Security Issues in Smart Card Based Password Authentication Scheme

Ajay Kumar Sahu¹, Dr. Ashish Kumar²

¹Asst. Professor, Dept. of CSE, RKGIT, Ghaziabad, Uttar Pradesh, India, ajay4989@gmail.com
²Professor, Dept. of CSE, ITS, Greater Noida, Uttar Pradesh, India, ashishcse29@gmail.com

ABSTRACT

To secure information from unauthorized access, various authentication schemes have been deployed. Among these, password based authentication schemes using smart card are widely used for various applications such as remote user login, online banking, ID verification, access control and e-commerce. It provides mutual authentication between user and server. However, previous schemes are vulnerable to various attacks and are neither efficient, nor user friendly. Today, there are many potential attacks that are targeted at authentication including masquerade attack, insider attack, parallel session attack, offline password guessing attack, server spoofing attack, and many more. This article portrays security requirements and possible attacks for smart card based password authentication scheme. In addition, various solutions for different attacks are also discussed.

KEYWORDS:— Authentication, password, security, smart card, Cryptography.

1. INTRODUCTION

With the rapid growth of computer networks, more and more users access the remote server's service in a distributed computing environment. Due to the fast development of Internet and wireless communications, many activities like online-shopping, online banking, online voting are conducted over it. Electronic transactions carried out over the network platform are gaining popularity and it is widely accepted in the Internet computing world. A number of organizations doing business in the traditional way are extending themselves to do business over the Internet. To protect data during their transmission over insecure channel, adequate network security measures are needed to resist potential attacks from eavesdropping, unauthorized retrieval and intended modification, etc. For every business transaction authentication is required. It ensures the origin of a message or electronic document correctly identified with an assurance that the identity is not a fake. It is the primary requirement before the user accesses the server over insecure channel as it prevents unauthorized access.

Various authentication schemes have been proposed to secure the information or resources from unauthorized user [1], [2]. One among them is password authentication scheme. In conventional password authentication schemes, server maintains password table or verification table which contains user identifier (ID) and password (PW) for all the registered users. It is used to authenticate the legitimate user. Every user has an ID and PW. Whenever a user wants to access resources from a server, he or she submits ID and PW to pass the authentication phase. The server verifies the PW corresponding to the ID from verification table. If the submitted password matches the one stored in the verification table then server authenticates the user. However, there is a threat in such a process; an intruder can impersonate a legal user by intercepting the messages from the network and login to the server later using the intercepted information. Even if the PW is encrypted during communication, such an impersonation attack is still possible. In addition, if an intruder breaks into the server; the contents of the verification table can be easily modified or stolen. Major downside of this scheme is securing the verification table which stores password in plain text form.

One of the solutions to cope up with this problem is to encode the password using hash function and store the resultant test pattern in a verification table [3]. Another alternate solution is to store the password in encrypted form which cannot be easily derived from an attacker even if attacker knows the content of the verification table. However, it consumes more memory space to store the encrypted password. In both the approaches, size of the verification table is directly proportional to the number of users. In other words, size of the
verification table increases as the number of users increases. Management of such a huge verification table increases burden to the server. In addition, they are not secure since an attacker can modify the contents of the verification table which result the entire system to break down. To resist all possible attacks on the verification table, smart card based password authentication scheme has been proposed. In this scheme, server does not maintain a verification table to authenticate the legitimate user.

2. POSSIBLE ATTACKS ON SMART CARD AUTHENTICATION SCHEME

The remote user authentication scheme is used to authenticate the legitimacy of the remote users over an insecure channel. Since last decade many elegant remote user authentication schemes using smart card have been proposed [4, 6], [8-10], [12-17], [21], [22]. Smart card authentication scheme is usually composed of three phases namely; registration phase, login phase and authentication phase.

The registration phase is invoked only once when a new user registers in the server. Upon receiving registration request over secure channel, server issues a smart card to user by storing the computed parameters into smart card memory. The login phase and authentication phase are invoked when a user wants to login the server. Upon receiving the login request, server checks the validity of the login request to authenticate the user. The notations used throughout in paper are defined in Table I.

TABLE I: NOTATIONS USED IN THIS PAPER

<table>
<thead>
<tr>
<th>Notations</th>
<th>Their Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$U_i$</td>
<td>Legitimate user</td>
</tr>
<tr>
<td>$ID$</td>
<td>Identifier of $U_i$</td>
</tr>
<tr>
<td>$PW$</td>
<td>Password of $U_i$</td>
</tr>
<tr>
<td>$X_r$</td>
<td>Secret key of the server</td>
</tr>
<tr>
<td>$U_a$</td>
<td>Adversary</td>
</tr>
<tr>
<td>$ID_a$</td>
<td>Identifier of $U_a$</td>
</tr>
<tr>
<td>$PW_a$</td>
<td>Password of $U_a$</td>
</tr>
<tr>
<td>$PW'$</td>
<td>Password guessed by $U_a$</td>
</tr>
<tr>
<td>$PW''$</td>
<td>New password entered during password change phase</td>
</tr>
<tr>
<td>$T_1$</td>
<td>Timestamp at which login request is created</td>
</tr>
<tr>
<td>$T_2$</td>
<td>Timestamp at which server receives the login request</td>
</tr>
<tr>
<td>$T_3$</td>
<td>Timestamp at which server sends a message back to $U_i$</td>
</tr>
<tr>
<td>$T_d$</td>
<td>Timestamp at which $U_i$ receives a message from server</td>
</tr>
<tr>
<td>$T'$</td>
<td>Current timestamp of $U_i$'s system</td>
</tr>
<tr>
<td>$\Delta T$</td>
<td>Predetermined time interval of Transmission delay</td>
</tr>
<tr>
<td>$r, b$</td>
<td>Random numbers</td>
</tr>
<tr>
<td>$a$</td>
<td>Primitive element</td>
</tr>
<tr>
<td>$p, q$</td>
<td>Prime numbers</td>
</tr>
<tr>
<td>$S_{ID}$</td>
<td>Shadowed identity</td>
</tr>
<tr>
<td>$h(.)$</td>
<td>Secure one way hash function</td>
</tr>
<tr>
<td>$\odot$</td>
<td>Bitwise Exclusive-OR operation</td>
</tr>
<tr>
<td>$E_{x()}$</td>
<td>Symmetric encryption using key ‘$x$’</td>
</tr>
<tr>
<td>$D_{x()}$</td>
<td>Symmetric decryption using key ‘$x$’</td>
</tr>
<tr>
<td>$C_{f(.)}$</td>
<td>Function to generate check digit</td>
</tr>
<tr>
<td>$Re\ d(.)$</td>
<td>One-way shadow function maintained at the server</td>
</tr>
</tbody>
</table>

A remote user authentication scheme based on ElGamal’s cryptosystem has been proposed [4]. It consists of registration phase, login phase and authentication phase.

In the registration phase, $U_i$ submit $ID$ to the server. Upon receiving the registration request, server computes $PW = ID^{\frac{1}{r}}$ mod $p$ and issues a smart card to $U_i$ by storing $(h(.), p)$ into smart card memory. The $PW$ is delivered to $U_i$ through a secure channel.

In the login phase, $U_i$ insert the smart card to the card reader and keys in $ID$ and $PW$. After entering the credentials, the card reader generates a random number ‘$r’ , computes $C_i = ID' \mod p$ and $t = h(T_1 \odot PW) \mod(p-1)$, $M = ID' \mod p$, $C2 = (M \times PW') \mod p$ and sends the login request $(ID, C_1, C_2, T_1)$ to the server.

In the authentication phase, upon receiving the login request, server first checks the validity of $ID$ to accept/reject the login request. If it is incorrect, the request is rejected else it is considered for next step of check. The validity of time interval between $T_2$ and $T_1$ is computed i.e., $T_2 - T_1 \geq \Delta T$. If true, server rejects the login request, else, test out whether $C_i \times C_{x}^{\frac{1}{r}} = ID^{x} \mod p$ holds or not in order to authenticate $U_i$.

It is claimed that the scheme does not maintain any password or verification table and secure against replay attack. However, it is vulnerable to impersonation attack [5].
A. Impersonation Attack

It is defined as the attack in which $U_a$ attempts to modify the intercepted messages to masquerade the legal $U_i$ and login to the server. $U_a$ submits $ID_a$ to the server to masquerade as legal $U_i$ [5]. The server issues $PW_a$ and smart card to $U_a$. Upon receiving, $U_a$ computes valid pair of $(ID_i, PW_i)$ of a legitimate user ‘$U_i’ without knowing server secret key, i.e. ‘$X_i’.

$ID_i = (ID_a \times ID_i) \mod p$

$PW_i = (ID_i \times X_i) \mod p$

$PW_i = (PW_a \times PW_i) \mod p$

Further improvement has been suggested [6] which is also crypt analyzed [7]. To withstand impersonation attack, modified scheme has also been proposed in which login request contents are ($S_{ID}$, $C_1$, $C_2$, $T_i$) [9], where $C_{ID} = C_k (S_{ID})$.

A remote user authentication scheme using one-way hash function has been proposed [10]. Its major drawbacks are i) password is issued by the server which results $U_i$, not to choose and change the password freely. ii) no mutual authentication. In addition, it is pointed out that the scheme is vulnerable to offline and online password guessing attacks [11].

B. Offline Password Guessing Attack

In offline password guessing attack, $U_a$ attempts to determine whether each of the guessed passwords is correct or not from the intercepted messages transmitted between $U_i$ and the server. $U_a$ can easily verify whether guessed password is correct or not if $U_i$’s password is weak. As per the scheme [11], $U_a$ intercepts the login request $(ID, T_i, C_1)$ where $C_1 = h(T_i \oplus PW)$, randomly guesses $PW$ and checks whether $C_1' = h(T_i \oplus PW')$ holds or not. If equal, the guessed $PW$ is a valid one. Using $PW'$, $U_a$ is able to impersonate the legitimate $U_i$ to login the server. The other possible attack is online password guessing attack.

C. Online Password Guessing Attack

In this, $U_a$ tries to use guessed passwords iteratively to pass the authentication phase online. As defined in [11], $U_a$ randomly guesses $PW$ and creates a forged login request $(ID, T_i', C_1')$ where $C_1' = h(T_i' \oplus PW)$ . If the login request is accepted by the server, $U_a$ successfully impersonates a valid $U_i$ to login the server and the guessed password $PW$ is $U_i$’s password. This attack can be restricted by limiting the number of attempts. To overcome these limitations, a remote user authentication scheme using one-way hash function has been proposed [12]. It is claimed that the scheme does not require any password or verification table in the server and legal user can choose password without the help of server. Moreover, it provides mutual authentication between user and server. However, it is pointed out that scheme is vulnerable to parallel session attack [11].

D. Parallel Session Attack

It is defined as the attack in which $U_a$ eavesdrops the messages transmitted between $U_i$ and server and sends it back as a valid login request to the server. As mentioned in [11], $U_i$ computes $C_2 = h(h(ID \oplus X_i) \oplus T_i)$ to create the login request $(ID, C_2, T_i)$ which is transmitted to server. In order to provide mutual authentication, server sends the response message $(T_3, C_3)$ to $U_i$ in which $C_3 = h(C_1' \oplus T_i) = h(h(ID \oplus X_i) \oplus T_i)$. $U_a$ intercepts the messages exchanged between $U_i$ and server to masquerade as legal $U_i$, to start a new session. $U_a$ sends $(ID, C_2, T_i)$ back to server where $T_3 = T_3$ and $C_2 = C_3$. This message passes the authentication phase since $C_2 = C_3 = h(C_1' \oplus T_i) = h(h(ID \oplus X_i) \oplus T_i)$ which results $U_a$ login into the server. Modified scheme has been proposed to overcome the parallel session attack [13]. However, still it is vulnerable to impersonation attack and reflection attack; does not provide user anonymity [14].

E. Reflection Attack

In reflection attack, $U_a$ eavesdrops login request during transmission between $U_i$ and server and sends it (or a modified version of the message) back to $U_i$ to masquerade as the legitimate server. As stated in [14], $U_a$ intercepts the login request $(ID, C_2, T_i)$ transmitted between $U_i$ and the server, where $C_2 = h(h(ID \oplus X_i) \oplus T_i)$ and sends back $(T_3, C_3)$ to $U_i$ immediately to impersonate the server where $T_3 = T_3$ and $C_3 = h(C_3) = h(h(ID \oplus X_i) \oplus T_3)$.

Upon receiving the message $(T_3, C_3)$, $U_i$ computes $h(h(C_1 \oplus T_i)) = h(h(ID \oplus X_i) \oplus T_i)$.
Since the computed result equals $C_3$, $U_i$ believes that the message is sent by the server. The reflection and parallel session attacks are possible due to the transmission of messages with similar structure. To overcome these weaknesses, further improvement has also been proposed [14]. However, the scheme does not solve the time synchronization problem and server maintains an extra database for each user. To withstand insider attack and reflection attack, an improvement over the scheme [12] has been suggested [15].

F. Insider Attack

During the registration phase, $U_i$'s password is revealed to server over a secure channel. An insider of server obtains $U_i$'s password to impersonate the legal $U_i$ and access other servers if same password is used to access other servers. In addition, the scheme [15] allows users to change their password freely through password change phase. The password change phase is invoked whenever $U_i$ wants to change old password $PW^\prime$ . $U_i$ inserts the smart card to the reader, keys $ID$ and $PW$ and requests to change the password by entering new $PW^\prime\prime$. Smart card computes $R = R \oplus h(b \oplus PW) \oplus h(b \oplus PW^\prime)$ which yields $h(EID \oplus X_i) \oplus h(b \oplus PW^\prime)$ where $EID = (ID \parallel n)$, $R = h(EID \oplus X_i) \oplus h(b \oplus PW)$, $n$ denotes the number of times $U_i$ re-registers to server. Smart card replaces $R$ with $R'$.

Since the password change operation is performed at the card reader side, there is no need to interact with the server. It is proved that the scheme is weak against parallel session attack and has insecure password change phase. Further improvement has also been suggested [16]. Nevertheless, the scheme is vulnerable to guessing attack, Denial-of-Service attack and forgery attack [17]. To overcome these drawbacks, an improved scheme has been proposed which is also cryptanalyzed [18]. Further, it is found that the scheme is insecure against guessing attack, denning-sacco attack.

G. Attack on Perfect Forward Secrecy

Perfect forward secrecy assures that even if one long term key is compromised, it does not reveal any session keys used before. Session key is $C_1 = h(r \oplus b)$ as per [18]. If $U_a$ gets server secret key $X_i$, and eavesdrops the login request $(ID, C_1, C_2, T_1)$, where

$$C_i = P \oplus h(r \oplus b)$$

$$C_1 = h(ID \oplus X_i) \oplus h(r \oplus b)$$

$$C_2 = h_p(h(r \oplus b) \oplus T_1)$$

$U_a$ easily computes $P = h(ID \oplus X_i)$ and $C_1 \oplus P = C_1' = h(r \oplus b)$ which results to obtain the session key. Another possible attack is denning sacco attack.

H. Denning Sacco Attack

It is an action where $U_a$ obtains a session key from an eavesdropped session and uses the same key either to impersonate the legitimate $U_i$ or mount a dictionary attack on $U_i$'s password. As per [18], $U_a$ gets the session key $C_1' = h(r \oplus b)$ and obtains $U_i$'s secret information as follows

$$C_1 \oplus C_1' = P \oplus h(r \oplus b) \oplus h(r \oplus b)$$

$$C_1 \oplus C_1' = P$$

Upon obtaining $U_i$'s secret value $h(ID \oplus X_i)$, $U_a$ can impersonate the legitimate $U_i$. Consider the case, $U_a$ computes $C_1'' = P \oplus h(r \oplus b')$, $C_2'' = h_p(h(r \oplus b') \oplus T)$ where $r'$, $b'$ are two random numbers chosen by $U_a$ and $T$ denotes current timestamp of $U_a$'s system. $U_a$ prepares a forged login request $M = (ID, C_1'', C_2'', T')$ to impersonate $U_i$. Another possible attack is smart card stolen attack.

I. Smart Card Stolen Attack

When a smart card is lost or stolen, unauthorized user who obtains the smart card can guess the password of $U_i$ using password guessing attacks or impersonate a valid $U_i$ to login into the server. $U_a$ who steals the smart card can obtain the secret information stored in the stolen smart card by monitoring the consumption of power [19] or by analyzing the revealed information [20]. $U_a$, who steals a smart card, knows $R$, $V$, $b$, $h$ and $ID$ [18]. Using these parameters, $U_a$ derives $PW$ from $V = h_p(h(b \oplus PW))$ without any knowledge of server secret key $X_i$ by checking $V = h(R \oplus (b \oplus PW))$ or not where $PW^\prime$ is the guessed password.
\[ V = h(\text{RID} \oplus \text{PW}), h(\text{b} \oplus \text{PW'}) \]
\[ P = R \oplus h(\text{b} \oplus \text{PW}) \]
\[ P = h(\text{ID} \oplus Xs) \]

If holds, \( PW' \) is the correct password of \( U_j \). In this approach, \( U_a \) derives \( h(\text{ID} \oplus X_j) \) and \( h(\text{b} \oplus PW) \) to impersonate \( U_j \). All the schemes discussed so far do not solve the time synchronization problem. None-based scheme [21] overcomes the time synchronization problem. It inherits all the previous advantages with additional feature such as session key generation agreed between \( U_i \) and server.

However, security flaws in this scheme are:

i) it is susceptible to insider attack.

ii) \( U \) is not allowed to change the password freely.

iii) it uses symmetric encryption and decryption which is inefficient for low computational powered smart cards.

Another nonce-based scheme has been proposed [22] to solve the serious time synchronization problem. It takes over all the previous advantages with an additional characteristic that it uses one way hash function to reduce the computational cost. Nevertheless, it is vulnerable to impersonation attack, offline password guessing attack, Denial-of-Service attack and man-in-the-middle attack [23].

J. Man-in-the-Middle Attack

The man-in-the-middle attack is a form of active eavesdropping in which \( U_a \) sits between the server and \( U_i \) by making independent connections between them. The messages which are exchanged between \( U_i \) and server are intercepted by \( U_a \) without the knowledge of \( U_i \) and server. As per the scheme [23], assume that \( S_i = (a)^{N_s} (\mod q) \) and \( W_i = a^{N_s} (\mod q) \) sent by server and \( U_i \) respectively. During the transmission, \( U_a \) intercepts both the messages and sends the modified/new message to both of them. \( U_a \) has two session keys, one is used between \( U_i \) and server and the other is used between server and \( U_a \), i.e., \( K_{sc} = (S_i)^{N_s} (\mod q) = (a^{N_s})^{N_s} (\mod q) \) and \( K_{su} = (W_i)^{N_s} (\mod q) = (a^{N_s})^{N_s} (\mod q) \). Hence, \( U_a \) intercepts the communication exchanged between \( U_i \) and the server. In addition, \( U_a \) can replay modified/new message to both of them. Another scheme which resists impersonation attack, smart card loss attack and insider attack has been proposed [24]. In this scheme, timestamp \( T \) is used in the login phase to resist replay attack. However, it fails to withstand this attack.

K. Replay Attack

The replay attack is one in which an attacker re-submits the intercepted login request to impersonate the genuine user. The login request contents are (ID, CID, r, K, y, T1) as per [24]. During the transmission of login request exchanged between \( U_i \) and server, \( U_a \) captures it and resends the request by changing the value of \( T_1 \). Upon receiving the request, server finds the difference between \( T_2 \) and \( T_1 \) to check the freshness of received request i.e., \( T_2 - T_1 \leq \Delta T \). If it holds, \( U_a \) is successful in an attempt of impersonation Another possible attack is Denial-of-Service attack.

L. Denial-of-Service Attack

The Denial-of-Service attack prevents/inhibits the use of network resources and communication facilities. For example, \( U_a \) sends invalid login requests continuously to make the server busy. The schemes [12], [13], [15], [16], [21], [22], [24], [27] are vulnerable to this attack. \( U_a \) inserts its own smart card into the card reader and keys in invalid IDa and \( PW_a \). The card reader performs the computation to create a login request and sends it to the server. \( U_a \) continuously does the same thing to overload the server which results in restrain the server accessibility for the other valid users.

An ID based scheme using RSA cryptosystem has been proposed [25]. However, it exhibits impersonation attack and further improvement has also been suggested which inherits all the merits of the previous scheme with an added feature of mutual authentication. A user friendly authentication scheme based on one way hash function has been proposed. However, it is pointed out that the scheme is vulnerable to impersonation attack. To overcome this attack, further improvement has also been proposed. Moreover, it is proved that the scheme is insecure against guessing attack and forgery attack. To overcome these security weaknesses, an improved scheme has also been proposed. Though, it is analyzed that scheme is vulnerable to reflection attack and parallel session attack. This is due to the symmetric structure of communication messages exchanged between user and server.

A dynamic ID-based remote user authentication scheme using one way hash function has been
proposed and claimed that the scheme allows the users to choose and change their passwords freely, secure against ID-theft and resists reply attack, forgery attack, guessing attack, insider attack and stolen verifier attack. However, it is proved that the scheme is insecure against guessing attack and does not provide mutual authentication. To defeat these security flaws, a new scheme has been suggested. Nevertheless, it is susceptible to impersonation attack. Further, improved scheme has also been proposed. Its major drawbacks are i) it does not provide secure password change phase, b) user needs to remember the secret number Yi. On the other hand, it is pointed out that the scheme does not provide mutual authentication and is password independent. To overcome these limitations, an enhanced scheme has been proposed. It is shown that the scheme is exposed to insider attack, does not provide users to choose the password, session key establishment and does not preserve user anonymity. An improved scheme has also been proposed to overcome these flaws. Major drawbacks in the improved scheme are i) server maintains an extra table for the value of N corresponding to every user, where N denotes the number of times user registers at the server, b) wrong password detection is slow. Smart card authentication scheme based on one way hash function and symmetric key cryptography has been proposed. It is claimed that the scheme resists impersonation attack, parallel session attack, replay attack and modification attack. In addition, it provides mutual authentication and generates shared session key. However, it is pointed out that the scheme fails to resist Denial-of-Service attack and provide perfect forward secrecy.

3. SECURITY REQUIREMENT AND GOALS

In general, an ideal smart card-based, password authentication scheme should satisfy some of the security requirements (SR) described in. Here, we list out and define the security attacks that an ideal password authentication scheme should withstand:

SR1: The ability to resist smart card loss attacks, SR2: The ability to resist offline password guessing attacks, SR3: The ability to resist denial of service attacks, SR4: The ability to resist forgery attacks or impersonation Attacks, SR5: The achievement of mutual authentication, SR6: The ability to resist replay attacks, SR7: The ability to resist parallel session attacks and reflection attacks, SR8: The ability to resist stolen-verifier attacks and modification attacks, SR9: The ability to resist insider attack, SR10: Perfect forward secrecy, SR11: The ability to resist server spoofing attacks, SR12: The achievement of session key agreement, SR13: The achievement of user anonymity, SR14: Resist online password guessing attack.

An ideal password authentication scheme should withstand all of the above attacks. Besides, it should achieve the following goals: G1: No verification table, G2: Freely chosen password by the user, G3: No password reveal, G4: Password dependent, G5: Mutual authentication, G6: Session key agreement, G7: Forward secrecy, G8: User anonymity, G9: Smart card revocation, G10: Efficiency for wrong password login. Furthermore, Ma et al. Suggested three principles that are important to design a secure remote user mutual authentication scheme. These principles include the following:

1. Public-key techniques are very important to withstand offline password guessing attack and to preserve user anonymity.
2. There is an unavoidable trade-off when fulfilling the goals of local password updates and resistance to smart card loss attack.
3. At least two exponentiation operations conducted on the server side are necessary for achieving forward secrecy.

To be called an ideal, a password authentication scheme should be able to withstand all of the above attacks and achieve all of the above goals. Unfortunately, none of the existing password authentication schemes can withstand all the above attacks and achieve all the goals. So, still there are opportunities to develop an ideal remote user password authentication scheme, which satisfies all security requirements and which meets all the goals.

4. CONCLUSION

Security and efficiency are the main factors for any authentication scheme from the user’s perspective. In view of the fact, several smart card based remote user authentication schemes have been proposed. Here, we have defined the security requirements and goals an ideal password authentication scheme must satisfy and achieve. This paper describes various possible attacks and primary requirements to consider any smart card based authentication scheme as secure and efficient for practical applications. These issues are helpful to design and develop a secure and efficient smart card authentication scheme.

REFERENCES


