Efficient Remote Mutual Authentication and Key Agreement with Perfect Forward Secrecy

Han-Cheng Hsiang and Wei-Kuan Shih

1Department of Computer Science, National Tsing Hua University, Taiwan, Republic of China
2Department of Information Management, Vanung University, Taiwan, Republic of China

Abstract: This study will demonstrate both Juang’s scheme and Shieh-Wang’s scheme do not provide perfect forward secrecy and are vulnerable to a privileged insider’s attack. Besides, their scheme has the problem of slow wrong password detection and user cannot change his password freely. To remedy these flaws, this study proposes an efficient remote mutual authenticated and key agreement scheme with perfect forward secrecy. The proposed scheme not only provides perfect forward secrecy but also satisfies all the security requirements needed in remote mutual authentication and key agreement scheme.

Key words: Password, privileged insider’s attack, smart card, slow wrong password detection

INTRODUCTION

In distributed computing environment, secure communication in insecure communication networks is a very important issue. Hence, user authentication and secret key distribution become the most important security service for communication networks.

Remote user authentication scheme allows a server to check the legitimacy of a remote user through insecure network. In addition, a smart card based remote mutual authentication scheme is very useful to authenticate remote users (Juang, 2006; Wang et al., 2007). Hwang et al. (1990) initially proposed a non-interactive password authentication scheme and its enhanced version, which additionally uses smart cards. Since, then Tan and Zhu (1999), Yang and Shieh (1999) and Hwang and Li (2000) have proposed new schemes and improved the efficiency and security of remote authentication. For evaluating a strong protocol, perfect forward secrecy is considered to be an important security issue. A protocol providing perfect forward secrecy means that even if one entity’s long-term secret key is compromised, it will never reveal any old short-term keys used previously. (Menezes et al., 1997; Sun and Yeh, 2006).

In 2002, based on Sun’s (2000) scheme Chien et al. (2002) proposed an efficient remote mutual authentication scheme using smart card allowing server and user to authenticate each other. The advantages in the scheme include freely chosen passwords, no verification tables, low communication and computation costs. However, as demonstrated by Hsu (2004) and Chien et al.’s (2002) scheme is vulnerable to the parallel session attack. Thereafter, Juang (2004) proposed another scheme preserving all the advantages of Chien et al.’s (2002) scheme. Juang’s (2004) scheme is nonce based authentication and key agreement scheme. Hence, no synchronized clocks are required in the scheme. In addition, Juang’s scheme generates a session key for the user and server in their subsequent communication.

Recently, Shieh and Wang (2006) pointed out the weakness of Juang’s scheme and then proposed another similar scheme to improve the weakness. Shieh and Wang (2006) claimed that their scheme not only preserves all the advantages of Juang’s scheme but also improves its efficiency. Later, Yoon and Yoo (2007) demonstrated that Shieh-Wang’s scheme (Shieh and Wang, 2006) does not provide perfect forward secrecy (Menezes et al., 1997; Sun and Yeh, 2006) and is vulnerable to a privileged insider’s attack (Ku et al., 2005; Ku and Chen, 2004). Besides, we find that Shieh-Wang’s scheme has the problem of slow wrong password detection (Wang et al., 2007) and user cannot change his password freely. These flaws also exist in Juang’s (2006) scheme. To remedy these flaws, this study presented an efficient improvement on them with more security. Present scheme providing perfect forward secrecy, can withstand the previously proposed attacks. When the user inputs the wrong passwords, it can be immediately detected and the user can freely change his password. The proposed scheme not only inherits the merits of their scheme but also enhances the security of their scheme.

Corresponding Author: Han-Cheng Hsiang, Department of Computer Science, National Tsing Hua University, 101, Kuang Fu Road, Sec. 2, Hsin Chu, 300, Taiwan, Republic of China

366
REVIEW OF JUANG’S AND SHIEH-WANG’S SCHEMES

The notations in Table 1 are used throughout this study.

Review of Juang’s scheme: Juang’s (2004) scheme is based on the symmetric encryption and adopts nonce to avoid the time-synchronization problem of using time stamps. The scheme consists of two phases: the registration phase and the login and session key agreement phase.

Registration phase: Assume a user $U_i$ submits his identity $ID_i$ and password $PW_i$ to the server over a secure channel for registration. The server computes $V_i = h(ID_i, x)$, $W_i = V_i \oplus PW_i$, and issues $U_i$ a smart card containing $W_i$, $ID_i$, and $h(\cdot)$.

Login and session key agreement phase: When $U_i$ wants to login to the server, he inserts his smart card into a card reader and inputs his identity $ID_i$ and password $PW_i$.

- The smart card computes $V_i = W_i \oplus PW_i$, then sends the message $(N_i, ID_i, E_{\sigma}(r_{N_i}, C_i))$ to the server $S$, where, $C_i = h(ID_i \parallel N_i)$, $E_{\sigma}$ denotes a symmetric encryption algorithm using $V_i$ as the secret key, $N_i$ is a nonce and $r_{N_i}$ is a random value chosen by the smart card to generate the session key $K_i$.
- After receiving the message, $S$ computes $V_i = h(ID_i, x)$ and $(r_{N_i}, C_i) = D_{\sigma}(E_{\sigma}(r_{N_i}, C_i))$, where, $D_{\sigma}(\cdot)$ denotes the corresponding symmetric decryption algorithm of $E_{\sigma}$ using $V_i$ as the secret key. After decryption, if $C_i$ is not equal to $h(ID_i \parallel N_i)$, or $N_i$ is not fresh, the server rejects $U_i$'s request. Otherwise, the server sends the message $E_{\sigma}(r_N, N_i + 1, N_{N_i})$ to $U_i$, where, $N_i$ is a nonce and $r_N$ is a random value chosen by the server to generate the session key $K_i$.
- When $U_i$ receives the message, the smart card decrypts and checks whether $N_i + 1$ is in it. If yes, the smart card computes the session key $K_i = h(r_N, r_{N_i}, V_i)$ and sends the message $E_{\sigma}(N_i + 1)$ back to $S$.
- On receiving the last message, $S$ computes $D_{\sigma}(E_{\sigma}(N_i + 1))$ to check whether $N_i + 1$ is in it. If $N_i + 1$ is found, $S$ and $U_i$ have achieved mutual authentication and session key agreement.

<table>
<thead>
<tr>
<th>Table 1: Notations</th>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h(\cdot)$</td>
<td>Secure one-way hash function</td>
<td></td>
</tr>
<tr>
<td>$h_{\sigma}(\cdot)$</td>
<td>Represents a cryptographic keyed hash function with secret $k$</td>
<td></td>
</tr>
<tr>
<td>$x$</td>
<td>The secret key maintained by the server</td>
<td></td>
</tr>
<tr>
<td>$p$</td>
<td>A large prime number</td>
<td></td>
</tr>
<tr>
<td>$g$</td>
<td>A primitive element in $GF(p)$</td>
<td></td>
</tr>
<tr>
<td>$\oplus$</td>
<td>Exclusive-or operation</td>
<td></td>
</tr>
<tr>
<td>$\parallel$</td>
<td>String concatenation operation</td>
<td></td>
</tr>
</tbody>
</table>

Review of Shieh-Wang’s scheme: Shieh and Wang’s (2006) scheme consists of two phases: the registration phase and the login and key agreement phase. The scheme works as follows:

Registration phase: Assume a user $U_i$ submits his identity $ID_i$ and password $PW_i$ to the server over a secure channel for registration. If the request is accepted, the server computes $R_i = h(ID_i \oplus x) \oplus PW_i$ and issues $U_i$ a smart card containing $R_i$ and $h(\cdot)$.

Login and key agreement phase: When the user $U_i$ wants to login to the server, he first inserts his smart card into a card reader then inputs his identity $ID_i$ and password $PW_i$. The smart card then performs the following steps to begin an access session:

- Compute $a_{\sigma} = R_i \oplus PW_i$.
- Acquire current time stamp $T_{\sigma}$ store $T_{\sigma}$ temporarily until the end of the session and compute $MAC_{\sigma} = h(T_{\sigma} \parallel a_{\sigma})$.
- Send the message $(ID_i, T_{\sigma}, MAC_{\sigma})$ to the server and wait for response from the server. If no response is received in time or the response is incorrect, report login failure to the user and stop the session.

After receiving the message $(ID_i, T_{\sigma}, MAC_{\sigma})$ from $U_i$, the server performs the following steps to assure the integrity of the message, respond to $U_i$, and challenge $U_i$ to avoid replay:

- Check the freshness of $Tu$. If $Tu$ has already appeared in a current executing session of user $U_i$, reject $U_i$'s login request and stop the session. Otherwise, $Tu$ is fresh.
- Compute $a'_i = h(ID_i \oplus x), MAC_{\sigma} = h(T_{\sigma} \parallel a')$ and check whether $MAC_{\sigma}$ is equal to the received $MAC_{\sigma}$. If it is not, reject $U_i$'s login and stop the session.
- Acquire the current time stamp $T_{\sigma}$. Store temporarily paired time stamps $(T_{\sigma}, T_{\sigma})$ and ID, for freshness checking until the end of the session. Compute $MAC_{\sigma} = h(T_{\sigma} \parallel T_{\sigma} \parallel a')$ and session key $K_{\sigma} = h((T_{\sigma} \parallel T_{\sigma}) \oplus a')$. Then, send the message $(T_{\sigma}, T_{\sigma}, MAC_{\sigma})$ back to $U_i$ and wait for response from $U_i$. If no response is received in time or the response is incorrect, reject $U_i$'s login and stop the session.

On receiving the message $(T_{\sigma}, T_{\sigma}, MAC_{\sigma})$ from the server, the smart card performs the following steps to authenticate the server, achieve session key agreement and respond to the server:
Check if the received $T_s$ is equal to the stored $T_s$ to assure the freshness of the received message. If it is not, report login failure to the user and stop the session.

- Compute $\text{MAC}_s^t = h(T_s || T || a)$ and check whether it is equal to the received $\text{MAC}_s$. If not, report login failure to the user and stop. Otherwise, conclude that the responding party is the real server.
- Compute $\text{MAC}_s^w = h(T_s || (a+1))$ and session key $K_s = h(T_s || T_s || a)$. Then, send the message $(T_s, \text{MAC}_s^w)$ back to the server. Note that, in the message $(T_s, \text{MAC}_s^w)$, $T_s$ is a response to the server.

When the message $(T_s, \text{MAC}_s^w)$ from $U_i$ is received, the server performs the following steps to authenticate $U_i$ and achieve key agreement:

- Check if the received $T_s$ is equal to the stored $T_s$. If it fails, reject $U_i$’s login request and stop the session.
- Compute $\text{MAC}_s^w = h(T_s || (a+1))$ and check whether it is equal to $\text{MAC}_s^w$. If it is not, reject $U_i$’s login request and stop the session. Otherwise, conclude that $U_i$ is a legal user and permit the user $U_i$’s login.

At this moment, mutual authentication and session key agreement between $U_i$ and the server are achieved. From now on, the user $U_i$ and the server can use the session key $K_s$ in their further secure communication until the end of the access session.

**SECURITY FLAWS ON JUANG’S AND SHIEH-WANG’S SCHEMES**

Here, we will show that both Juang’s and Shieh-Wang’s scheme do not provide perfect forward secrecy (Menezes et al., 1997; Sun and Yeh, 2006) and are vulnerable to a privileged insider’s attack (Ku et al., 2005; Ku and Chen, 2004). Besides, the schemes have the problem of slow wrong password detection.

**Perfect forward secrecy problem:** In Juang’s scheme, if an attacker Eve obtains the secret key $x$ from the compromised server and intercepts transmitted values $(N_0, ID_0, E_0, a_0, C_0)$ and $E_0(x, N_0, T_0)$, from an open network. It will be easy to obtain the information since it is exposed over an open network. Then Eve can compute $V_i = h(ID_0, x)$ by using $ID_0$. After decryption, Eve can obtain $r_s$, $r_t$, and can compute the shared session key $K_s = h(r_s, r_t, V_i)$ by using $V_i, r_s$, and $r_t$.

In Shieh-Wang’s (2006) scheme, if an attacker Eve obtains the secret key $x$ from the compromised server and intercepts transmitted values $(ID_0, T_0, T_0)$, from an open network. It will be easy to obtain the information since it is exposed over an open network. Then Eve can compute $a_i = h(ID_0, x)$ by using $ID_0$, and can compute the shared session key $K_s = h(T_0 || T_0 || a_i)$ by using $a_i, T_0$, and $T_0$.

Once Eve obtains the shared session key, by using the shared session key, Eve can get all previous communication messages. Clearly, both Juang’s and Shieh-Wang’s (2006) scheme do not provide perfect forward secrecy.

**Privileged insider’s attack:** In real environments, a user uses the same password to access several servers for his convenience. If a privileged insider of the server has learned the user’s password, he may try to impersonate the user to access other servers. These situations can be found in the registration phase of Juang’s and Shieh-Wang’s scheme. For example, $U_i$’s password $PW_i$ will be revealed to the remote server $S$ because it is transmitted directly to $S$. Then, the privileged insider of $S$ may try to use $PW_i$ to impersonate $U_i$ to login the other servers that $U_i$ has registered with outside this system. If the targeted outside server adopts the normal password authentication scheme, it is possible that the privileged insider of the server can successfully impersonate $U_i$ to login it by using $PW_i$.

**Slow wrong password detection:** The Juang’s scheme and the Shieh-Wang’s scheme also have the problem of slow wrong password detection. If $U_i$ inputs a wrong password by mistake, the smart card still sends $U_i$’s login request unconditionally to server. The wrong password is not detected by the remote server $S$ checks $C_s = h(ID_s, || N_0)$ at the login and session key agreement phase of Juang’s scheme and checks $MAC_s = MAC_0$ at the login and key agreement phase of Shieh-Wang’s scheme. Therefore, their scheme is slow to detect the user’s wrong password. In other words, the password authentication is delayed and inefficient.

**THE PROPOSED SCHEME**

Here, we propose a remote mutual authentication and key agreement scheme with perfect forward secrecy. The proposed scheme is an improvement to Shieh-Wang’s (2006) scheme that can withstand the security flaws described above. The system selects two large prime numbers $p$ and $q$, such that $q$ divides $p-1$. One generator $g$ with order $q$ in the Galois field $GF(p)$, where $GF(p)$ is the set of integers $\{0, 1, \ldots, p-1\}$ with arithmetic operations defined modulo $p$. The proposed scheme consists of four phases: the registration phase, the login phase, the authentication and key agreement phase and the password change phase. The scheme works as follows.
Registration phase: Assume a user $U_i$ selects a random number $b$, password $PW$, and computes $h(b\oplus PW)$. He submits his identity $ID_i$ and $h(b\oplus PW)$ to the server over a secure channel for registration. If the request is accepted, the server computes $a_i = h(ID_i \oplus x_i), R_i = a_i \oplus h(b\oplus PW)$ and $N_i = h_i(h(b\oplus PW))$. Then issues $U_i$ a smart card containing $R_i, N, p, g, h(\cdot)$ and $h_i(\cdot)$. $U_i$ enters $b$ into his smart card so that he does not need to remember $b$ anymore.

Login phase: When the user $U_i$ wants to login to the server, he first inserts his smart card into a card reader then inputs his identity $ID_i$, password $PW$, and a large random integer $d\in\mathbb{Z}_p^\ast$. The smart card then performs the following steps to begin an access session:

- Compute $a_i = R_i \oplus h(b\oplus PW)$ and check whether $h_i(h(b\oplus PW)) \oplus N_i \not= N_i$ holds. If not, smart card terminates this session.
- Acquire current time stamp $T_{\sigma}$, store $T_i$ temporarily until the end of the session and compute $MAC_{ac} = h(T_{\sigma} \| a_i)$
- Compute $C = g^{a_i}$
- Smart card generates a random number $r$ and computes $C_r = a_r \oplus h(r\oplus b)$
- $U_i$ sends the message $(ID_i, T_{\sigma}, MAC_{ac}, C_r, C_r)$ to the server $S$ and wait for response from the server. If no response is received in time or the response is incorrect, report login failure to the user and stop the session.

Authentication and key agreement phase: After receiving the message $(ID_i, T_{\sigma}, MAC_{ac}, C_r, C_r)$ from $U_i$, the server $S$ performs the following steps to assure the integrity of the message, respond to $U_i$ and challenge $U_i$ to avoid replay:

- Check the freshness of $T_{\sigma}$. If $T_{\sigma}$ has already appeared in a current executing session of user $U_i$, reject $U_i$'s login request and stop the session. Otherwise, $T_i$ is fresh.
- Compute $a_i' = h(ID_i \oplus x_i)$, $C_i' = a_i' \oplus h(T_{\sigma} \| a_i')$ and check whether $MAC_{ac} = h(T_{\sigma} \| a_i')$ is equal to the received $MAC_{ac}$. If it is not, reject $U_i$'s login and stop the session.
- $S$ chooses a large random integer $e\in\mathbb{Z}_p^\ast$ and computes $g^e \equiv C \oplus a_i$. After all parameters are known, $S$ computes $g^{a_i} \equiv (mod\ p)$. $S$ retrieves $g^e$ by computing $g^e = C \oplus a_i$. After all parameters are known, $S$ computes $g^e \equiv (mod\ p)$. $S$ receives the raised $g^e$ to $e$.
- Acquire the current time stamp $T_{\sigma}$. Store temporarily paired time stamps $(T_{\sigma}, T_{\sigma})$ and $ID_i$ for freshness checking until the end of the session. Compute $MAC_{ac} = h(T_{\sigma} \| T_{\sigma} \| a_i), C = g^e \oplus a_i$ and $C$.

Password change phase: This phase is invoked whenever a user $U_i$ wants to change his password $PW$, with a new one, say $PW_{new}$.

- $U_i$ inserts his smart card into card reader, enters $ID_i$ and $PW$, and requests to change password
- $U_i$'s smart card computes $a_i = R_i \oplus h(b\oplus PW)$ and $N'_i = h_i(h(b\oplus PW))$.
Check whether $N^*$ equals to the stored $N$ or not. If not, reject the password change request, otherwise $U_i$ chooses a new password $PW_{new}$.

- Compute $R_{new} = a \oplus h(b \oplus PW_{new})$ and $N_{new} = h_c(h(b \oplus PW_{new}))$, then stores $R_{new}, N_{new}$ into the user’s smart card and replaces the old values $R, N$, respectively. The new password is successfully updated and this phase is terminated.

**SECURITY ANALYSIS**

Analyzes the security of present scheme as following.

The proposed scheme inherits the security features of Shieh-Wang’s (2006) remote user authentication and key agreement protocol. Challenge tokens $C_1, C_2, C_3$ are used to ensure authenticity of the server $S$ and the freshness of the communication and prevent replay attacks. Under these circumstances, the messages in the scheme are fresh and replays of old messages can be detected. Moreover, the protocol can withstand the attacks presented in previous sections.

**Claim 1:** The proposed scheme can provide perfect forward secrecy.

**Proof:** Present scheme is based on the following well-known hard problem, which is believed infeasible to solve in polynomial time. Given a prime $p$, a generator $g$ and two numbers $g^a \mod p$ and $g^b \mod p$, try to find $g^{ab} \mod p$, it is computationally infeasible due to the Diffie-Hellman (1976) problem. We assume that the user $U_i$’s password $PW_i$, the server $S$’s secret key $x$ are all known by an attacker. Then the attacker can decrypt $C_1$ to obtain $g_1^*$ and decrypt $C_2$ to obtain $g^*$. But he cannot calculate $g^*$ because the difficulty is similar to solve the Diffie-Hellman problem. So, the attacker does not have any opportunity to get the session key $K_{bc}$. Therefore, the session key is still secure. Hence, present scheme provides perfect forward secrecy with high security (Sun and Yeh, 2006).

**Claim 2:** The proposed scheme can resist privileged insider’s attack.

**Proof:** In the registration phase of the proposed scheme, a user $U_i$ selects a random number $b$, password $PW$, and computes $h(b \oplus PW)$. He submits $ID$, and $h(b \oplus PW)$ to the remote server $S$. If the privileged insider of $S$ wants to use $U_i$’s password to impersonate $U_i$ to login the other servers, the action will fail. Since $U_i$ registers to $S$ by presenting $h(b \oplus PW)$ instead of $PW$, the insider of $S$ cannot directly obtain $PW$. Furthermore, as $b$ is not revealed to $S$, the privileged insider of $S$ cannot obtain $PW$ by performing an off-line guessing attack on $h(b \oplus PW)$.

Thus, the proposed scheme can resist the privileged insider attack (Ku et al., 2005; Ku and Chen, 2004).

**Claim 3:** The proposed scheme is high efficiency in password authentication.

**Proof:** In login phase, if $U$ inputs an error password $PW^*$, the smart card computes $a = R_i \oplus (b \oplus PW^*)$ and checks equation $h_c(h(b \oplus PW^*)) \oplus = V$ in step 1. Obviously, the result is negative when $Pw \neq PW^*$ and smart card terminates the login session. Thus, the wrong password will be immediately detected by smart card yet need not wait for server authenticating, which results in high efficiency.

**Claim 4:** The user can freely change his/her password.

**Proof:** In present scheme, each user can choose her/his favorite password in the registration phase. It will make users easy to remember their own passwords. The proposed scheme also provides the mechanism of changing password. When any user wants to change password, the user’s smartcard will compute $N^* = h_c(h(b \oplus PW))$ and verify $N^*$ compares with the stored $N$ in smart card, respectively. If they are equal, i.e., $N^* = N$, the password change action will be executed. Otherwise, the smartcard rejects the password change request. Thus, the proposed scheme could effectively provide the mechanism of password change.

After above discussion, we summarize the comparisons of the proposed scheme with related schemes in Table 2.

**CONCLUSIONS**

This study have demonstrated that both Juang’s scheme and Shieh-Wang’s scheme do not provide perfect forward secrecy and are vulnerable to a privileged...
inside's attack. It is also found that the schemes have the problem of slow wrong password detection and user cannot change the password freely. Then, this research presented an efficient remote mutual authenticated and key agreement scheme with perfect forward secrecy to remedy these flaws. The important merits of the proposed scheme include: no verification table in the server, users being able to choose own password freely, both the server and the user can do mutual authentication, no synchronization required, generation of a session key, perfect forward secrecy, fast wrong password detection and users being able to change the password freely. After the analysis, it is concluded that the proposed scheme is more secure than the other schemes examined in this study.

REFERENCES


