A smart card based remote user authentication scheme

Mohammed Misbahuddin¹, P.Premchand², A.Govardhan³
¹Centre for Development of Advanced Computing
#68, Electronic City, Bangalore
India
mdmisbahuddin@yahoo.com

²Department of Computer Science and Engineering
Osmania University, Hyderabad
India
drpremchand_p@yahoo.com

³School of Information Technology
Jawaharlal Nehru Technological University, Hyderabad
India
govardhan_cse@yahoo.co.in

ABSTRACT: Password based authentication schemes are commonly used to authenticate remote users. Many schemes have been proposed both with and without smart cards but each have its own merits and demerits. This paper analyzes the security of an enhanced Dynamic ID based remote user authentication scheme and shows that the enhanced scheme has major security weaknesses. The paper also presents a new scheme which overcomes the weaknesses and provides better security strength.

Categories and Subject Descriptors
C 2.0 [General Communication]: Security and protection; D.4.6 [Security and Protection] Authentication; H.3 [Special-Purpose and Application-Based Systems] Smart Cards

General Terms
User authentication scheme, Password protection, Smart cards

Keywords
Smart Cards, User Passowords, User authentication

Received 1 Nov. 2007; Revised 12 Mar. 2008; Accepted 25 Mar. 2008

1. Introduction

Computer security has become an important issue as more and more systems in a networked environment have added control to the access process through authentication to avoid illegitimate user’s access. Password based authentication is one of the commonly used mechanisms to authenticate legitimate users. But since it only requires the user to enter his/her username and password (ie something you know), it does not achieve the level of assurance that the user is the one who he is claiming to be. So, researchers have proposed a more strong way of authentication called two factor authentication by combining something a user know (password) and something a user has (smart card or token etc). In two-factor authentication, whenever a user wants to access resources of a remote system, the user has to login by inserting the smart card into a card terminal and keying his secret information such as password to get authenticated [17].

In 1981, Lamport [9] proposed the first two-factor based remote user authentication scheme using smartcards. But this scheme maintains a password table to verify the validity of a user which is an insecure way of validating as the password table information could be read or altered by an intruder. Moreover, the server requires extra memory space to store the password table.

In order to overcome this, many schemes have been proposed [5][8] that replaced the password table with verification table (which stores the hash of password instead of plaintext password). However, these schemes are vulnerable to stolen verifier attack and incur an additional cost of maintaining verifier table on the server. To address this, many schemes have been proposed without verification table [4, 7, 10, 12, 14, 15] requiring no verifier table at the server.

In 2004, Das, Saxena and Gulati proposed a Dynamic ID based remote user authentication scheme[4] using smart cards that does not maintain verifier table and allows the user to choose and change their passwords freely without going to the server. They claimed that their scheme is secure against ID theft, replay attack, forgery attack, guessing attack, insider attack and stolen verifier attack. However, Awasthi and Lal [1],[2] have pointed out that Das et al. scheme is completely insecure and is like an open server access. Further, Wei-Chi Ku and Shen-Tein Chang [6] have shown that Das et al. scheme is insecure against impersonation attack where an unauthorized user can impersonate a valid user and successfully log onto the remote system.

In 2005, Liao et al. [7] have shown that Das et al scheme is vulnerable to guessing attack and proposed few modifications to Das et al. scheme and claimed that there scheme achieves mutual authentication. But the weaknesses described by Awasthi et al and Ku et al are still present in their scheme. Moreover, the mutual authentication in their scheme does not work successfully [16].

This paper proposes a scheme that not only overcomes the weaknesses of Liao et al scheme but also presents a secure mutual authentication. Moreover, besides inheriting the features of Das et al.[4] and Liao et al scheme[7], the proposed scheme uses random nonce to resist the replay attack instead of timestamps which leads to serious time concurrency problems.

The paper is organized as follows. Section 2 briefly reviews Liao et al. scheme. Section 3, analyzes the security of Liao et al
scheme. Section 4, presents the proposed scheme. Section 5, analyzes the security of the proposed scheme. Section 6, compares the functionality of both the schemes, Section 7 analyzes the efficiency of both the schemes and finally section 8 gives brief conclusion.

Review of LIAO et al Scheme
The scheme is composed of three phases, namely, registration phase, authentication phase and password change phase. The registration phase is performed only once, and the authentication phase is executed every time the user logs in to the remote system. The notations used throughout this paper are as follows:

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ui</td>
<td>The user i.</td>
</tr>
<tr>
<td>PWi</td>
<td>Password of Ui.</td>
</tr>
<tr>
<td>S</td>
<td>Server</td>
</tr>
<tr>
<td>h(.)</td>
<td>One-way hash function.</td>
</tr>
<tr>
<td>⊕</td>
<td>Bitwise XOR operation.</td>
</tr>
<tr>
<td>A =⇒ B: M</td>
<td>A sends M to B through a secure channel.</td>
</tr>
<tr>
<td>A → B: M</td>
<td>A sends M to B through insecure channel.</td>
</tr>
</tbody>
</table>

2.1 Registration Phase
This phase is invoked whenever a user Ui registers with the remote system. The user submits the identity IDi and chooses password PWi and submits IDi and h(PWi) to the remote system through secure channel.

Upon receiving the registration request, the remote system performs the following steps:

Step R1: S computes Ni = h(PWi) ⊕ h(x || IDi), where x is a secret key of the remote system and h(.) is the one way hash function.

Step R2: S personalizes the smart card with the parameters h(.) , Ni and y ; where y is the remote system’s secret number stored in each registered user’s smart card.

Step R3: S ⇒ Ui: Smart card.

2.2 Authentication Phase
Whenever a registered user wants to login to the server, he has to insert the smart card (eg: JavaCard) into card terminal attached to the system and must keep the card attached to it unless the session is terminated. This is because these cards are computationally capable of doing computations within it at client side[16].

2.2.1 Login Phase: The user Ui inserts his smart card into the card reader of a terminal, and keys his password PWi. Then, the smart card performs the following operations:

Step L1: Computes a dynamic ID called CIDi by performing CIDi = h(PWi) ⊕ h(Ni ⊕ y ⊕ T), where T is the current date and time of Ui’s system.

Step L2: Computes Bi = h(CIDi ⊕ h(PWi)); Ci = h(T ⊕ Ni ⊕ Bi ⊕ y).

Step L3: Ui → S: (CIDi, Ni, Ci, T).

2.2.2 Verification Phase:
Upon receiving the login message (CIDi, Ni, Ci, T) at time T', the server authenticates the user Ui with the following steps:

Step V1: Verifies the validity of the time interval between T and T'. If (T' - T) ≤ T, S accepts UI’s login request, otherwise rejects, where T denotes the valid time interval.

Step V2: Computes h(PWi) = CIDi ⊕ h(Ni ⊕ y ⊕ T); Bi' = h(CIDi ⊕ h(PWi)). Thereafter, checks whether Ci = h(T ⊕ Ni ⊕ Bi' ⊕ y). If it holds, the remote system accepts the login request. Otherwise, rejects the login request and terminates the operation. Then S computes D = h(T* ⊕ Ni ⊕ Bi ⊕ y), where T* is the time stamp.

Step V4: S → Ui: (Di, T*).

Step V5: Upon receiving the mutual authentication message at time T*, smart card verifies whether (T' - T*) d ≤ T, where T is an expected valid threshold. If it does not hold, smartcard terminates the session, else, it computes h(T* ⊕ Ni ⊕ Bi ⊕ y) and compares it with the received Di. If it holds, smart card confirms that he/she is communicating with valid S.

2.3 Password Change Phase
Whenever a user Ui wants to change his password, he can easily do that without requesting to the server. This phase works as follows:

Step P1: The user Ui inserts the smart card into the smart card terminal. He submits the password PWi and request to change the password.

Step P2: The user Ui then chooses a new password PWi*.

Step P3: The smart card computes

Ni* = Ni ⊕ h(PWi) ⊕ h(PWi*), which yields Ni* = h(PWi*) ⊕ h(x).

Step P4: Thus, the password PWi is changed to PWi* and Ni (stored in smart card) is replaced with Ni*.

3. Security Analysis of LIAO Et al Scheme
In this section we discuss the security weaknesses of Liao et al scheme.

3.1 The Login and Verification phase is completely insecure as described by Awasthi et al [1].

In Login phase if a user keys a random password P instead of his real password PWi, then CIDi, Bi and Ci will be computed using the random password P as follows:

CIDi = h(P) ⊕ h(Ni ⊕ y ⊕ T).

Bi = h(CIDi ⊕ h(P))

here if CIDi value, computed in previous step, is substituted then Bi will be

Bi = h(h(p) ⊕ h(Ni ⊕ y ⊕ T) ⊕ h(P)).

With reference to the property of XOR operation i.e. 1 ⊕ 1 = 0, it implies that h(p) ⊕ h(p) = 0
So, Bi = h(h(Ni ⊕ y ⊕ T)). This proves that Bi does not carry the user password. This Bi is XORRed in the next step.

Ci = h(T ⊕ Ni ⊕ Bi ⊕ y). Here Ci value does not have the password entered by the user. So, if the user enters correct or random password, the Ci value does not have any effect.

After these computations, the smart card sends a login message (CIDi, Ni, Ci, T) consists of the computed values and the current time of the server. The server then computes the following with the received data:
Verification Phase

\[ h(P) = C_{Di} \oplus h(N_i \oplus y \oplus T) \]

\[ B' = h(C_{Di} \oplus h(P)) \text{ here substitute the value of } h(p) \text{ computed in previous step, we get} \]

\[ B' = h(C_{Di} \oplus h(N_i \oplus y \oplus T)). \text{ Again using the same XOR property,} \]

\[ B' = h(h(N_i \oplus y \oplus T)). \text{ Then the server computes the values of } C' \]

\[ C' = h(T \oplus N_i \oplus B' \oplus y) \]

Since \( C' \) is equivalent to \( C_i \), the login request will be accepted. Hence even with any random password user / adversary may access the server.

3.2 The scheme cannot withstand the impersonation attack by [6]

In verification phase, Impersonation Attack works as:

Assume that an adversary has intercepted one of \( U_i \)'s previous login messages; say \( (C_{Di}, N_i, C_i, T) \). If the adversary attempts to impersonate \( U_i \) to login \( S \) at time \( T'' \) (\( > T \)), an Impersonation attack can be performed as:

Step I1: The adversary computes \( t = T \oplus T'' \) and \( N' = N_i \oplus \Delta t \).

Step I2: Adversary sends \( (C_{Di}, N_i, C_i, T'') \) to \( S \)

Step I3: Since \( T'' \) is valid, i.e., fresh, \( S \) will proceed to compute

\[ h(PW_i) = C_{Di} \oplus h(N_i \oplus y \oplus T'') \]

\[ = C_{Di} \oplus h((N_i \oplus y \oplus T) \oplus \Delta t)) \]

Next, \( S \) computes

\[ Bi = h(C_{Di} \oplus h(PW_i)); \quad Ci = h(T'' \oplus N_i \oplus Bi \oplus y) \]

\[ = h(T \oplus \Delta t \oplus N_i \oplus \Delta t \oplus Bi \oplus y). \]

Since the computed result equals the received \( C_i \), \( S \) accepts the adversary's login request.

3.3 The scheme does not protect against reflection attack

In order to login to server, the smart card sends a login request which is verified by server. Once the validity is checked, the server creates a mutual authentication message to be sent to smart card to prove that the smart card is communicating a valid server. This mutual authentication message is then verified by smart card.

Assume that an adversary has intercepted and blocked the message transmitted in Step L3 of login phase i.e., \( (C_{Di}, N_i, C_i, T) \). He can then attempt to impersonate \( S \) to \( U_i \) by performing reflection attack as follows.

Step RA1: The Adversary sends \((C_i, T)\) from the intercepted message to smart card

Step RA2: Upon receipt of \((C_i, T)\), if the time stamp \( T \) is valid the smart card believes that the server has authenticated it and hence it proceeds to compute \( h(T \oplus N_i \oplus Bi \oplus y) \) which is equivalent to the received \( C_i \).

Therefore, the adversary can successfully impersonate \( S \) to make the user believe that user is communicating with the valid server. Hence, Liao et al.'s scheme fails to achieve successful mutual authentication.

3.4 The scheme suffers from time concurrency problem

The time concurrency mechanism is used in some authentication schemes using smart cards to resist replay attacks. But, some drawbacks exist such as the different time zone, the delivery latency etc[14]. Further, clocks can become unsynchronized due to sabotage on or faults in the clocks or the synchronization mechanism, such as the overflows and dependencies on potentially unreliable clocks on remote sites[18-20]. Thus, it can be inferred that since Liao et al scheme uses the time stamps, it is practically very difficult to achieve the synchronization of clocks.

4. Proposed Scheme

The proposed scheme also consists of three phases, Registration, Authentication and Password Change phase. The notations are same as Liao et al. scheme. The password change phase is same as Liao et al.'s scheme.

4.1 Registration Phase

If a user \( U_i \) wants to register with a server, he chooses password \( PW_i \) and submits \( ID_i \) and \( h(PW_i) \) to server through secure channel.

Upon receiving the registration request, the remote system performs the following steps:

Step R1: \( S \) computes \( N_i = h(PW_i) \oplus h(x \oplus ID_i) \), where \( x \) is a secret key of the remote system and \( h(.) \) is the one way hash function.

Step R2: \( S \) personalizes the smart card with the parameters \( h(.) \), \( N_i \) and \( y \); where \( y \) is the remote system’s secret number stored in each registered user’s smart card.

Step R3: \( S \Rightarrow U_i: \text{Smart card.} \)

4.2 Authentication Phase

4.2.1 Login Phase: The user \( U_i \) inserts his smart card to the card reader of a terminal, and keys his Identity \( ID_i \) and password \( PW_i \). The smart card then performs the following computations:

Step 1: Computes \( C_{Di} = h(PW_i) \oplus h(N_i \oplus y \oplus h(PW_i)); \quad Bi = h(h(C_{Di}) \oplus h(PW_i)) \)

Step 2: Generates a random number \( p_1 \) and Computes \( Ci = h(p_1 \oplus Bi \oplus y) \)

Step 3: Computes \( Di = ID_i \oplus h(p_1) \)

Step 4: \( U_i \rightarrow S: (C_{Di}, C_i, Di, p_1) \)

4.2.2 Verification Phase: Upon receiving the login message \( (C_{Di}, C_i, Di, p_1) \), the server computes:

Step 1: Computes \( ID_i = Di \oplus h(p_1) \); \( h(PW_i) = C_{Di} \oplus h(h(x \oplus \oplus ID_i) \oplus y \oplus p_1) \).

Step 2: Computes \( Bi' = h(h(C_{Di}) \oplus h(PW_i)) \). It then checks whether received \( CI \) equals \( h(p_1 \oplus Bi' \oplus y) \). If it holds, the remote system accepts the login request. Otherwise, rejects the login request.
Step 3: S generates a random number $p_2$ and computes $M = h(B_i) \oplus h(p_1 + 1) \oplus p_2$.

Step 4: $S \rightarrow U_i$: $(M)$

Step 5: Upon receiving the mutual authentication message ‘M’, the smart card computes

$$p_2 = M \oplus h(B_i) \oplus h(p_1 + 1)$$

$$M' = h(B_i) \oplus h(p_1 + 1) \oplus p_2.$$ It then compares $M'$ with the received $M$. If it holds, it confirms that he communicates with valid $S$.

The Registration phase and Authentication Phase are depicted in Fig 1 & 2.

5. Security Analysis of Proposed Scheme

The proposed scheme addresses the security weaknesses of Liao et al. scheme. The advantages of the proposed scheme are explained as follows:

5.1 The scheme is secure against Impersonation attack

The impersonation attack discussed in section 3.2 does not work because,

In step 2 of verification phase, $h(PWi)$ is computed as

$$h(PWi) = CIDI \oplus h(h(x \oplus ID) \oplus y \oplus p_1)$$

Since, in the proposed scheme computation of $h(PWi)$ in step 2 of verification phases does not uses $Ni$ value as compared to step 2 of verification phase of Liao et al., the adversary cannot succeed. Moreover, to perform an impersonation attack the adversary must have the knowledge of server’s secret ‘x’ and smart card secret ‘y’, which are never transmitted during transmission.

5.2 The scheme is secure against reflection attack.

During Step L4 of login phase, even if the adversary copies and blocks the message sent by the smart card i.e., $(CIDI, Ci, Di, p_1)$, he cannot impersonate $S$, since he has no knowledge of the format of $M$ ($M$ is computed using new random number $p_2$ and $h(p_1 + 1)$ and XORed with $B_i$). Moreover, in order to calculate $M$ adversary must have the knowledge of $B_i$, $p_2$ which are never transmitted. Hence the proposed scheme can withstand reflection attack.

5.3 The scheme has better security strength and is completely secure.

The weakness discussed in section 3.1 has been addressed. Since the value of CIDI, Bi are computed differently. So, even if the adversary enters a random / wrong password, the proposed scheme does not authenticate such users. Moreover, since the proposed scheme uses nonce to resist replay attacks, the scheme does not have any time concurrency problems.

5.4 Loss of Smart Card

Even if the smart card is lost the attacker cannot retrieve the Server’s secret parameter i.e $h(x)$. To know the $h(x)$, the attacker must know the ID and Password of a user which is very difficult and impractical.

6. Functional Analysis of Proposed Scheme

<table>
<thead>
<tr>
<th></th>
<th>Our Scheme</th>
<th>Liao et al Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintains Verification Table</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>User can Freely Choose and Change</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Password</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Reflection Attack works</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Impersonation Attack works</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Password Independence weakness</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Time Concurrency problem</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Achieves Mutual Authentication successfully</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 1. Functional Analysis of the proposed scheme
7. Efficiency Analysis of the Proposed Scheme

The total no. of hash functions (computed by smart card + Server) in Liao et al scheme are 13 H, whereas in the proposed scheme the total no. of hash functions are 20 H which are 7 more than the Liao scheme.

Though, the proposed scheme has compromised with the no. of hash functions, it has not compromised with the security which is essential in ensuring the correct authentication which the Liao et al scheme has not achieved. Further, the hash functions itself are very lightweight computational algorithms to run on smart cards as compared to the cryptographic algorithms. Moreover, today's smart cards have much more computational capability where in the difference of 7 extra hash functions are not much significant.

6. Conclusion

This paper discussed the weaknesses of Liao et al. scheme and proposed an improved scheme which not only inherits the features of Liao et al. scheme but also addresses the weakness of their scheme. Additionally, the improved scheme achieves successful mutual authentication and has better security.

References


Mohammed Misbahuddin received his B.Tech Degree in Computer Science & Engineering from K.B.N. College of Engineering, Gulbarga University, Gulbarga, Karnataka, India in 2001; M.Tech in Software Engineering from JNTU, Anantapur, India in 2006. He is presently pursuing Ph.D. in Computer Science from JNTU, Hyd, India. His area of interest includes Network Security, Image Processing, Pattern Recognition and Software Engineering. He is working as Senior Staff Scientist in Center for Development of Advanced Computing (CDAC), Electronic City, Bangalore, India.

Dr. P. Premchand did his BE in CS from Osmania University College of Engineering, Hyderabad, M.Tech from JNU, Delhi and Ph.D from JNTU, Hyd. He is presently Professor in the Dept. of CSE at Univ. College of Engg, Osmania University, Hyd. His areas of interest include Databases, Computer Networks, Image Processing and Object Oriented Technologies.

Dr. A. Govardhan did his BE in Computer Science and Engineering from Osmania University College of Engineering, Hyd, M.Tech from JNU, Delhi and Ph.D from JNTU, Hyderabad. He is presently Professor in CSE at School of Information Technology, JNTU Hyderabad. He has guided numerous M.Tech projects MCA and B.Tech projects. He has 40 research publications at International/National Journals and Conferences. He has been a TPC member, reviewer for various International and National conferences. He has delivered number of Keynote addresses and invited lectures. He is also a member in various professional bodies. His areas of interest include Databases, Data Warehousing & Mining, Information Retrieval, Computer Networks, Image Processing and Object Oriented Technologies.