Cryptanalysis of Two Dynamic ID-Based Remote User Authentication Schemes for Preserving User Privacy

Jian-Wen Peng¹, Wen-Bing Horng²*, Ying-Ching Chiu² and Chao-Sheng Liu²

¹Department of Commerce Technology and Management, Chihlee Institute of Technology, Taipei, Taiwan 220, R.O.C.
²Department of Computer Science and Information Engineering, Tamkang University, Tamsui, Taiwan 251, R.O.C.

Abstract

Remote user authentication is an essential part in electronic commerce to identify legitimate users over the Internet. However, how to protect user privacy in the authentication has become an important issue recently. Therefore, many secure authentication schemes with smart cards have been proposed. In this paper, we will analyze the security weaknesses of two recently proposed authentication schemes for preserving user privacy. First, Chang et al. (2011) proposed a robust and efficient remote user authentication scheme to provide user anonymity. However, this scheme fails to protect user privacy in terms of anonymity and traceability. In addition, it is vulnerable to the server counterfeit attack and it does not provide perfect forward secrecy for session keys. Furthermore, if the smart card is lost, it will suffer from the offline password guessing attack as well as the user impersonation attack. Second, Wen and Li (2012) recently presented an improved dynamic ID-based authentication scheme with key agreement. However, this scheme is vulnerable to traceability. In addition, it does not support perfect forward secrecy for session keys. Furthermore, the insecure offline password change phase and online secret renewal phase will result in the denial of service attack.

Key Words: Authentication, Cryptanalysis, Perfect Forward Secrecy, Session Key, Smart Card

1. Introduction

Remote user authentication is a verification mechanism which will identify legal users over insecure networks. It becomes an essential part in the rapid growth of electronic commerce. Because of its simplicity and convenience, the password-based remote authentication is one of the most promising techniques used to secure the Internet based applications. Since Lamport [1] proposed the first password-based authentication scheme over insecure networks, many research works [2-21] have been proposed to improve security and efficiency on remote user authentication.

Conventional remote user authentication schemes [2-11] transmit user’s ID in plaintext. However, this may cause the ID-theft attack; individual privacy information is leaked over the Internet and may be traced. To protect user privacy, Das et al. (2004) [12] first presented a dynamic ID-based remote user authentication scheme using secure one-way hash functions to protect user anonymity. However, as pointed out in [13-15], Das et al.’s scheme is insecure in that it is vulnerable to the offline password guessing attack and the user impersonation attack. Therefore, many improved schemes [15-20] have been proposed.

In this paper, we will analyze the security weaknesses of two recently proposed remote authentication schemes for protecting user privacy. Firstly, Chang et al. (2011) [19] indicated that Yeh et al.’s (2010) dynamic ID-based authentication scheme [17] is vulnerable to the replay attack, the user impersonation attack, and the offline password guessing attack. They then proposed an im-

*Corresponding author. E-mail: horng@mail.tku.edu.tw
proved version to remedy the above security pitfalls. However, we will show that Chang et al.’s improved scheme is vulnerable to the user traceability and is subject to revealing real identity, which cannot protect user privacy. In addition, it is vulnerable to the server counterfeit attack and cannot provide perfect forward secrecy for session keys. Furthermore, if the smart card is lost, the scheme is also vulnerable to the offline password guessing attack and the user impersonation attack. Therefore, Chang et al.’s scheme fails to achieve mutual authentication.

Secondly, Wen and Li (2012) [20] pointed out that Wang et al.’s (2009) [16] more efficient and secure dynamic ID-based remote authentication scheme does not provide user anonymity and is vulnerable to the user impersonation attack and offline password guessing attack. They then proposed an improved scheme to provide more functionalities. However, we will show that Wen-Li’s scheme is vulnerable to traceability. In addition, it does not provide perfect forward secrecy for session keys. Moreover, the insecure offline password change phase and online secret renewal phase may result in the denial of service attack.

The rest of the paper is organized as follows. In section 2, we review Chang et al.’s authentication scheme and analyze its security weaknesses. In section 3, we review Wen-Li’s authentication scheme and analyze its security weaknesses. We conclude the paper in the last section.

### 2. Review and Cryptanalysis of Chang et al.’s Scheme

#### 2.1 Review of Chang et al.’s Scheme

In this subsection, we briefly review Chang et al.’s scheme [19]. It applies the challenge-response mechanism to achieve mutual authentication. The security of the scheme is based on the one-way hash function. Chang et al.’s scheme, as shown in Figure 1, includes

<table>
<thead>
<tr>
<th>User U</th>
<th>Server S</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Registration Phase</strong></td>
<td><strong>Login and Authentication Phases</strong></td>
</tr>
<tr>
<td>Choose ID, PW</td>
<td>Compute ( N = h(r</td>
</tr>
<tr>
<td>Generate a nonce ( r )</td>
<td>Rotate ( Z = h(ID</td>
</tr>
<tr>
<td>Compute ( h(PW) )</td>
<td>Store ( { r, N, Z, h(r) } ) in a smart card</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
K & = h(r || x') \oplus n_1 \\
L & = ID \oplus h(n_1) \\
CID & = h(ID || x') \oplus n_i \\
\end{align*}
\]

\[
\begin{align*}
M_1 & = h(s || h(r || x)) \\
M_2 & = h(s || h(r || x)) \oplus n_1 \\
\end{align*}
\]

\[
\begin{align*}
M_3 & = h(s || h(r || x)) \oplus n_2 \\
M_4 & = h(s || n' \oplus h(r || x')) \\
SK & = h(n_1 || ID \ | \ n_i) \\
\end{align*}
\]

**Figure 1.** Chang et al.’s authentication scheme.
four phases: registration, login, authentication, and password change phases.

2.1.1 Registration Phase

When a user $U$ wants to register with the remote server $S$, he/she first selects his/her identity $ID$ and password $PW$ as well as a random number $r$. Then, the user $U$ submits $\{ID, h(PW), r\}$ to the server $S$ over a secure channel for registration, where $h(\cdot)$ is a secure one-way function.

After receiving the registration request, the server $S$ first computes $N = h(r || x) \oplus h(PW)$ and $Z = h(ID || x) \oplus h(PW)$, where $x$ is the long-term secret key of the server, ‘$\oplus$’ is a concatenation operator, and ‘$\oplus$’ is an exclusive-or (XOR) operator. Finally, the server $S$ issues to $U$ a smart card containing $\{r, N, Z, h(\cdot)\}$ over a secure channel.

2.1.2 Login and Authentication Phases

If the user $U$ wants to login with the server $S$, he/she inserts his/her smart card into a card reader of a terminal and enters his/her $ID$ and $PW$. After receiving the user $U$’s input, the smart card is responsible for establishing mutual authentication with the server $S$ as follows.

Step 1. The smart card first computes $h(r || x') = N \oplus h(PW)$ and $h(ID || x') = Z \oplus h(PW)$. It then generates a random nonce $n_1$ and computes $K = h(r || x') \oplus n_1, L = ID \oplus h(n_1)$, and $CID = h(ID || x') \oplus n_1$. Then, the smart card sends the login request message $\{CID, r, K, L\}$ to the server $S$.

Step 2. After receiving the message $\{CID, r, K, L\}$, the server $S$ first computes $M = h(r || x)$ by using the received $r$ and its own long-term secret key $x$. Next, it derives $n_1'$ and $ID'$ by computing $n_1' = K \oplus M$ and $ID' = L \oplus h(n_1')$, respectively. The server then computes $CID' = h(ID' || x') \oplus n_1'$ and verifies if it is equal to the received $CID$. If they are not equal, the server terminates the login request. Otherwise, it generates a random nonce $n_2$ and computes $M_{s1} = h(n_1' || h(r || x)) \oplus n_2$ and $M_{s2} = h(n_1' || n_2)$. Next, the server sends the response message $\{M_{s1}, M_{s2}\}$ to the user $U$’s smart card.

Step 3. Upon receiving the message $\{M_{s1}, M_{s2}\}$ from $S$, the smart card first derives $n_2'$ by computing $n_2' = M_{s1} \oplus h(n_1' || h(r || x'))$. It then checks whether $h(n_1 || n_2')$ is equal to $M_{s2}$. If they are not equal, the smart card terminates the session. Otherwise, it believes that the server $S$ is authentic. Finally, the smart card computes the response $M_U = h(n_2' + 1)$ and the session key $SK = h(n_2' || ID || n_1)$ and sends the response $\{M_U\}$ to the server.

Step 4. After receiving the message $\{M_U\}$, the server $S$ verifies the legitimacy of the user $U$ by comparing $h(n_2 + 1)$ and $M_U$. If they are not equal, the server terminates the session. Otherwise, it believes that $U$ is a legitimate user and computes the session key $SK = h(n_2 || ID || n_1')$ for encrypting/decrypting subsequent communicating messages.

2.1.3 Password Change Phase

When the user $U$ wants to change his/her old password $PW$, he/she first logs in with the server $S$. After successful mutual authentication, he/she enters a new password $PW_{new}$. The smart card then simply replaces $N$ with $N_{new} = N \oplus PW \oplus PW_{new}$ in its card memory.

2.2 Cryptanalysis of Chang et al.’s Scheme

In this subsection, we will analyze the security issues in Chang et al.’s scheme. The security of the scheme is based on the light-weight secure one-way hash function and the XOR operation. However, we will show that Chang et al.’s scheme is obviously vulnerable to traceability, which fails to protect user privacy. In addition, as assumed in Chang et al.’s scheme that the data stored in the smart card can be retrieved, the scheme cannot protect user anonymity, is vulnerable to the server counterfeit attack, and fails to provide perfect forward secrecy for session keys. Furthermore, if user’s smart card is lost, the scheme is also vulnerable to the offline password guessing attack and the user impersonation attack.

2.2.1 Vulnerability to Traceability

In each user $U$’s login request message $\{CID, r, K, L\}$, it contains a static (i.e., fixed) random number $r$ associated with $U$. Although the real identity of $U$ is not sent in the login request message, it is easy for an attacker to trace this particular user $U$ with the same value of $r$, as shown in [21]. Thus, Chang et al.’s scheme is vulnerable to user traceability.

2.2.2 Failure to Protect User Anonymity

As mentioned above, the user $U$ is associated with a fixed $r$ in each login request message. An attacker can further reveal the real identity $ID$ of user $U$ as follows. If
the attacker is not a legal user or he/she is a legal user but not with the same \( r \), the attacker can first use \( r \) to register with the server \( S \) (with \( ID_A \) and \( PW_A \)) as a legal user \( U_A \). He/she then receives a smart card containing \( \{ r, N_A, Z_A, h(\cdot) \} \), where \( N_A = h(r \parallel x) \oplus h(PW_A) \) and \( Z_A = h(ID_A \parallel x) \oplus h(PW_A) \). Once the attacker becomes a legal user with the same \( r \) as the user \( U \), he/she can launch the following attack to derive the user \( U \)'s real identity \( ID \).

In Chang et al.'s scheme, it is assumed that the data stored in the smart card can be obtained. Therefore, the attack can derive \( h(r \parallel x) \) from \( N_A \) in the smart card by computing

\[
N_A = h(r \parallel x) \oplus h(PW_A)
\]

Finally, the user \( U \)'s real identity \( ID \) can be derived by the attacker by computing \( ID = L \oplus h(n_1) \) using \( L \) in the login request message. Hence, Chang et al.'s scheme fails to protect user anonymity.

### 2.2.3 Vulnerability to Server Counterfeit Attack

Once the user \( U \)'s identity \( ID \), \( r \), and \( h(r \parallel x) \) are known, the attacker can masquerade as the server \( S \) to deceive the user \( U \) as shown in Figure 2. The attacker can first intercept user \( U \)'s login request message \( \{ CID, r, K, L \} \) in the insecure network. Next, he/she retrieves \( n_1 ' \) from \( K \) and \( ID ' \) from \( L \) and then checks whether \( ID ' \) is equal to \( ID \). If they are not equal, stop the session. Otherwise, the attacker generates a random nonce \( n_A \) and computes the response messages \( M_{A1} = h(n_1 ' \parallel h(r \parallel x)) \oplus n_A \) and \( M_{A2} = h(n_1 ' \parallel n_A) \). Finally, the masqueraded server sends \( \{ M_{A1}, M_{A2} \} \) to the user \( U \) for authentication. Surely, \( M_{A2} \) will successfully pass the checking of user \( U \)'s smart card. This is because the smart card will extract \( n_1 ' \) from \( M_{A1} \) and \( M_{A2} \) will be equal to \( h(n_1 \parallel n_1 ') \).

Furthermore, the masqueraded server can correctly compute the session key \( SK_A = h(n_1 \parallel ID \parallel n_1 ') \) to communicate with the user \( U \), because the attacker knows all the information (i.e., \( n_1 ' , ID , n_A \)) for computing \( SK_A \), which will be equal to the session key \( SK_U = h(n_1 ' \parallel ID \parallel n_1) \) computed by the user \( U \). Thus, the attacker will successfully masquerade as the server \( S \) to deceive the user \( U \).

### 2.2.4 Failure to Provide Perfect Forward Secrecy

The perfect forward secrecy means that if some of the long-term secret information of the user \( U \) (such as \( ID \) and/or \( PW \)) is compromised, all previous session keys \( SK \) can be derived. In Chang et al.'s scheme, even without knowing user \( U \)'s \( ID \) and \( PW \), previously used session keys can be easily derived as follows. Suppose that an attacker has obtained the user \( U \)'s \( ID \), \( r \), and \( h(r \parallel x) \) as before. Then, for each session of \( U \)'s login, the

<table>
<thead>
<tr>
<th>User U</th>
<th>Masqueraded Server (Attacker)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input ( ID, PW )</td>
<td><strong>known</strong>: ( { r, h(r \parallel x), ID } )</td>
</tr>
<tr>
<td>Compute ( h(r \parallel x)' = N_A \oplus h(PW_A) ) ( h(ID \parallel x)' = Z_A \oplus h(PW_A) )</td>
<td>Compute ( M = h(r \parallel x) ) ( n_1 ' = K \oplus M ) ( ID ' = L \oplus h(n_1 ') )</td>
</tr>
<tr>
<td>Generate a nonce ( n_1 ) Compute ( K = h(r \parallel x)' \oplus n_1 ) ( L = ID \oplus h(n_1) ) ( CID = h(h(ID \parallel x)' \parallel n_1) )</td>
<td>Compute</td>
</tr>
<tr>
<td>Compute ( n_1 ' = M_{A1} \oplus h(n_1 \parallel h(r \parallel x) \parallel x)' )</td>
<td>Verify if ( h(n_1 \parallel n_1 ') = ? M_{A2} ) Compute ( M_1 = h(n_1 ' + 1) ) ( SK_U = h(n_1 ' \parallel ID \parallel n_1) ) ( { M_1 } )</td>
</tr>
</tbody>
</table>

**Figure 2.** Server counterfeit attack on Chang et al.'s scheme.
login request message \{CID, r, K, L\} sent from user U and the response message \{MS_{S1}, MS_{S2}\} sent from the server S have all the required information to compute the session key SK. As before, the attacker can use K to derive n1 and then use L to derive ID. Similarly, he/she can use MS_{S1} to derive n2 by computing \( n_2 = MS_{S1} \oplus h(n_1 \| h(r \| x)) \). Once the attacker obtains n1, ID, and n2, he/she can easily compute the session key \( SK = h(n_2 \| ID \| n_1) \). Hence, Chang et al.’s scheme fails to provide perfect forward secrecy for session keys, even without knowing any user’s long-term secret information.

### 2.2.5 Vulnerability to Offline Password Guessing Attack

In Chang et al.’s scheme, if the user U’s smart card is lost (i.e., the well-known lost smart card problem), it is vulnerable to the offline password guessing attack as follows. Suppose that an attacker knows user U’s ID, r, and \( h(r \| x) \) as before. In addition, he/she also has U’s smart card. Therefore, the attacker can obtain \( N = h(r \| x) \oplus h(PW) \) from the smart card (by the assumption of Chang et al.’s scheme). Since password PW is chosen by the user, it is usually selected to be an easy-to-remember one (i.e., the weak password). Therefore, the attacker can launch an offline password guessing attack by selecting a PW* from a dictionary of all possible passwords and checking whether \( h(r \| x) \oplus h(PW^*) \) is equal to N. If they are not equal, try another one until the correct password PW* is guessed. Therefore, Chang et al.’s scheme is vulnerable to the offline password guessing attack for the lost smart card problem.

### 2.2.6 Vulnerability to User Impersonation Attack

Continuing the above discussion, if the attacker has the user U’s smart card, he/she can also masquerade as the user U to login with the remote server S as shown in Figure 3. Assume that the attacker has guessed the correct password PW. Then, he/she can extract \( h(ID \| x) \) from Z, stored in user U’s smart card, by computing \( h(ID \| x) = Z \oplus h(PW) \). Up to now, since the attacker has obtained r, \( h(r \| x) \) and \( h(ID \| x) \) of the user U, he/she can masquerade as U as follows. He/she first generates a random nonce \( n_A \) and computes \( K_A = h(r \| x) \oplus n_A \), \( L_A = ID \oplus h(n_A) \), and \( CID_A = h(h(ID \| x) \| n_A) \). Then, the attacker sends the forged login request message \{CID_A, r, K_A, L_A\} to the remote server S.

Upon receiving the login request message, the server S first computes \( M = h(r \| x) \) and derives \( n_A \) and ID’ by computing \( n_A' = K_A \oplus M \) and \( ID' = L_A \oplus h(n_A') \), respectively. The server then computes \( CID_A' = h(h(ID' \| x) \| n_A') \) and verifies if it is equal to the received CID_A. They are surely equal and the server generates a random nonce

---

**Figure 3.** User impersonation attack on Chang et al.’s scheme.
n_2 and computes \( M_{A1} = h(n'_2 \| h(r \| x)) \oplus n_2 \) and \( M_{A2} = h(n'_2 \| n_2) \). Next, the server sends the response message \( \{ M_{A1}, M_{A2} \} \) to the attacker.

After receiving the message, the attacker first derives \( n'_2 = M_{A1} \oplus h(n_4 \| h(r \| x)) \). He/she then computes \( M_{AU} = h(n'_2 + 1) \) and the session key \( SK_{A} = h(n'_2 \| ID \| n_A) \) and sends the forged response \( \{ M_{AU} \} \) to the server \( S \). Certainly, the faked \( M_{AU} \) will pass the authentication of the server because it is equal to \( h(n_2 + 1) \). Therefore, Chang et al.’s scheme is also vulnerable to the user impersonation attack for the lost smart card problem.

3. Review and Cryptanalysis of Wen-Li’s Scheme

3.1 Review of Wen-Li’s Scheme

The security of Wen-Li’s authentication scheme [20] is also based on the challenge-response mechanism and the secure one-way hash function. The scheme is composed of six phases: registration, login, authentication, revocation, password change, and secret renewal. Figure 4 illustrates Wen-Li’s authentication scheme.

3.1.1 Registration Phase

When a user \( U \) wants to register with the remote server \( S \), he/she first chooses his/her identity \( ID \) and password \( PW \) and then sends \( \{ ID, PW \} \) to \( S \) over a secure channel for registration.

Upon receiving the registration request, the server \( S \) first computes \( n = h(ID \| PW) \), \( m = n \oplus x \), and \( N = h(ID) \oplus h(PW) \oplus h(x) \oplus h(m) \), where \( x \) is the secret key of the server and \( n \) is kept by \( S \) for checking the validity of the smart card. Finally, \( S \) personalizes a smart card with \( \{ h(\cdot), N, n \} \) and sends it to \( U \) through a secure channel.

<table>
<thead>
<tr>
<th>User ( U )</th>
<th>Server ( S )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Registration Phase</strong></td>
<td></td>
</tr>
<tr>
<td>Choose ( ID, PW )</td>
<td>Compute ( n = h(ID | PW) )</td>
</tr>
<tr>
<td>( { ID, PW } )</td>
<td>( m = n \oplus x )</td>
</tr>
<tr>
<td>( { \text{smart card} } )</td>
<td>( N = h(ID) \oplus h(PW) \oplus h(x) \oplus h(m) )</td>
</tr>
<tr>
<td><strong>Login and Authentication Phases</strong></td>
<td>Save ( n ) in the registration table in ( S )</td>
</tr>
<tr>
<td>Input ( ID, PW )</td>
<td>Store ( { n, N, h(\cdot) } ) in a smart card</td>
</tr>
<tr>
<td>Compute ( A = h(ID) \oplus h(PW) )</td>
<td></td>
</tr>
<tr>
<td>( B = h(ID) \oplus h(PW) = h(x) \oplus h(m) )</td>
<td>Check if ( T' - T \leq \Delta T ) and ( n ) is valid (where ( T' ) is the current timestamp)</td>
</tr>
<tr>
<td>( CID = h(A) \oplus h(x) \oplus B \oplus h(N) \oplus T ) (where ( T ) is the current timestamp)</td>
<td>Compute</td>
</tr>
<tr>
<td></td>
<td>( m = n \oplus x )</td>
</tr>
<tr>
<td></td>
<td>( B = h(x) \oplus h(m) )</td>
</tr>
<tr>
<td></td>
<td>( A = N \oplus B = h(ID) \oplus h(PW) )</td>
</tr>
<tr>
<td></td>
<td>Verify if ( CID \oplus h(A) \neq ? )</td>
</tr>
<tr>
<td></td>
<td>( h(B) \oplus h(N) \oplus h(n) \oplus T )</td>
</tr>
<tr>
<td>Check if ( T'' - T'' \leq \Delta T ) (where ( T'' ) is the current timestamp)</td>
<td>Compute</td>
</tr>
<tr>
<td>(where ( T'' ) is the current timestamp)</td>
<td>( C'' = h(A \oplus T' \oplus h(n)) )</td>
</tr>
<tr>
<td>Verify if ( h(A \oplus T' \oplus h(n)) = ? C'' )</td>
<td>( SK = h(A | T | B | T') )</td>
</tr>
<tr>
<td>Compute</td>
<td>( KC'' = h(B | SK | T') )</td>
</tr>
<tr>
<td>( SK = h(A | T | B | T) )</td>
<td></td>
</tr>
<tr>
<td>Check if ( KC'' ) is correct</td>
<td></td>
</tr>
<tr>
<td>Compute</td>
<td></td>
</tr>
<tr>
<td>( KC = h(A | SK | T'') )</td>
<td>Check if ( KC = ? h(A | SK | T'') )</td>
</tr>
</tbody>
</table>

Figure 4. Wen-Li’s authentication scheme.
Note than in this scheme the server needs a verification table for storing the values of $n$.

### 3.1.2 Login Phase

When the user $U$ wants to login with the server $S$, he/she inserts his/her smart card into a card reader of a terminal and enters his/her ID and $PW$. The smart card first computes $A = h(ID) \oplus h(PW)$, $B = N \oplus h(ID) \oplus h(PW)$ ($= h(x) \oplus h(m)$), and $CID = h(A) \oplus h(n) \oplus B \oplus h(N) \oplus T$, where $T$ is the current timestamp. It then sends $\{CID, n, N, T\}$ to $S$ for login request.

### 3.1.3 Authentication Phase

After receiving the login request message $\{CID, n, N, T\}$ at time $T'$, the server $S$ performs the following steps:

**Step 1.** Check the validity of the timestamp $T$. If $T' > T$, $\Delta T$ or $n$ is not in the registered list, reject the login request, where $\Delta T$ is a time interval of transmission delay.

**Step 2.** Compute $m = n \oplus x, B = h(x) \oplus h(m)$, $A = N \oplus B$ ($= h(ID) \oplus h(PW)$). Verify whether $CID \oplus h(A) = h(B \oplus h(N) \oplus h(n) \oplus T)$. If it does not hold, terminate the login request.

**Step 3.** Compute $C' = h(A \oplus T' \oplus h(n))$, the session key $SK = h(A \| T \| B \| T')$, and the key confirmation code $KC' = h(B \| SK \| T')$. Finally, reply the message $\{C', KC', T'\}$ to the smart card of $U$.

Upon receiving the reply message $\{C', KC', T'\}$ at time $T''$, the smart card of user $U$ performs the following steps for authentication and key confirmation.

**Step 4.** Check the validity of timestamp $T'$. If $T'' > T' + \Delta T$, then terminate.

**Step 5.** Compute $h(A \oplus T' \oplus h(n))$ and verify if it is equal to $C'$. If it is not, stop.

**Step 6.** Compute the session key $SK = h(A \| T \| B \| T')$ and check whether the received $KC'$ is equal to $h(B \| SK \| T')$. If so, compute $KC = h(A \| SK \| T'')$ and send $\{KC, T''\}$ to the server $S$ for key confirmation.

After receiving $\{KC, T''\}$, the server $S$ verifies whether the received $KC$ is equal to $h(A \| SK \| T'')$. If it is, the scheme is complete.

### 3.1.4 Revocation Phase

If the smart card is lost or stolen, the user $U$ can revoke it as follows. The server $S$ first verifies $U$'s credentials by checking whether $n = h(ID \parallel PW)$ is in the registration table, where $PW$ is the original password during the registration phase. If it is, the server $S$ deletes $U$'s registration information including $n$ in the registration table. The user $U$ can change his/her ID or $PW$ and re-register with the server $S$, as performed in the registration phase.

### 3.1.5 Offline Password Change Phase

When the user $U$ wants to change his/her password, he/she inserts his/her smart card into the card reader of the terminal and inputs his/her identity $ID$, old password $PW$, and new password $PW^*$. Then the smart card computes $N^* = N \oplus h(PW) \oplus h(PW^*)$ ($= h(ID) \oplus h(PW^*) \oplus h(x) \oplus h(m)$) and replaces $N$ with the new value $N^*$.

### 3.1.6 Online Secret Renewal Phase

When the remote server $S$ wants to renew its secret key $x$ to enhance the security of the system, $S$ will interact with its clients as follows. When the user $U$ has successfully authenticated with the server $S$, the session key $SK$ is established to provide a private secure channel over a public network. Since $N$ and $n$ were received from $U$’s smart card in the login phase, the server $S$ computes $m = x \oplus n, m^* = x^* \oplus n$, and $N^* = N \oplus h(x) \oplus h(m) \oplus h(x^*) \oplus h(m^*)$, and sends $N^*$ to $U$’s smart card, where $x^*$ is the new secret key of $S$. Finally, the smart card replaces $N$ with $N^*$ upon receiving it.

### 3.2 Cryptanalysis of Wen-Li’s Scheme

#### 3.2.1 Vulnerability to Traceability

In Wen-Li’s scheme, each login request message $\{CID, n, N, T\}$ for the user $U$ contains two static (i.e., fixed) values, $n$ and $N$, which are stored in the smart card of $U$. As in Chang et al.’s scheme [19], although the real identity of $U$ is not transmitted in the login request message, it is easy for an attacker to trace this particular user $U$ with this pair of values $n$ and $N$. Thus, Wen-Li’s scheme is vulnerable to user traceability.

#### 3.2.2 Insecure Password Change Phase

In Wen-Li’s scheme, the password change phase is executed without checking the validity of input identity and password. Therefore, if a legal user $U$ wants to change his/her old password $PW$ to new one $PW^*$ but inputs wrong old password $PW^*$ by mistake, the smart
card will compute \( N' = N \oplus h(PW) \oplus h(PW*) = [h(ID) \oplus h(PW) \oplus h(x) \oplus h(n)] \oplus h(PW^*) \oplus h(PW*) \) and replace \( N \) with \( N' \). Note that \( h(PW) \) and \( h(PW^*) \) cannot be canceled by the XOR operation in \( N' \). Therefore, \( N' \) is incorrect. Thus, after password change the smart card can no longer be used to login with the server \( S \) because it will fail to pass the authentication of \( S \) due to \( CID \oplus h(A) \neq h(B \oplus h(N') \oplus h(n) \oplus T') \). Furthermore, if an attacker who has stolen \( U \)'s smart card changes to any new password and returns it to the owner, the user \( U \) can no longer uses it to login with the remote server \( S \). Therefore, this insecure password change phase may result in the denial of service (DOS) attack.

3.2.3 Insecure Secret Renewal Phase

During the online secret renewal phase, after mutual authentication and exchange of the session key \( SK \), the server \( S \) computes \( N^* = N \oplus h(x) \oplus h(m) \oplus h(x^*) \oplus h(m^*) \), where \( x^* \) is the new secret key of \( S \) and \( m^* = x^* \oplus n \). It then sends \( N^* \) to the user \( U \)'s smart card via the private secure channel established by \( SK \). However, an attacker may modify the value of \( N^* \) during the transmission because Wen-Li’s scheme does not check the integrity of \( N^* \). Therefore, if \( N^* \) is modified to \( N' \) and is used to replace old \( N \) value in \( U \)'s smart card, then the modified smart card can no longer be used to login with the server \( S \) since \( N' \) value is not valid. Hence, the lack of checking integrity of \( N^* \) may also result in the DOS attack.

3.2.4 Failure to Provide Perfect Forward Secrecy

As discussed in subsection 2.2.4, the perfect forward secrecy means that if some of the long-term secret information of the user \( U \) (such as \( PW \) ) is compromised, all previous session keys \( SK \) can be derived. In Wen-Li’s scheme, if \( PW \) is known to an attacker, all previous session keys will be compromised as follows. First, the attacker perform the offline ID-guessing attack to obtain the user \( U \)'s identity \( ID \) by checking the equality: \( CID = h(A) \oplus h(h(n) \oplus B \oplus h(N) \oplus T) \), where \( A = h(ID) \oplus h(PW) \) and \( B = N \oplus h(ID) \oplus h(PW) \). Since \( CID \), \( n \), \( N \), and \( T \) are transmitted in the login request message, the attacker can obtain all of them. Moreover, since \( PW \) is known, the attacker can guess correct \( ID \) by trying each possible \( ID^* \) and checking whether the above equality holds. Once the correct \( ID \) is obtained, the attacker can first compute \( A = h(ID) \oplus h(PW) \) and \( B = N \oplus h(ID) \oplus h(PW) \). Hence, all previous session keys can be derived by \( SK = h(A || T \parallel B || T') \), where \( T \) and \( T' \) are the timestamps obtained from each session. On the other hand, if \( ID \) is known, the attacker can launch the offline password guessing attack by checking the above equality to obtain the correct \( PW \). Once \( PW \) and \( ID \) are known, the attacker can derive previous session keys in the same way. Therefore, Wen-Li’s scheme fails to provide perfect forward secrecy for session keys.

3.2.5 Vulnerability to the Privileged-Insider Attack

In Wen-Li’s scheme, the user \( U \) presents his/her \( ID \) and \( PW \) in plaintext form to the server \( S \) for registration. In addition, the computed shared secret values \( n \) and \( N \) by the server \( S \), which are stored in \( U \)'s smart card, are also transmitted in the login request message over the network. However, if the system manager or a privileged-insider of the server knows \( U \)'s \( ID \) and \( PW \), he/she may impersonate the user \( U \) by using \( n \) and \( N \) without \( U \)'s smart card. Thus, this scheme is vulnerable to the privileged-insider attack.

3.2.6 Requiring Additional Maintenance Cost of the Registration Table

Over the last decade, it is a trend to develop new smart card-based authentication schemes without storing verification tables to enhance security. However, in Wen-Li’s scheme, they require the server to maintain a registration table storing each registered user \( U \)'s \( n \) value, where \( n = h(ID \parallel PW) \). This table is used to verify whether the login user has valid \( ID \) and \( PW \). It is a kind of verification tables. However, such kind of tables may suffer from the following drawbacks. First, it requires additional cost for maintenance. Second, it must be kept secretly without malicious modification from attackers. Thus, Wen-Li’s scheme requires additional maintenance cost and provides less security.

3.2.7 Problems in the Revocation Phase

In Wen-Li’s scheme, it allows users to revoke their smart cards through the server \( S \) in case of the card lost or stolen. The server \( S \) will first verify the user \( U \)'s credentials and check whether \( n = h(ID \parallel PW) \) is in the registration table. Note that \( PW \) in the value \( n \) is the original password provided in the registration phase, and the value \( n \) will not be changed even if the user \( U \) changes his/her password later (as shown in the offline password
change phase). However, when a user changes his/her password several times, he/she usually remembers only the current password and forgets the original password except he/she takes a note on somewhere. The inconsistency of password update in the offline password change phase of the scheme will result in some problems when users want to revoke their lost smart cards.

4. Conclusions

In this paper, we analyzed the security weaknesses of two recently proposed dynamic ID-based remote user authentication schemes for protecting user privacy. Firstly, we showed that Chang et al.’s (2011) robust and efficient remote user authentication scheme does not protect user privacy, including anonymity as well as traceability. In addition, it is vulnerable to the server counterfeit attack and it does not provide perfect forward secrecy for session keys. Moreover, for the lost smart card problem, the scheme will further suffer from the offline password guessing attack and the user impersonation attack. Therefore, it does not achieve mutual authentication.

Secondly, we showed that Wen-Li’s (2012) improved dynamic ID-based authentication scheme with key agreement is vulnerable to traceability. In addition, the insecure offline password change phase and the online secret renewal phase scheme may result in the denial of service attack. Furthermore, the scheme does not provide perfect forward secrecy for session keys.

As we know so far, a secure and robust remote authentication scheme with smart cards should have the following security features:

- without password or verification tables stored in the server,
- allowing users to freely select and change their passwords,
- providing mutual authentication,
- establishing session keys with perfect forward secrecy,
- protecting user privacy (including anonymity and non-trackability),
- withstanding various attacks (such as offline password guessing attacks, user impersonation attacks, server impersonation attacks, and so on),
- and more difficulty for the lost smart card problem (even if the smart card is lost and the secret information stored in it is disclosed, the scheme is still secure and robust).

Nevertheless, these security requirements are evolved over the last two decades with the development of designing secure authentication schemes. Thus, the value of finding pitfalls of the proposed schemes may help in designing new secure schemes.

Furthermore, it is interesting to observe that both Chang et al.’s and Wen-Li’s schemes and all their predecessors only apply light-weight operators (such as the secure one-way hash function and the XOR operator) to protect its security. It is conjectured that only using light-weight operations cannot achieve sufficient security for remote user authentication with smart cards.

Acknowledgements

This work was partially supported by the National Science Council, Taiwan, R.O.C., under Contract No. NSC 99-2511-S-032-004-MY3 and NSC 100-2632-E-032-001-MY3.

References


Authentication Scheme over Insecure Networks,” 


**Manuscript Received: Feb. 18, 2013**

**Accepted: Jun. 28, 2013**