dynamic id-based remote user authentication with key agreement scheme. *Computers and Electrical Engineering, 38*, 381-387. doi:10.1016/j.compeleceng.2011.11.010

Yang, W. H., & Shieh, S. P. (1999). Password authentication schemes with smart cards. *Computers and Security, 18*(8), 727–733. doi:0167-4048/99