A SECURITY ENHANCED REMOTE USER AUTHENTICATION SCHEME USING SMART CARDS

EUN-JUN YOON¹, SUNG-HO KIM² AND KEE-YOUNG YOOG²,*

1Department of Cyber Security
Kyungil University
33 Buho-Ri, Hayang-Ub, Kyungsan-Si, Kyungsangpuk-Do 712-701, Republic of Korea
ejyoon@kiu.ac.kr

2School of Computer Science and Engineering
Kyungpook National University
1370 Sankyuk-dong, Buk-gu Daegu 702-701, Republic of Korea
shkim@knu.ac.kr; *Corresponding author: yook@knu.ac.kr

Received November 2010; revised March 2011

ABSTRACT. A remote user authentication system has become an important part of security, along with confidentiality and integrity, for systems such as the Internet that offer remote access over untrustworthy networks. In 2006, Liaw et al. proposed an efficient and complete remote user authentication scheme using smart cards that includes a session key being agreed and an updated password phase. However, the current paper demonstrates that Liaw et al.'s scheme is vulnerable to some attacks and then presents an improved scheme in order to isolate such security problems.

Keywords: Security, Authentication, Smart card, Diffie-Hellman key agreement, Cryptography, Cryptanalysis

1. Introduction. Recently, a remote user authentication system has become an important part of security, along with confidentiality and integrity, for systems such as the Internet that offer remote access over untrustworthy networks [1-22]. In a remote password authentication scheme, based on knowledge of the password, a user can use it to create and send a valid login message to a remote system to gain the right to access. The remote system also uses the shared password to check the validity of the login message and authenticate the user. However, these remote password authentication schemes are vulnerable to password guessing attacks since most users usually choose easy-to-remember passwords. In 1981, a remote password authentication scheme was first proposed by Lamport [23] over an insecure channel. Since then, several schemes [24-43] have been proposed for improving security and achieving greater functionality.

In 2006, Liaw et al. [34] proposed an efficient and complete remote password authentication scheme using smart cards including an agreed session key and updated password phase. Their scheme had several merits: (1) the remote system does not need a dictionary of verification tables to authenticate users; (2) users can choose their passwords freely; (3) mutual authentication was achieved, between the user and the remote system; (4) the communication and computational costs are very low; (5) users can update their passwords after the registration phase; (6) a session key agreed by the user and the remote system can be generated in every session; and (7) the timestamp is discarded in order to avoid the serious time synchronization problem.
However, we found out that Liaw et al.’s scheme does not secure against some attacks [44-46]. It means that the scheme cannot practically be used for smart card-based authentication applications. Based on these motivations, the current paper demonstrates that Liaw et al.’s scheme is vulnerable to some attacks. That is, their session phase is vulnerable to a forgery attack, their registration phase is vulnerable to an insider attack and their updated password phase is vulnerable to a denial of service attack, where an unauthorized user can easily change the smart card password. Furthermore, we present an improved scheme in order to isolate and solve such security problems. Compared with Liaw et al.’s scheme, the proposed scheme can provide strong key agreement function with the property of perfect forward secrecy to reduce the computation loads for smart cards. As a result, compared with related authentication schemes, the proposed scheme has strong security and enhanced computational efficiency. Thus, the proposed scheme is extremely suitable for use in smart card-based authentication applications.

The remainder of this paper is organized as follows. Section 2 briefly reviews Liaw et al.’s remote user authentication scheme using smart cards. Section 3 demonstrates the security weaknesses of Liaw et al.’s scheme. The proposed authentication scheme is presented in Section 4, while Sections 5 and 6 discuss the security and efficiency of the proposed scheme. The conclusion is provided in Section 7.

2. Review of Liaw et al.’s Scheme. This section briefly reviews Liaw et al.’s remote user authentication scheme using smart cards [34]. The security of Liaw et al.’s scheme depends on the secure one-way hash function. Liaw et al.’s scheme consists of five phases: registration, login, verification, session and updated password phases.

2.1. Registration phase. Let $x$ be a secret key maintained by the remote system, $h(\cdot)$ be a secure one-way hash function [47, 48] with fixed-length output such as SHA-2 while $U_i$ denotes the $i$th user who submits his/her identity $ID_i$ and password $PW_i$ to the remote system for registration purpose. For $U_i$’s registration request, the remote system then performs the following operations:

1. Compute $U_i$’s secret information $v_i = h(ID_i, x)$.
2. Compute $e_i = v_i \oplus PW_i$, where $\oplus$ is a bit-wise exclusive-OR operation.
3. Write $h(\cdot)$ and $e_i$ into the memory of a smart card.
4. Issue the smart card to $U_i$.

Figure 1 shows Liaw et al.’s registration phase.

| Information held by User $U_i$: $ID_i$, $PW_i$. |
| Information held by Remote system: $x$. |

**User $U_i$**

Select $ID_i$, $PW_i$  

**Remote system**

$ID_i$, $PW_i$  

$v_i = h(ID_i, x)$  

$e_i = v_i \oplus PW_i$  

Store $e_i$, $h(\cdot)$ into $U_i$’s smart card

Smart card  

(Secure channel)

**Figure 1.** Liaw et al.’s registration phase
A SECURITY ENHANCED REMOTE USER AUTHENTICATION SCHEME

Shared Information: \(h(\cdot), E(\cdot), D(\cdot).\)
Information held by User \(U_i\): \(ID_i, PW_i\), smart card(\(e_i, h(\cdot)\)).
Information held by Remote system: \(x\).

<table>
<thead>
<tr>
<th>User (U_i)</th>
<th>Remote system</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login phase:</td>
<td></td>
</tr>
<tr>
<td>Input (ID_i) and (PW_i)</td>
<td></td>
</tr>
<tr>
<td>Generate (N_i)</td>
<td>(v_i = e_i \oplus PW_i)</td>
</tr>
<tr>
<td>Verification phase:</td>
<td>Check (ID_i)</td>
</tr>
</tbody>
</table>

\[
(N'_i, N'_s) = D_{v'_i}(M) \\
\text{Check } N'_i \overset{2}{=} N_i \\
N'_s
\]

**Figure 2.** Liaw et al.’s login and verification phases

2.2. **Login phase.** When \(U_i\) wishes to log into the remote system, he/she inserts the smart card into the terminal and types his/her identity \(ID_i\) and password \(PW_i\). The smart card then performs the following operations:

1. Generate a random nonce \(N_i\).
2. Compute \(v_i = e_i \oplus PW_i\).
3. Compute \(C = h(v_i, N_i)\).
4. Send an authentication request message \((ID_i, C, N_i)\) to the remote system.

2.3. **Verification phase.** After receiving the authentication request message \((ID_i, C, N_i)\), the remote system and smart card execute the following steps to facilitate a mutual authentication process between the user and the remote system. The remote system first performs the following operations:

1. Verify whether \(ID_i\) is a valid user identity: If not, the login request is rejected.
2. Compute \(v'_i = h(ID_i, x)\) and then check whether \(C = h(v'_i, N_i)\). If not, the request is rejected; otherwise, it proceeds to Step 3.
3. Generate a random nonce \(N_s\).
4. Create the encrypted message \(M = E_{v'_i}(N_i, N_s)\) by using \(v'_i\).
5. Send \(M = E_{v'_i}(N_i, N_s)\) to the smart card.

After receiving the message \(M\), the smart card then performs the following operations:

1. Compute \(v_i = e_i \oplus PW_i\).
2. Decrypt \(M\) by computing \(D_{v_i}(M)\) to derive \((N'_i, N'_s)\).
3. Verify whether \(N'_i \overset{2}{=} N_i\). If yes, \(N'_s\) is sent to the remote system. If no, the connection is disconnected.

After receiving the message \(N'_s\), the remote system verifies whether \(N'_s \overset{2}{=} N_s\) regarding the smart card. If yes, the mutual authentication process is complete. Figure 2 shows Liaw et al.’s login and verification phases.

2.4. **Session phase.** The security of a session phase is based on the Diffie-Hellman key exchange protocol [49]. In the session phase, a common session key is generated in order to encrypt an individual conversation between the user and the remote system within a session. The session phase involves two public parameters \(p\) and \(\alpha\), where \(p\) is a large
Shared Information: \( h(\cdot), E(\cdot), D(\cdot) \).

Information held by User \( U_i \): \( ID_i, PW_i \), smart card \((e_i, h(\cdot)), v_i, N_i \).

Information held by Remote system: \( x, v^0_i, N_s \).

User \( U_i \) \hspace{2cm} Remote system

\[
\begin{align*}
W_i &= \alpha^{N_i} \mod p & S_i &= \alpha^{N_s} \mod p \\
K_u &= (S_i)^{N_i} \mod p & K_s &= (W_i)^{N_s} \mod p \\
\text{Check } K_u &\equiv K_s & \text{Check } K_s &\equiv K_u \\
\text{Select a message } M_u &\hspace{2cm} \text{Select a message } M_s \\
M_s &= D_{v^0_i}(E_{v_i}(M_u \oplus K_u)) \oplus K_u & M_u &= D_{v^0_i}(E_{v_i}(M_u \oplus K_u)) \oplus K_u
\end{align*}
\]

**Figure 3.** Liaw et al.’s session phase

prime number and \( \alpha \) is a primitive element \( \mod p \). In order to agree a secure session key, the remote system and smart card perform the following operations:

1. The remote system computes \( S_i = \alpha^{N_s} \mod p \) and sends \( S_i \) to the smart card.
2. The smart card computes \( W_i = \alpha^{N_i} \mod p \) and sends \( W_i \) to the remote system.
3. The remote system computes \( K_s = (W_i)^{N_s} \mod p \) and the smart card computes \( K_u = (S_i)^{N_i} \mod p \). Then both determine whether \( K_s = K_u \). If yes, a new session is created. That is because

\[
K = (S_i)^{N_i} \mod p \\
= (\alpha^{N_s} \mod p)^{N_i} \mod p \\
= (\alpha^{N_s N_i} \mod p) \mod p \\
= (\alpha^{N_i} \mod q)^{N_s} \mod p \\
= (W_i)^{N_s} \mod p.
\]

4. If the remote system wants to send private data or message \( M_s \) to \( U_i \), it encrypts message \( E_{v^0_i}(M_u \oplus K_u) \) with \( v^0_i \) and sends it to \( U_i \). After \( U_i \) receives the message, the smart card decrypts the message and makes an exclusive operation to derive \( M_s \).
5. If \( U_i \) wants to send private data or message \( M_u \) to the remote system, it encrypts message \( E_{v_i}(M_u \oplus K_u) \) and sends it to the remote system. After the remote system receives the message, it decrypts the message and makes an exclusive operation to derive \( M_u \).

Figure 3 shows Liaw et al.’s session phase.

2.5. **Updated password phase.** If \( U_i \) wants to change his/her password from \( PW_i \) into \( PW'_i \) after registration, the following procedure is performed.

1. Calculate \( e'_i = e_i \oplus PW_i \oplus PW'_i = v_i \oplus PW'_i \).
2. Update $e_i$ on the memory of smart card to set $e_i^\prime$. That is done because

$$
e_i^\prime = e_i \oplus PW_i \oplus PW_i^\prime$$
$$= v_i \oplus PW_i \oplus PW_i^\prime$$
$$= v_i \oplus PW_i^\prime$$
$$= h(ID_i, x) \oplus PW_i^\prime.$$

Figure 4 shows Liaw et al.’s updated password phase.

| Information held by User $U_i$: $ID_i$, $PW_i$ |
| Information held by Smart card: $e_i$, $h(\cdot)$ |

**User $U_i$**

Input $PW_i$ and $PW_i^\prime$ → $(PW_i, PW_i^\prime)$ → $e_i^\prime = e_i \oplus PW_i \oplus PW_i^\prime$

Update $e_i$ with $e_i^\prime$

**Figure 4.** Liaw et al.’s updated password phase

3. Cryptanalysis of Liaw et al.’s Scheme. This section shows that Liaw et al.’s scheme has the following security flaws.

3.1. **Integrity violence of the session key due to illegal modification at the session phase.** Liaw et al.’s session phase is vulnerable to session key integrity violence due to illegal modification. In Steps 4 and 5 of the session phase, since $U_i$’s smart card and the remote system do not check the integrity of the derived private data or message $M_s$ and $M_u$, respectively, an attacker can easily conduct an illegal modification attack as follows. When the remote system sends encrypted message $E_{v_i}(M_s \oplus K_s)$ to $U_i$, the attacker intercepts and replaces it with a random nonce $X$. After $U_i$ receives the forged message $X$, the smart card will decrypt $X$ and make an exclusive operation to derive private data or message $M_s$. Since the derived $M_s$ is a random value, $U_i$ cannot receive the correct $M_s$. In addition, when $U_i$’s smart card sends an encrypt message $E_{v_i}(M_u \oplus K_u)$ to the remote system, the attacker intercepts and replaces it with a random nonce $X$. Upon receiving the forged message $X$, the remote system will decrypt $X$ and make an exclusive operation to derive private data or message $M_u$. Since the derived $M_u$ is also a random value, the remote system cannot receive the correct $M_u$. In fact, an illegal modification attack is not a serious attack, since it cannot prevent the two communication parties from reaching a common secret key, even though this key is not the correct one. Most important, the attacker cannot access the agreed common key as a result of this illegal modification attack. However, since the Diffie-Hellman session key $\alpha^{X,N_s}$ mod $p$ is invalid, it cannot guarantee the integrity of the session key. As a result, Liaw et al.’s session phase is vulnerable to session key integrity violence due to illegal modification procedures.

3.2. **Insider attack on the registration phase.** Liaw et al.’s registration phase is vulnerable to an insider attack. In practice, it is likely that user $U_i$ uses the same password $PW_i$ to access several servers for his/her convenience. If the intruder of the remote system has obtained the user’s password $PW_i$, he/she can impersonate the user $U_i$ to access other remote systems [29]. In the registration phase of Liaw et al.’s scheme, the user $U_i$ sends his/her password $PW_i$ to the remote system with plain-text. It is very easy to mount an insider attack because the system recognizes $U_i$’s password $PW_i$ and an insider attacker
may get it to login to other remote systems for the purpose of accessing data. Furthermore, if a user loses his/her smart card and it is located by the insider, or if the insider stole the user’s smart card, then the insider can easily impersonate the legitimate user by using the password $PW_i$ as well as the smart card at the login phase. In addition, if users use the same password for multiple accounts, those will be compromised as well. Although it is also possible that all the privileged insiders of the remote system are trusted and $U_i$ does not use the same password to access several servers, the implementers and users of the system should be aware of this potential weakness. As a result, Liaw et al.’s scheme is vulnerable to an insider attack.

3.3. Denial of service attack on the update password phase. When a smart card is stolen, an unauthorized user can easily create a new password for the smart card at Liaw et al.’s password change phase [30-32]. The attack can be performed as follows. First, an unauthorized user inserts $U_i$’s smart card into the smart card reader of a terminal, enters the $ID_i$ and $PW_a$, where $PW_a$ is the unauthorized user’s arbitrary password, and request a password change. Next, the unauthorized user enters an arbitrary new password $PW_i'$ and then the smart card will compute $e'_i = e_i \oplus PW_a \oplus PW_i'$, which yields $v_i \oplus PW_i \oplus PW_i$ and $PW_i$. Finally, the smart card will replace $e_i$ with $e'_i$ without any confirmation. Procedures being followed. If a malicious user stole user $U_i$’s smart card for a short period of time and changed to an arbitrary new password as above described, then the legal user $U_i$’s succeeding login requests will be denied unless he/she re-registers with the remote server again due to $C \neq h(v'_i, N_i)$ in regards to Step 2 of the verification phase. In addition, if user $U_i$ types an incorrect password $PW_{\text{wrong}}$ by mistake at the update password phase, then $U_i$’s smart card will compute meaningless $e'_i = e_i \oplus PW_{\text{wrong}} \oplus PW_i'$, which yields $v_i \oplus PW_i \oplus PW_{\text{wrong}} \oplus PW_i'$ and replaces it with existing $e_i$. As a result, the user $U_i$ cannot login to the remote system anymore by using the new password $PW_i'$ because the remote system always rejects $U_i$’s login request. As outlined, Liaw et al.’s password change phase is vulnerable to a denial of service attack.

3.4. Inefficiency for error password login. Even if $U_i$ inputs an error password in the login phase, the smart card will still send $U_i$’s login request unconditionally to the remote system. This error is not detected until the remote system checks $C = h(v'_i, N_i)$ at the authentication phase. Therefore, the password authentication procedure is delayed and inefficient.

4. Proposed Scheme. In this section, we propose improvements to Liaw et al.’s remote user authentication scheme using smart cards. The security of the proposed scheme also depends on a secure one-way hash function and its nonce-based scheme. The proposed scheme consists of five phases: registration, login, verification, session and updated password phases.

4.1. Registration phase. When a new user $U_i$ wants to access resources from the remote system, he/she must register in the remote system over a secure channel and perform the following operations:

1. Chooses his/her identity $ID_i$, password $PW_i$ and a random number $R$, then computes password verifier $vpw = PW_i \oplus R$.
2. Sends his/her identity $ID_i$ and password verifier $vpw$ to the remote system.

The remote system then performs the following operations:

1. Compute $U_i$’s secret information $v_i = h(ID_i, x)$.
2. Compute $e_i = v_i \oplus vpw$ and $vk_i = h(v_i, e_i)$.
3. Write $h(\cdot)$, $e_i$ and $vk_i$ into the memory of a smart card.
4. Issue the smart card to $U_i$.

After receiving the smart card, $U_i$ stores the random number $R$ in his/her smart card. Figure 5 shows the proposed registration phase.

4.2. Login phase. When $U_i$ wishes to log into the remote system, he/she inserts the smart card into the terminal and types his/her identity $ID_i$ and password $PW_i$. The smart card then performs the following operations:

1. Generate a random nonce $N_i$.
2. Compute $vpw' = PW_i \oplus R$ and $h(e_i \oplus vpw', e_i)$ and verify whether it is equal to the stored $vk_i$.
3. If it holds, compute $C = h(e_i \oplus vpw', N_i) = h(v_i, N_i)$.
4. Send an authentication request message $(ID_i, C, N_i)$ to the remote system.

4.3. Verification phase. After receiving the authentication request message $(ID_i, C, N_i)$, the remote system and smart card will execute the following steps to facilitate mutual authentication between the user and the remote system. The remote system first performs the following operations:

1. Verify that $ID_i$ is a valid user identity. If not, the login request is rejected.
2. Compute $v_i' = h(ID_i, x)$ and then confirm whether $C \overset{?}{=} h(v_i', N_i)$. If not, the request is rejected; otherwise, it proceeds to Step 3.
3. Generate a random nonce $N_s$.
4. Computes $h(v_i', N_i, N_s)$ and sends it back with $N_s$ to the smart card.

After receiving the message $h(v_i', N_i, N_s)$, the smart card then performs the following operations:

1. Computes $h(v_i, N_i, N_s)$ and then verifies whether $h(v_i, N_i, N_s) \overset{?}{=} h(v_i', N_i, N_s)$.
2. If yes, computes $h(v_i, N_s, N_i)$ which is sent to the remote system for mutual authentication. If no, the connection is disconnected.

After receiving the message $h(v_i, N_s, N_i)$, the remote system verifies whether $h(v_i', N_s, N_i) \overset{?}{=} h(v_i, N_s, N_i)$. If yes, the mutual authentication is complete. If no, the connection is terminated. Figure 6 illustrates the proposed login and verification phases.
Shared Information: $h(\cdot), E(\cdot), D(\cdot)$
Information held by User $U_i$: $ID_i, PW_i, Smart card(e_i, vk_i, R, h(\cdot))$
Information held by Remote system: $x$

**User $U_i$**

**Login phase:**
- Input $ID_i$ and $PW_i$
- $vpw' = PW_i \oplus R$
- Check $vk_i \overset{?}{=} h(e_i \oplus vpw', e_i)$
- Generate $N_i$
- $C = h(e_i \oplus vpw', N_i)$

**Verification phase:**
- $(ID_i, C, N_i)$

**Remote system**

- Check $ID_i$  
- $v'_i = h(ID_i, x)$
- Check $C \overset{?}{=} h(v'_i, N_i)$
- Generate $N'_s$
- Compute $h(v'_i, N'_i, N_s)$

- $N_s, h(v'_i, N_i, N_s)$

- Check $h(v_i, N_i, N'_s) \overset{?}{=} h(v'_i, N_i, N_s)$
- Compute $h(v_i, N_i, N'_s)$

- $h(v_i, N_s, N'_s)$

- Check $h(v'_i, N_s, N_i) \overset{?}{=} h(v_i, N_s, N'_s)$

**Figure 6. Proposed login and verification phases**

4.4. **Session phase.** This subsection describes how to confirm the shared session key $K$ is correctly computed by the remote system and $U_i$’s smart card unlike Liaw et al.’s session phase. The proposed session phase involves two public parameters $p$ and $\alpha$, where $p$ is a large prime number and $\alpha$ is a primitive element mod $p$. In order to agree a secure session key, the remote system and smart card perform the following operations:

1. The remote system computes $S_i = \alpha^{N_s} \mod p$ and sends $S_i$ to the smart card.
2. The smart card computes $W_i = \alpha^{N_i} \mod p$ and sends $W_i$ to the remote system.
3. The remote system computes $K_s = (W_i)^{N_s} \mod p$ and the smart card computes $K_u = (S_i)^{N_i} \mod p$. Then both the remote system and $U_i$’s smart card check whether $K_s = K_u$ by sending $h(v'_i, W_i, K_s)$ and $h(v_i, S_i, K_u)$, respectively. If yes, a new session is created due to the following:

$$K = (S_i)^{N_i} \mod p$$

$$= (W_i)^{N_s} \mod p$$

$$= \alpha^{N_sN_i} \mod p.$$  

4. If the remote system wants to send private data or a message $M_s$ to $U_i$, it encrypts message $E_{v'_i}(M_s \oplus K_s)$ with $v'_i$ and sends it and $h(M_s)$ to $U_i$. After $U_i$ receives the message, the smart card decrypts the message and makes an exclusive operation to derive $M'_s$. Finally, $U_i$ checks that hashed $M'_s$ is equal to the received $h(M_s)$. If yes, $U_i$ confirms the integrity of $M'_s$ and accepts it.

5. If $U_i$ wants to send private data or a message $M_u$ to the remote system, it encrypts message $E_{v_i}(M_u \oplus K_u)$ and sends it and $h(M_u)$ to the remote system. After the remote system receives the message, it decrypts the message and makes an exclusive operation to derive $M'_u$. Finally, the remote system checks whether the hashed $M'_u$ is equal to the received $h(M_u)$. If yes, the remote system confirms the integrity of $M'_u$ and accepts it.

Figure 7 shows the proposed session phase.
security features. The remaining features are the same as the original Liaw et al.’s scheme

Authentication scheme using smart cards. We only illustrate and discuss the enhanced

Updated password phase.

This section analyzes the security of the proposed remote user

If it holds, the smart card calculates

Select a message $M_u$

Select a message $M_s$

Check $h(M'_u) \overset{?}{=} h(M_s)$

Check $h(M'_u) \overset{?}{=} h(M_u)$

Figure 7. Proposed session phase

Information held by User $U_i$: $ID_i$, $PW_i$

Information held by Smart card: $e_i$, $vk_i$, $R$, $h(\cdot)$

User $U_i$

Input $PW_i$ and $PW'_i$  

$(PW_i, PW'_i)$

Check $h(e_i \oplus PW_i \oplus R, e_i) \overset{?}{=} h(vk_i)$

$e'_i = e_i \oplus PW_i \oplus PW'_i$

Update $e_i$ with $e'_i$

Figure 8. Proposed updated password phase

4.5. Updated password phase. If $U_i$ wants to change his/her password from $PW_i$ into $PW'_i$ after registration, the following procedure is performed.

1. Calculate $h(e_i \oplus PW_i \oplus R, e_i)$ and verify whether it is equal to the stored $vk_i$.
2. If it holds, the smart card calculates $e'_i = e_i \oplus PW_i \oplus PW'_i = v_i \oplus R \oplus PW'_i$.
3. Update $e_i$ on the memory of smart card to set $e'_i$. That is done because

$$e'_i = e_i \oplus PW_i \oplus PW'_i$$
$$= v_i \oplus vpw \oplus PW_i \oplus PW'_i$$
$$= v_i \oplus PW_i \oplus R \oplus PW_i \oplus PW'_i$$
$$= v_i \oplus R \oplus PW_i$$
$$= h(ID_i, x) \oplus R \oplus PW_i.$$  

Figure 8 shows the proposed updated password phase.

5. Security Analysis. This section analyzes the security of the proposed remote user authentication scheme using smart cards. We only illustrate and discuss the enhanced security features. The remaining features are the same as the original Liaw et al.’s scheme
as described in the research literature [34]. Readers are referred to [34] for more comprehensive references. First, we define the security terms [49-52] needed to conduct an analysis of the proposed scheme. They are as follows:

**Definition 5.1.** A weak secret key (user’s password $PW_i$) is the value of low entropy $W(k)$, which can be guessed in polynomial time.

**Definition 5.2.** A strong secret key (server’s secret key $x$) is the value of high entropy $S(k)$, which cannot be guessed in polynomial time.

**Definition 5.3.** The discrete logarithm problem (DLP) is explained by the following: Given a prime $p$, a generator $\alpha$ of $\mathbb{Z}_p^*$, and an element $R \in \mathbb{Z}_p^*$, find the integer $a$, $0 \leq a \leq p - 2$, such that $\alpha^a \equiv R \mod p$.

**Definition 5.4.** The Diffie-Hellman problem (DHP) is explained by the following: Given a prime $p$, a generator $\alpha$ of $\mathbb{Z}_p^*$, and elements $\alpha^a \mod p$ and $\alpha^b \mod p$, find $\alpha^{ab} \mod p$.

**Definition 5.5.** A secure one-way hash function $y = h(x)$ is one where given $x$ to compute $y$ is easy and given $y$ to compute $x$ is difficult.

The following four security properties must be considered for the proposed protocol; an illegal modification attack at the session phase, an insider attack at the registration phase, a secure password change, and incorrect password detection. Regarding the above mentioned definitions, the following theories are used to analyze the eight security properties of the proposed scheme.

**Theorem 5.1.** In the proposed session phase, an attacker cannot successfully initiate the forgery attack, which is described Subsection 3.1.

**Proof:** After decrypting the received $E_{\psi_i}(M_s \oplus K_s)$ from the remote system and $E_{\psi_i}(M_u \oplus K_u)$ from the $U_i$’s smart card in Steps 4 and 5 of the proposed session phase, $U_i$’s smart card and the remote system always check that the hashed $M'_s$ and $M'_u$ are equal to the received $h(M_s)$ and $h(M_u)$, respectively; no one can forge the private data or messages $M_s$ and $M_u$. Therefore, the proposed session phase is secure from the forgery attack, which is described in Subsection 3.1.

**Theorem 5.2.** In the proposed registration phase, an insider attacker cannot successfully initiate an insider attack, which is described Subsection 3.2.

**Proof:** Since $U_i$ registers to the remote system by presenting $ID_i$ and $PW_i \oplus R$ instead of $ID_i$ and $PW_i$ unlike Liaw et al.’s registration phase, the insider attacker of the remote system cannot directly obtain or guess the password $PW_i$ without knowing the random number $R$. Therefore, the proposed registration phase is secure from an insider attack, which is described in Subsection 3.2.

**Theorem 5.3.** In the proposed updated password phase, an unauthorized user cannot successfully initiate the denial of service attack, which is described Subsection 3.3.

**Proof:** Because the smart card can verify $h(e_i \oplus PW_i \oplus R, e_i)$ using the stored $vk_i$ in Step 1 of the proposed updated password phase, when the smart card has been stolen or lost, an unauthorized user cannot change the password because the card always verifies $h(e_i \oplus PW_i^* \oplus R, e_i)$ using the stored $vk_i$, where $PW_i^*$ is an unauthorized user’s guessed random password. Thus, no one can initiate a denial of service attack using the stolen or lost smart card. Therefore, the proposed updated password phase is secure from the denial of service attack, which is described in Subsection 3.3.
Table 1. Security properties of the proposed scheme and other related schemes

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>No verification table</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Freely chosen password</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Mutual authentication</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Lower communication and computation cost</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
<td>Medium</td>
<td>Low</td>
</tr>
<tr>
<td>Updated password</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Session key agreement</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Perfect forward secrecy</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Time synchronization</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Replay attack</td>
<td>Secure</td>
<td>Secure</td>
<td>Secure</td>
<td>Secure</td>
<td>Secure</td>
<td>Secure</td>
</tr>
<tr>
<td>Guessing attack</td>
<td>Secure</td>
<td>Insecure</td>
<td>Secure</td>
<td>Secure</td>
<td>Secure</td>
<td>Secure</td>
</tr>
<tr>
<td>Impersonation attack</td>
<td>Secure</td>
<td>Secure</td>
<td>Secure</td>
<td>Secure</td>
<td>Secure</td>
<td>Secure</td>
</tr>
<tr>
<td>Server spoofing attack</td>
<td>Secure</td>
<td>Secure</td>
<td>Secure</td>
<td>Secure</td>
<td>Secure</td>
<td>Secure</td>
</tr>
<tr>
<td>Illegal modification attack on the session phase</td>
<td>Insecure</td>
<td>No support</td>
<td>No support</td>
<td>No support</td>
<td>No support</td>
<td>Secure</td>
</tr>
<tr>
<td>Insider attack on the registration phase</td>
<td>Insecure</td>
<td>Insecure</td>
<td>Secure</td>
<td>Secure</td>
<td>Insecure</td>
<td>Secure</td>
</tr>
<tr>
<td>Denial of service attack on the updated password phase</td>
<td>Insecure</td>
<td>Secure</td>
<td>Secure</td>
<td>Insecure</td>
<td>No support</td>
<td>Secure</td>
</tr>
<tr>
<td>Wrong password detection</td>
<td>Slow</td>
<td>Fast</td>
<td>Fast</td>
<td>Slow</td>
<td>Slow</td>
<td>Fast</td>
</tr>
</tbody>
</table>

**Theorem 5.4.** In the proposed login phase, the incorrect input password can easily detected by the smart card without being revealed it to the remote system.

**Proof:** In Liaw et al.’s remote user authentication scheme, if user $U_i$ inputs an incorrect password by mistake, this wrong password will be detected by the remote system at the authentication phase. Therefore, Liaw et al.’s scheme is slow to detect the user’s incorrect password. In contrast to Liaw et al.’s scheme, at the proposed login phase, if user $U_i$ inputs the incorrect password by mistake, this incorrect password will be quickly detected by a smart card since the smart card can verify $vk_i = h(e_i \oplus vpw', e_i)$ using the stored $e_i$ and $vk_i$ in Step 2 of the login phase. Therefore, the proposed login phase quickly detects that an incorrect input password has been entered by the user.

We compared the proposed scheme with other related schemes [35, 36, 40, 42] as well as Liaw et al.’s scheme [34]. Table 1 shows the comparison results of the security properties of the proposed scheme and various other remote authentication schemes based on smart cards.

6. **Efficiency Analysis.** This section analyzes efficiency of the proposed scheme. Table 2 provides computational costs of the proposed scheme with various other related schemes [35, 36, 40, 42] as well as Liaw et al.’s scheme [34] in regards to the registration, login, verification, session and updated password phases.
Table 2. Computational costs of the proposed scheme and other related scheme

<table>
<thead>
<tr>
<th></th>
<th>Registration</th>
<th>Login</th>
<th>Verification</th>
<th>Session</th>
<th>Updated password</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed scheme</td>
<td>$2T(f)$</td>
<td>$2T(\oplus)$</td>
<td>$2T(f)$</td>
<td>$6T(f)$</td>
<td>$4T(ME)$</td>
</tr>
<tr>
<td></td>
<td>$2T(\oplus)$</td>
<td></td>
<td>$2T(\oplus)$</td>
<td></td>
<td>$4T(S)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$2T(f)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$4T(\oplus)$</td>
</tr>
<tr>
<td>Liaw et al.’s scheme [34]</td>
<td>$1T(f)$</td>
<td>$1T(\oplus)$</td>
<td>$2T(f)$</td>
<td>$4T(ME)$</td>
<td>$3T(f)$</td>
</tr>
<tr>
<td></td>
<td>$1T(\oplus)$</td>
<td></td>
<td>$2T(\oplus)$</td>
<td></td>
<td>$3T(\oplus)$</td>
</tr>
<tr>
<td>Cheng et al.’s scheme [35]</td>
<td>$2T(f)$</td>
<td>$(n+1)T(f)$</td>
<td>$(n+3)T(f)$</td>
<td>No support</td>
<td>$3T(f)$</td>
</tr>
<tr>
<td></td>
<td>$1T(\oplus)$</td>
<td></td>
<td>$2T(\oplus)$</td>
<td></td>
<td>$5T(\oplus)$</td>
</tr>
<tr>
<td>Wang et al.’s scheme [36]</td>
<td>$3T(f)$</td>
<td>$4T(f)$</td>
<td>$4T(f)$</td>
<td>No support</td>
<td>$4T(f)$</td>
</tr>
<tr>
<td></td>
<td>$3T(\oplus)$</td>
<td></td>
<td>$5T(\oplus)$</td>
<td></td>
<td>$4T(\oplus)$</td>
</tr>
<tr>
<td>Yang et al.’s scheme [40]</td>
<td>$5T(f)$</td>
<td>$1T(\oplus)$</td>
<td>$3T(ME)$</td>
<td>No support</td>
<td>$2T(f)$</td>
</tr>
<tr>
<td></td>
<td>$3T(\oplus)$</td>
<td></td>
<td>$1T(\oplus)$</td>
<td></td>
<td>$2T(\oplus)$</td>
</tr>
<tr>
<td>Xu et al.’s scheme [42]</td>
<td>$1T(ME)$</td>
<td>$3T(f)$</td>
<td>$6T(f)$</td>
<td>No support</td>
<td>No support</td>
</tr>
<tr>
<td></td>
<td>$2T(\oplus)$</td>
<td></td>
<td>$3T(\oplus)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$2T(ME)$</td>
</tr>
</tbody>
</table>

$T(f)$: computation cost of one-way function; $T(\oplus)$: computation cost of exclusive-OR operation or addition operation; $T(S)$: computation cost of symmetric encryption; $T(A)$: computation cost of asymmetric encryption; $T(ME)$: computation cost of modular exponentiation.

In the registration phase, Liaw et al.’s scheme requires 1 time one-way function operation and 1 time exclusive-OR operation. However, Liaw et al.’s registration phase is insecure to an insider attack. In the proposed registration phase, 1 time one-way function operation and 1 time exclusive-OR operation are additionally required to resist an insider attack compared with Liaw et al.’s scheme. In the verification phase, Liaw et al.’s scheme requires 2 times one-way function operations and 2 times symmetric encryption operations. However, in the proposed verification phase, it does not require any computation costs of symmetric encryption unlike Liaw et al.’s scheme. The proposed verification phase requires only 6 times one-way function operations. In the session phase, Liaw et al.’s scheme requires 4 times modular exponentiations, 4 times symmetric encryption operations, and 2 times one-way function operations. However, Liaw et al.’s session phase is insecure to forgery attacks. In the proposed session phase, 2 times one-way function operations and 2 times exclusive-OR operations are additionally required to resist the forgery attacks compared with Liaw et al.’s scheme. In the updated password phase, Liaw et al.’s scheme requires 2 times one-way function operations. However, Liaw et al.’s updated password phase is insecure to DoS attacks. In the proposed updated password phase, 1 time one-way function operation and 1 time exclusive-OR operation are additionally required to resist a stolen or lost smart card attack compared with Liaw et al.’s scheme.

Therefore, as in Table 2, we can see that the proposed scheme has the lowest computational costs and is well suited to the smart card’s applications.

7. Conclusion. In 2006, Liaw et al. proposed an efficient and complete remote password authentication scheme. Their scheme has several merits. However, the current paper demonstrated that Liaw et al.’s scheme is vulnerable to some attacks. We proved that their session phase is vulnerable to a forgery attack, that their registration phase is vulnerable to...
an insider attack and that their updated password phase is vulnerable to a denial service attack, where an unauthorized user can easily exchange a new password for the smart card. Furthermore, we presented an improved scheme in order to isolate such security problems. As a result, the proposed scheme is more secure than Liaw et al.’s scheme and provides similar computational efficiency.

Acknowledgment. We would like to thank the anonymous reviewers for their helpful comments in improving our manuscript. This research (Grants Nos. 00047239 and 00047610) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2011 and was also partially supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program (NIPA-2012-H0301-12-2004) supervised by the NIPA (National IT Industry Promotion Agency).

REFERENCES


