Weaknesses of a Dynamic ID Based Remote User Authentication Protocol for Multi-Server Environment

R. Madhusudhan, Adireddi Praveen
Department of Mathematical & Computational Sciences, National Institute of Technology Karnataka, Surathkal, India
Email: madhu_nitks@yahoo.com
Received December 2013

Abstract
Currently, smart card based remote user authentication schemes have been widely adopted due to their low cost and convenient portability. With the purpose of using various different internet services with single registration and to protect the users from being tracked, various dynamic ID based multi-server authentication protocols have been proposed. Recently, Li et al. proposed an efficient and secure dynamic ID based authentication protocol using smart cards. They claimed that their protocol provides strong security. In this paper, we have demonstrated that Li et al.'s protocol is vulnerable to replay attack, denial of service attack, smart card lost attack, eavesdropping attack and server spoofing attacks.

Keywords
Authentication; Smart Card; Dynamic ID; Multi-Server Environments; Password

1. Introduction
With the rapid growth of internet technologies and mobile communication services, remote user authentication is being more and more critical in order to prevent access to illegal users. Password based authentication is one of the simplest and the most convenient authentication mechanisms over remote access networks but it is not secure over insecure communication channels. Hence a large number of smart card based authentication protocols have been proposed to overcome the drawbacks of traditional password based protocols. These can be categorized as static ID [1-3] and dynamic ID based protocols. To achieve user’s anonymity, dynamic ID based authentication techniques [4-6] have been developed by many researchers.

In general, an efficient remote user authentication protocol should satisfy some functional and security requirements [7-11]. Based on the use of environment, authentications protocols can be divided into two categories: single-server and multi-server environments. Multi server architecture [12-15] provides the flexibility of using single registration across various different networks.

In 2009, Liao and Wang [16] proposed a dynamic ID based authentication protocol for multi-server environ-
ments. Hsiang and Shih [17] found that Liao-Wang’s protocol is vulnerable to insider’s attack, masquerade attack, server spoofing attack, registration center spoofing attack. To overcome these weaknesses Hsiang and Shih proposed an improved protocol. But Lee et al. [18] found that this protocol is also not secure and susceptible to masquerade attack and server spoofing attack. To overcome the weaknesses of Hsiang and Shih’s protocol, Lee et al. proposed an improved protocol and claimed that their protocol can resist all kinds of attacks.

In 2013, Li et al. [19] found that Lee et al.’s protocol is vulnerable to forgery attack, server spoofing attack and proposed a dynamic ID based authentication protocol for multi-server environments. They claimed that it is secure and can resist various attacks. However, in this paper we have demonstrated that Li et al.’s protocol is vulnerable to replay attack, denial of service attack, smart card lost attack, eavesdropping attack and server spoofing attacks.

The rest of this paper is organized as follows. In Section 2, we have given a brief review on Li et al.’s protocol. Section 3 provides the cryptanalysis of Li et al.’s protocol. Finally we conclude this paper in Section 4.

2. Review of Li et al.’s Protocol

The notations used in this paper are described in Table 1.

Li et al.’s protocol contains three participants, the user U_i, the server S_j, and the registration center RC. RC chooses the master secret key x and a secret number y to compute h(x||y) and h(SID_j||h(y)), and then shares them with S_j via a secure channel. There are four phases in the protocol: registration phase, login phase, verification phase, and password change phase.

2.1. Registration Phase

When the user U_i wants to access the services, the user U_i and the registration center RC need to perform the following steps to finish the registration phase:

1) U_i freely chooses his identity ID_i, the password PW_i, and computes A_i = h(b ⊕ PW_i), where b is a random number generated by U_i. Then U_i sends ID and A_i to the registration center RC for registration through a secure channel.

2) Now, Registration center, RC computes B_i = h(ID_i||x), C_i = h(ID_i||h(y)||A_i), D_i = h(B_i||h(x||y)), E_i = B_i ⊕ h(x||y). RC stores \{C_i, D_i, E_i, h(.), h(y)\} on the user’s smart card and sends it to user U_i via a secure channel.

3) User keys b into smart card and finally it contains \{C_i, D_i, E_i, b, h(.), h(y)\}.

2.2. Login Phase

Whenever U_i wants to login S_j, he must perform the following steps to generate a login request message:

<table>
<thead>
<tr>
<th>Table 1. Notations used.</th>
</tr>
</thead>
<tbody>
<tr>
<td>U_i</td>
</tr>
<tr>
<td>ID_i</td>
</tr>
<tr>
<td>PW_i</td>
</tr>
<tr>
<td>S_j</td>
</tr>
<tr>
<td>RC</td>
</tr>
<tr>
<td>SID_j</td>
</tr>
<tr>
<td>CID_i</td>
</tr>
<tr>
<td>x</td>
</tr>
<tr>
<td>y</td>
</tr>
<tr>
<td>h(.)</td>
</tr>
<tr>
<td>⊕</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>→</td>
</tr>
<tr>
<td>⇒</td>
</tr>
</tbody>
</table>
1) Ui inserts his smart card into the card reader and inputs IDi and PWi. Then the smart card computes A_i = h(b \oplus PWi), C_i^* = h(IDi||h(y)||A_i), and checks whether the computed C_i^* is equal to C_i. If they are equal, Ui proceeds the following steps. Otherwise the smart card aborts the session.

2) The smart card generates a random number N_i and computes P_{ij} = E_i \oplus h(h(SIDj||h(y))||N_i), CID_i = A_i \oplus h(D_i||SIDj||N_i), M_i = h(P_{ij}||CID_i||D_i||N_i) and M_2 = h(SIDj||h(y)) \oplus N_j.

3) Ui submits \{P_{ij}, CID_i, M_1, M_2\} to S_j as a login request message.

### 2.3. Verification Phase

After S_j receiving the login message \{P_{ij}, CID_i, M_1, M_2\}, S_j and Ui perform the following steps for mutual authentication and session key agreement:

1) S_j computes N_i = M_2 \oplus h(SIDj||h(y)), E_i = P_{ij} \oplus h(h(SIDj||h(y))||N_i), B_i = E_i \oplus h(x||y), D_i = h(B_i||h(x||y)) and A_i = CID_i \oplus h(D_i||SIDj||N_i) by using \{P_{ij}, CID_i, M_1, M_2\}, h(SIDj||h(y)) and h(x||y).

2) S_j computes h(P_{ij}||CID_i||D_i||N_i) and checks whether it is equal to M_1. If they are not equal, S_j rejects the login request and terminates this session. Otherwise, S_j accepts the login request message. Then S_j generates a nonce N_j and computes M_3 = h(D_i||A_i||N_j||SIDj), M_4 = A_i \oplus N_i \oplus N_j. Finally, S_j sends the message \{M_3, M_4\} to Ui.

3) After receiving the response message \{M_3, M_4\} sent from S_j, Ui computes N_j = A_i \oplus N_i \oplus N_j, h(D_i||A_i||N_j||SIDj) and checks this with the received message M_3. If they are equal, Ui successfully authenticates S_j. User Ui and the server S_j compute SK = h(D_i||A_i||N_j||SIDj), which is taken as their session key for further communication.

### 2.4. Password Change Phase

This phase is invoked whenever U_i wants to change his password PW_i to a new password PW\_new. There is no need for a secure channel for password change and it can be finished without communicating with the registration center RC.

1) Ui inserts smart card into the card reader and inputs ID_i and PW_i.

2) The smart card computes A_i = h(b \oplus PW_i), C_i^* = h(ID_i||h(y)||A_i), and checks whether the computed C_i^* is equal to C_i. If they are not equal, the smart card rejects the password change request. Otherwise, the user Ui inputs a new password PW\_new and a new random number b\_new.

3) The smart card computes A_{inew} = h(b\_new \oplus PW\_new) and C_{inew} = h(ID_i||h(y)||A_{inew}).

4) Finally, the smart card replaces C_i with C_{inew} to finish the password change phase.

### 3. Cryptanalysis of Li et al.’s Protocol

In this section, we demonstrate that Li et al.’s protocol is vulnerable to replay attack, denial of service attack, smart card lost attack, eavesdropping attack and server spoofing attacks.

#### 3.1. Vulnerable to Replay Attack

Assume that a malicious attacker can eavesdrop the communication channel and intercepts the message \{P_{ij}, CID_i, M_1, M_2\}. Now if he resends this message, server S does not verify the freshness of nonce, N_i and computes N_i = h(SIDj||h(y)) \oplus M_2, E_i = P_{ij} \oplus h(h(SIDj||h(y))||N_i), B_i = E_i \oplus h(x||y), D_i = h(B_i||h(x||y)) and compares with M_i. The condition satisfies and S accepts the login request.

Now, S computes M_3 = h(D_i||A_i||N_j||SIDj), M_4 = A_i \oplus N_i \oplus N_j and sends \{M_3, M_4\} to U_i. Here the attacker cannot find N_j but he is successful in wasting server’s valuable computing resources. A large number of replay attacks launched at the same time can also form denial-of-service attack.

#### 3.2. Vulnerable to Denial-of-Service Attack

An active attacker who is also a valid user knowing h(y) can fabricate the message M_2 using different nonce, say N_A and sends the fabricated message \{P_{ij}, CID_i, M_1, M_2\} to server, S_j where M_2 = h(SIDj||h(y)) \oplus N_A. After
performing the steps mentioned in 2.3.1, server $S_j$ rejects the login request of $U_i$, who is a legitimate user, as $h(P_{ij}||CID_i||D_i||N_i)$ does not equal to the received $M_1$. Hence, denial-of-service attack is possible.

3.3. Vulnerable to Smart Card Lost Attack and Password Guessing Attack

Assume that the user’s smart card has been lost or stolen. The attacker can extract the information $C_i, D_i, E_i, h(.)$, $h(y)$, $b$ from the smart card [20,21]. By previously intercepted message, attacker can find $N_i, E_i$ using the following calculations.

$$N_i = M_2 \oplus h(SID_j || h(y)),$$

$$E_i = P_{ij} \oplus h(h(SID_j || h(y)) || N_i).$$

Now, $A_i = CID_i \oplus h(D_i || SID_j || N_i)$.

Using offline dictionary attack, attacker can find the ID, password PW of $U_i$ by performing following operations:

1) Compare $C_i$ with $h(ID_{guess} || h(y) || A_i)$. Whenever it equals, $ID_{guess}$ is $ID_i$ of the user $U_i$

2) Compare $A_i$ with $h(b \oplus PW_{guess})$. Whenever it equals, $PW_{guess}$ is the original PW of $U_i$.

As the ID and Password are known, attacker can use the smart card impersonating the original user.

3.4. Vulnerable to Eavesdropping Attack

Assume that attacker found the smart card details. He can intercept the message $\{P_{ij}, CID_i, M_1, M_2\}$. He can find $N_i = h(SID_j || h(y)) \oplus M_2$ and $A_i = ID_i \oplus h(D_i || SID_j || N_i)$, intercepts the message $\{M_3, M_4\}$ from server and computes $N_j = A_i \oplus N_i \oplus M_4$. Now he acquires the session key, $SK = h(D_i || A_i || N_i || N_j || SID_j)$. Hence, the entire communication is compromised using this passive attack as the attacker has known the session key.

3.5. Vulnerable to Server Spoofing Attack

If we assume the attacker, A broke into a server or acquired a malicious server, then attacker have $h(x || y)$ and $h(SID_j || h(y))$. Attacker, A can masquerade as server, $S_j$ to spoof user, $U_i$.

After intercepting the login request message $\{P_{ij}, CID_i, M_1, M_2\}$, A can compute $N_i = h(SID_j || h(y)) \oplus M_2$, $E_i = P_{ij} \oplus h(h(SID_j || h(y)) || N_i)$, $B_i = E_i \oplus h(x || y)$, $D_i = h(B_i || h(x || y))$, $A_i = CID_i \oplus h(D_i || SID_j || N_i)$. A can choose a nonce, $N_a$ and compute $M_a = h(D_i || A_i || N_a || SID_j)$. $M_4 = A_i \oplus N_i \oplus N_a$. Then A sends the message $\{M_3, M_4\}$ to user $U_i$, $U_i$ computes $N_a = A_i \oplus N_i \oplus M_4$, and compares $M_4$ with $h(D_i || A_i || N_a || SID_j)$. Then $U_i$ computes mutual authentication message $M_5 = h(D_i || A_i || N_a || SID_j)$ and sends to attacker, A who is masquerading as $S_j$. Then A verifies $M_5$ and mutual authentication is done. Finally attacker, A and User, $U_i$ computes the session key, $SK = h(D_i || A_i || N_a || N_j || SID_j)$.

4. Conclusion

In this paper, we have shown that Li et al.’s dynamic ID based authentication protocol cannot resist many attacks and is vulnerable to replay attack, denial of service attack, smart card lost attack, eavesdropping attack and server spoofing attacks. We strongly feel that a remote user authentication protocol should provide security against the above mentioned attacks so that it can be used in the real world applications.

References


