Security Issues in Smart Card Authentication Scheme

Ravi Singh Pippal, Jaidhar C. D., and Shashikala Tapaswi

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. 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.

Index Terms—Authentication, password, security, smart card.

I. 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 break 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 the 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.

The rest of the paper is organized as follows. Section II describes various possible attacks on smart card authentication scheme with solutions for them. Section III explores the security requirements. Finally, section IV concludes the paper.

II. 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], [24-35]. 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 this

Manuscript received February 16, 2012; revised March 27, 2012.

R. S. Pippal, S. Tapaswi are with ABV-Indian Institute of Information Technology and Management, Gwalior-474010, India (e-mail: ravi@iiitm.ac.in, tapaswi@iiitm.ac.in).

Jaidhar C. D. is with Defence Institute of Advance Technology, Pune-411025, India (e-mail: jaidhardc@diat.ac.in).
article are defined in Table I.

<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>$X_s$</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>
</tbody>
</table>

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

In the login phase, $U_i$ inserts 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 $C_1 = ID^r \mod p$, $t = h(T_t \oplus PW) \mod (p - 1)$, $M = ID^r \mod p$, $C_2 = (M \times PW') \mod p$ and sends the login request $(ID, C_1, C_2, T_t)$ 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 $T_t$ and $T_i$ is computed i.e., checks $T_2 - T_i \geq \Delta T$. If true, server rejects the login request, else, test out whether $C_2 \times (C_1^{X_s})^{-1} = ID^{h(T_t \oplus PW')} \mod p$ 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 $PW$ and smart card to $U_a$. Upon receiving, $U_a$ computes valid pair of $(ID_f, PW_f)$ of a legitimate user ‘$U_f$’ without knowing server secret key, i.e. ‘$X_s$’.

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

$$PW_f = (ID_f)^{X_s} \mod p$$

Further improvement has been suggested [6] which is also cryptanalyzed [7]. To withstand impersonation attack, modified scheme has also been proposed in which login request parameters are computed from $S_{ID}$ instead of $ID$ [8]. Additional improvement has been given in which $S_{ID}$ is computed instead of $ID$, its login request contents are $(S_{ID} \parallel C_{ID}, C_1, C_2, T_t)$ [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 $(ID, T_t, C_1)$, where $C_1 = h(T_t \oplus PW)$, randomly guesses $PW'$ and checks whether $C_1 = h(T_t \oplus PW')$ 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 $(ID, T', C'_1)$ where $C'_1 = h(T' \oplus PW')$. 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 \( C_2 = h(h(ID \oplus X_s) \oplus T_i) \) 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 \((T_i, C_3)\) to \( U_i \) in which
\( C_3 = h(C_2 \oplus T_i) = h(h(ID \oplus X_s) \oplus T_i) \). \( U_a \) 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 \( T_i = T_3 \) and \( C_2 = C_3 \).
This message passes the authentication phase since
\( C_2 = C_3 = h(C_2 \oplus T_3) = h(h(ID \oplus X_s) \oplus T_3) \) which
results \( U_a \) to 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 \( U_i \) as the legitimate
server. As stated in [14], \( U_i \) intercepts the login request
\((ID, C_2, T_i)\) transmitted between \( U_i \) and the
server, where \( C_2 = h(h(ID \oplus X_s) \oplus T_i) \) and sends back
\((T_3, C_3)\) to \( U_i \) immediately to impersonate the server where
\( T_3 = T_i \) and \( C_3 = h(C_2) = h(h(ID \oplus X_s) \oplus T_i) \). Upon
receiving the message \((T_3, C_3)\), \( U_i \) computes
\( h(h(C_2 \oplus T_i)) = h(h(ID \oplus X_s) \oplus T_i) \). 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 \( PW \) with a new password
\( PW' \). \( U_i \) inserts the smart card to the reader, keys \( ID \) and
\( PW \) and requests to change the password by entering new
\( PW' \). Smart card computes
\( R' = R \oplus h(b \oplus PW) \oplus h(b \oplus PW') \) which yields
\( h(EID \oplus X_s) \oplus h(b \oplus PW') \) where
\( EID = (ID \parallel n) \), \( R = h(EID \oplus X_s) \oplus h(b \oplus PW) \), \( n \)
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 and
has inadequacy to provide perfect forward secrecy.

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 \( C_i' = h(r \oplus b) \) as per [18]. If \( U_a \) gets
server secret key \( X_s \) and eavesdrops the login request
\((ID, C_1, C_2, T_i)\), where
\[
C_1 = P \oplus h(r \oplus b) \\
C_1 = h(ID \oplus X_s) \oplus h(r \oplus b) \\
C_2 = h(r \oplus b) \oplus T_i
\]
\( U_a \) easily computes \( P = h(ID \oplus X_s) \) and
\( C_1 \oplus P = C_i' = h(r \oplus b) \) 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
\( C_i' = h(r \oplus b) \) and obtains \( U_i \)'s secret information as follows
\[
C_i \oplus C_i' = P \oplus h(r \oplus b) \oplus h(r \oplus b) \\
C_i \oplus C_i' = P \\
C_i \oplus C_i' = h(ID \oplus X_s)
\]
Upon obtaining \( U_i \)'s secret value \( h(ID \oplus X_s) \), \( U_a \) can
impersonate the legitimate \( U_i \). Consider the case, \( U_a \)
computes \( C_i'' = P \oplus h(r' \oplus b') \), \( C_2'' = h_r(h(r' \oplus b') \oplus T') \)
where \( r' \), \( b' \) are two random numbers chosen by \( U_a \) and \( T' \)
I. Smart Card Stolen Attack

When a smart card is lost or stolen, an 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 $R$, $V$, $b$, $h(\cdot)$, $h_i(\cdot)$ and ID [18]. Using these parameters, $U_a$ derives $PW$ from $V = h_i(h(b \oplus PW))$ without any knowledge of server secret key $X_s$ by checking $V = h_{R^{\oplus b \oplus PW}}(h(b \oplus PW))$ or not where $PW'$ is the guessed password

$$V = h_{R^{\oplus b \oplus PW}}(h(b \oplus PW'))$$

$$P = R \oplus h(b \oplus PW')$$

$$P = h(ID \oplus X_s)$$

If holds, $PW'$ is the correct password of $U_i$. In this approach, $U_a$ derives $h(ID \oplus X_s)$ and $h(b \oplus PW)$ to impersonate $U_i$.

All the schemes discussed so far do not solve the time synchronization problem. Nonece-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) $U_i$ 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 nonece-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 $S_i = (\alpha)^{N_s} (\mod q)$ and $W_i = (\alpha)^{N_i} (\mod q)$ 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 $U_a$ i.e., $K_{sw} = (S_j)^{N_a} (\mod q) = (\alpha)^{N_a} (\mod q)$ and the other is used between server and $U_a$ i.e., $K_{sw} = (W_j)^{N_a} (\mod q) = (\alpha)^{N_a} (\mod q)$. 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 $T_i$ 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 $(ID, C_{id}, r, k, y, T_i)$ 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 $T_i$. Upon receiving the request, server finds the difference between $T_i$ and $T_a$ to check the freshness of received request i.e., $T_a - T_i \leq \Delta T$. 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], [30-35] are vulnerable to this attack. $U_a$ inserts its own smart card into the card reader and keys in invalid ID$_a$ 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 to 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 [26] 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 [27]. However, it is pointed out that the scheme is vulnerable to impersonation attack. To overcome this attack, further improvement has also been proposed [28]. 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 [29]. 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 [30] 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 [31]. 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 [32]. Its major drawbacks are i) it does not provide secure password change phase. b) user needs to remember the secret number Ys. On the other hand, it is pointed out that the scheme [30] does not provide mutual authentication and is password independent. To overcome these limitations, an enhanced scheme has been proposed [33]. 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 [34]. 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 [35]. 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 [36].

III. SECURITY REQUIREMENTS FOR SMART CARD AUTHENTICATION SCHEME

This section describes major security requirements for an ideal smart card authentication scheme. To consider any authentication scheme as secure and efficient, it has to encompass the following features:
1) Server need not to maintain password or verification table to authenticate the valid user.
2) User finds difficulty to remember the password if it is issued by the server. Hence, scheme is needed to allow the users to choose and change their password freely at any time without any interaction from the server.
3) Secure transmission of password is desirable at the time of registration. The password is not embedded in login request or in other words it is not revealed to the server in a plain text as a part of login request.
4) The computational workload of the smart card needs to be low. Even though the capabilities of smart cards are increasing, they are still limited. Intense computations need to be avoided both at the card reader side as well as server side.
5) Attacks such as Impersonation attack, Parallel session attack, reflection attack, Online Password Guessing attack, Offline Password Guessing attack, Reply attack, Stolen Verifier attack, attack on Perfect Forward Secrecy, Card Smart Loss attack, Denial-of-Service attack, Denning Sacco attack, Man-in-the-Middle attack must be withstand by the authentication scheme.
6) If an adversary tries to access the resources from the server by entering a wrong password, it has to detect quickly.
7) In timestamp-based authentication schemes, the clock of the server and all user systems are to be synchronized. In addition, transmission delay of the login request needs to be limited. However, it is inefficient from the practical point of view. To overcome clock synchronization problem and the limitation of transmission delay time, some of the schemes use nonce in place of timestamps. In order to check the freshness of the nonce, server needs to maintain an older nonce value for short span of time.
8) It is necessary that not only server verifies the legal users, but users also verify the legal server to achieve two way authentication. Every authentication scheme must support mutual authentication.
9) Session keys are used to secure the communication between user and server till the end of the session. Every smart card authentication scheme needs to support session key generation.
10) It is better that the user’s identity is anonymous. It is necessary that an efficient remote user authentication scheme preserves strong anonymity not only from the eavesdropper but also from the server.

IV. 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. 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.

ACKNOWLEDGMENT

The authors would like to thank ABV-Indian Institute of Information Technology and Management, Gwalior, India for providing the academic support.

REFERENCES


Shashikala Tapaswi has obtained her Ph.D. (Computer Engineering) from Indian Institute of Technology, Roorkee, India in 2002, M.Tech (Computer Science) from University of Delhi, India in 1993 and B.E. (Electronics Engineering) from Jiwaji University, Gwalior, India in 1986. She is a Professor in Information Technology Department, ABV-Indian Institute of Information Technology and Management, Gwalior, India. Her primary research areas of interest are Artificial Intelligence, Neural Networks, Fuzzy Logic, Digital Image Processing, Computer Networks, Mobile Networks, Adhoc Networks, Network Security etc.