An Enhanced Remote User Authentication Scheme Providing Mutual Authentication and Key Agreement with Smart Cards

Chun-Ta Li
Tainan University of Technology
Department of Information Management
529 Jhong Jheng Road, Tainan 71002, Taiwan, R.O.C.
th0040@mail.tut.edu.tw

Abstract

In 2005, Yoon et al. proposed a user-friendly remote user authentication scheme using smart cards. The security of their scheme is based on one-way hash function and they claimed that their scheme is secured from attacks and achievable for proving mutual authentication, freely choosing password, no verification tables, and involving very lightweight hashing operations. However, we find that Yoon et al.’s scheme suffers from the denial of service attack and performs only unilateral authentication (only user authentication). In this paper we consequently propose an enhanced version to eliminate the vulnerability. Furthermore, our enhanced scheme can also provide mutual authentication and key agreement between a remote server and login users.

Keywords: Cryptanalysis; Key agreement; Mutual authentication; Remote user authentication; Information security

1. Introduction

With the speedy growth of information science, more and more people use network technologies to communicate with people. However, an attacker may inject false packets, alter them, or re-send them on the communication channels between two parties and defeat the communication among users in the networks [6, 4, 7]. Is there a simple and secure way for people to use their own computer devices to access on-line resources over insecure channels. The security service for remote user authentication and session key agreement comes into being for communication networks. A easy-to-remember password authentication scheme is the most common method for a remote server to authenticate the login user and verify the validity of the login message. Several password authentication schemes using smart card have been proposed [2, 5, 8].

In 2004, Hwang and Li developed a simple remote user authentication scheme using smart cards [2]. The advantage of Hwang-Li scheme is that it does not need any password/verification tables in the remote server. However, in 2005, Yoon et al. proposed an improvement scheme because they found that Hwang and Li’s scheme was insecure and does not provide mutual authentication. Yoon et al.’s scheme provides an option for the users to change their old password to a new password in local without the help of the remote server. Besides, client and server are able to resist the parallel session attack [1] and the server spoofing attack [3] problems by providing mutual authentication. However, we find that Yoon et al.’s scheme is still vulnerable to the denial of service attack that an attacker without registering to the remote server can illegally login the remote server for gaining access right by intercepting legal user’s login message and the remote server is still unknown. The problem on their scheme is due to authentication is unilateral (only user authentication) and the remote server has no way to authentication the login user, so remote server cannot make trust on the originality of the login user. Therefore, the enhanced scheme applies challenge-response mechanism and provides key agreement to overcome such potential attacks. Moreover, the nonce mechanism is applied in our scheme to provide key agreement which enables the remote server and the login user to establish a session key common session key over an insecure channel. Key agreement is very important for sending secret messages using the session key between two communication parties. What is more, the nonce mechanism does not need the time stamp so as to avoid the problem of time synchronism between the server and the user. In fact, it is fairly complicated to achieve time synchronism and some problems exists such as the delivery latency and the different time zone, and so on [5].

The remainder of this paper is organized as follows: Section 2 briefly reviews Yoon et al.’s scheme, then shows its
security flaw in Section 3. Further the enhanced version of remote user authentication scheme is presented, while in the following sections the security and performance of the proposed scheme are discussed. Finally, some conclusions are given in Section 5.

\section*{2. Brief review of Yoon et al.’s scheme}

In this section, some notations used in Yoon et al.’s scheme [8] and this paper are defined in Table 1. There are three phases in Yoon et al.’s scheme and the detailed phases are briefly described as follows:

\subsection*{2.1. Registration phase}

Before the user $U_c$ logs in to the remote server $S_i$, $U_c$ submits their identifier $ID_c$ and chosen password $PW_c$ to $S_i$ over a secure channel. Upon receiving the registration request, $S_i$ performs the following steps.

\textbf{Step 1} : Computes $B_c = A_c \oplus PW^*_c$ and compares whether $B_c \subseteq V_c$ or not. If it does not hold, the smart card terminates the login procedure. Otherwise, it computes $C_1 = H(B_c,T_c)$, where $T_c$ is the current time of the input device.

\textbf{Step 2} : Sends the login message $m_1 = \{ID_c,C_1,T_c\}$ to the remote server.

\subsection*{2.2. Login phase}

If the user $U_c$ wants to login the remote server $S_i$, $U_c$ wants to login the remote server $S_i$, $U_c$ wants to login the remote server $S_i$ for gaining the access right, $U_c$ attaches his/her smart card to the card reader and keys in his/her identifier $ID_c$ and password $PW_c^*$, then the smart card performs the following steps.

\textbf{Step 1} : Computes $B_c = A_c \oplus PW^*_c$ and compares whether $B_c \subseteq V_c$ or not. If it does not hold, the smart card terminates the login procedure. Otherwise, it computes $C_1 = H(B_c,T_c)$, where $T_c$ is the current time of the input device.

\textbf{Step 2} : Sends the login message $m_1 = \{ID_c,C_1,T_c\}$ to the remote server.

\subsection*{2.3. Authentication phase}

Upon receiving the login message $m_1$ at $T_s$, where $T_s$ is the time of receiving $m_1$ at the remote server $S_i$, $S_i$ authenticates $U_c$ as follows.

\textbf{Step 1} : Checks the format of $ID_c$. If the format of $ID_c$ is incorrect, $S_i$ rejects $U_c$’s login request.

\textbf{Step 2} : Checks the validity of $T_c$. If $(T_s - T_c) \geq \Delta T$, $S_i$ rejects $U_c$’s login request.

\textbf{Step 3} : Computes $B_s = H(ID_c,T_{TSA},x)$ and $C'_1 = H(B_s,T)$ and verifies whether $C'_1 \subseteq C_1$. If it holds, $S_i$ accepts $U_c$’s login request and proceeds to Step 4; otherwise, $S_i$ rejects $U_c$’s login request.

\textbf{Step 4} : Acquires the current time stamp $T_s'$ and sends the message $m_2 = \{C_2 = H(B_s,C'_1,T'_s),T'_s\}$ to $U_c$.

Upon receiving the message $m_2$ at $T_s'$, $U_c$ verifies the validity of the time interval between $T'_s$ and $T'_c$. If it holds, $U_c$ computes $C'_2 = H(B_c,C'_1,T'_c)$ and verifies whether $C'_2 \subseteq C_2$ or not. If they are equal, $U_c$ believes that the communicating part is the real server and the unilateral authentication is achievable; otherwise, $U_c$ terminates the connection.

\section*{3. Denial of service attack on Yoon et al.’s scheme}

Let us consider the scenario of the denial of service attack, an attacker $U_a$ without registering to the remote server $S_i$ can freely masquerade as a legal user $U_c$ by intercepting the valid login message from the eavesdropped communication between $U_c$ and $S_i$ to cheat $S_i$ into believing $U_a$’s login request. The detailed steps of denial of service attack on Yoon et al.’s scheme are illustrated in Figure 1.

In the login phase, when a legal user $U_c$ sends the login message $\{ID_c,C_1,T_c\}$ to $S_i$, where $T_c$ is the current time stamp of $U_c$. $U_a$ intercepts $U_c$’s login message and replies it to $S_i$. Now $U_a$ can masquerade as $U_c$ to login $S_i$ for gaining access right. If the login message $\{ID_c,C_1,T_c\}$ is valid, $S_i$ will respond $\{C_2,T'_s\}$ to $U_a$ for unilateral authentication, where $T'_s$ is the current time stamp of $S_i$. In addition, $U_a$
drops this message and replies the fake response \( \{C'_2, T_{a}\} \) to \( U_c \), where \( C'_2 \) is a meaningless value and \( T_{a} \) is an invalid time stamp generated by \( U_a \). \( U_c \) terminates the connection as a result of unsuccessful verification. Moreover, \( S_i \) believes that \( U_a \) is an authentic user, while \( U_a \) is an attacker and becomes successful in deceiving the remote server \( S_i \). As a result, in Yoon et al.’s scheme, authentication is only one-way and the remote server has no way to authenticate the login user, so remote server cannot make trust on the originality of the login user. Finally, their scheme is susceptible to the denial of service attack.

4. The enhanced scheme and analyses

To overcome the denial of service attack, we propose an improvement on Yoon et al.’s scheme in this section. Registration phase in our enhanced scheme is the same as that in Yoon et al.’s scheme. The only difference between the enhanced scheme and Yoon et al.’s scheme is in login and authentication phases. Figure 2 illustrates the login and authentication phases in proposed scheme and the details of the enhanced scheme are described as follows.

4.1. Login phase

When \( U_c \) wants to login \( S_i \) for gaining the access right, \( U_c \) attaches his/her smart card to the card reader and keys in his/her identifier \( ID_c \) and password \( PW^*_{c} \), then the smart card performs the following steps.

**Step 1**: Computes \( B_c = A_c \oplus PW^*_c \) and compares whether \( B_c \neq V_c \) or not. If it is invalid, the smart card terminates the login procedure. Otherwise, it computes \( C_1 = B_c \oplus N_c \), where \( N_c \) is a nonce.

**Step 2**: Sends the login message \( m_1 = \{ID_c, C_1\} \) to the remote server.

4.2. Authentication phase

Upon receiving the login message \( m_1 \) from \( U_c \), \( S_i \) verifies \( U_c \) as follows.

**Step 1**: Checks the format of \( ID_c \). If the format of \( ID_c \) is incorrect, \( S_i \) rejects \( U_c \)'s login request.

**Step 2**: Computes \( B_s = H(ID_c, T_{TSA}, x) \), \( C_2 = C_1 \oplus B_s = N_c \) and \( C_3 = B_s \oplus N_s \) and \( C_4 = H(C_1||C_3||SK) \), where \( SK = H(B_c||C_2||N_s) \).

**Step 3**: Sends the message \( m_2 = \{C_3, C_4\} \) to \( U_c \) for unilateral authentication.

Upon receiving the message \( m_2 \) from \( S_i \), \( U_c \) authenticates \( S_i \) as follows.

**Step 4**: Computes \( C_5 = C_3 \oplus B_c = N_s \) and \( C_6 = H(C_1||C_3||H(B_c||N_c||C_5)) \), where \( H(B_c||N_c||C_5) \) is the common session key.

**Step 5**: Verifies whether \( C_6 = C_4 \) or not. If they are equal, \( U_c \) believes that the communicating part is the real server and the unilateral authentication is achievable; otherwise, \( U_c \) terminates the connection.

**Step 6**: When the user authentication is completed, \( U_c \) computes \( C_7 = H(N_c||C_5||B_c) \) and sends it to \( S_i \) for mutual authentication.
Upon receiving the message $C_7$ from $U_c$, $S_i$ authenticates $U_c$ as follows.

**Step 7**: Computes $C_8 = H(C_2||N_a||B_s)$.

**Step 8**: Verifies whether $C_8 = C_7$ or not. If they are equal, $S_i$ believes that the communicating part is the real user and the mutual authentication is achievable; otherwise, $S_i$ terminates the connection. Finally, both $S_i$ and $U_c$ are offered an implicit key authentication and the requirement of authenticated key agreement is achievable.

### 4.3. Security analysis

In this subsection, we analysis the security of the enhanced scheme. Consider the denial of service attack as mentioned in Section 3, in authentication phase of our enhanced scheme, an attacker $U_a$ must reply $C_7 = H(N_c, N_s, B_s)$ to convince $S_i$ in Step 6. Obviously, $U_a$ does not know the secret hash values $B_s = H(ID_s, TSA, x) = B_s$, $N_c$, and $N_s$ and the equation $C_7$ is computationally infeasible. Hence, both $S_i$ and $U_c$ will verify the validation of the hash function and mutual authentication between them is completed. As a result, denial of service attack cannot work in our enhanced scheme.

On the other hand, during the login phase, if an attacker $U_a$ intercepts the message $\{ID_c, C_1\}$ and modifies the message to $\{ID_c, C_1 + N_a\}$, where $N_a$ is a nonce chosen by $U_a$. For the server, this is a valid request with a different nonce $N_c + N_a$. However, this attack is still not work because $U_a$ is unable to compute the message $C_7 = H(N_c + N_a, N_s, B_s)$ to convince $S_i$ in Step 6 of authentication phase unless he/she knows the secret information $B_s = H(ID_s, TSA, x) = B_s$.

### 4.4. Performance analysis

In this subsection we evaluate the performance of our proposed scheme and compare it with Yoon et al.’s scheme in terms of efficiency and functionality in Table 2. For the total number of hashing operations performed during the registration, login, and authentication phases, our scheme requires two additional hashing operation in comparison with Yoon et al.’s scheme. However, in terms of functionality, our scheme can provide mutual authentication, session key agreement, and without synchronized clock.

### 5. Conclusions

In this paper, we have shown that Yoon et al.’s scheme performs only unilateral authentication and suffers from denial of service attacks. To resist this security problem, we have proposed an enhanced scheme, which solves denial of service attacks by providing mutual authentication. As a result, in our enhanced scheme, login user and remote server both can trust on the authenticity of each other. Furthermore, our scheme achieves the requirement of key agreement which generates a session key between two authenticated parties.

### Table 2. Comparisons of our scheme with Yoon et al.’s scheme

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Phases</th>
<th>Yoon et al.’s scheme</th>
<th>Our scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registration</td>
<td>1 H</td>
<td>1 H</td>
<td></td>
</tr>
<tr>
<td>Login</td>
<td>1 H</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Authentication</td>
<td>4 H</td>
<td>7 H</td>
<td></td>
</tr>
<tr>
<td>Mutual authentication</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Session key agreement</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Without synchronized clock</td>
<td>No</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

H: The number of hashing operation performed.  
- : None of hashing operation performed.

References