JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 29, 1121-1134 (2013)

DOI:10.1688/JISE.2013.29.6.4

Weakness and Improvement of the Smart Card Based Remote User Authentication Scheme with Anonymity*

YUNG-CHENG LEE
Department of Security Technology and Management
WuFeng University
Chiayi, 621 Taiwan

Today, people benefit various services through networks. However, due to the open environment of communications, networks are vulnerable to variety of security risks. Remote access capability is one of the critical functions for network systems. The remote user authentication scheme provides the server a convenient way to authenticate users before they are allowed to access database and obtain services. The smart card is one of the most reliable and efficient tools for remote user authentication. In some scenarios, remote user authentication schemes even require mechanisms to preserve user anonymity. In 2012, Shin *et al.* proposed a smart card based remote user authentication scheme. Their scheme has merits of providing user anonymity, key agreement, freely updating password and mutual authentication. They also claimed that their scheme can provide resilience to potential attacks of smart card based authentication schemes. In this article, we show that their scheme has several defects such as it cannot resist the impersonation attack, denial-of-service attack, off-line guessing attack and stolen-verifier attack. Furthermore, their scheme also suffers from high hash computation overhead and validations steps redundancy. We propose an improved scheme to overcome the drawbacks. The improved scheme has the merits of dynamic identity, user anonymity, forward and backward secrecy, mutual authentication, and low computation overhead. Moreover, the scheme can resist the replay attack, off-line guessing attack, smart card loss attack, impersonation attack and insider attack.

Keywords: authentication scheme, anonymity, smart cards, smart card loss attack, network security

1. INTRODUCTION

For modern people, network is the most common used platform to obtain various services. Remote access capability is one of the critical functions provided through networks. The remote user authentication scheme is a widely used mechanism for servers to identify and verify the users over insecure communication channel [1, 2, 12, 15-17, 19-23]. The smart card is one of the most reliable and efficient tools for remote user authentication. Until now, there are many remote user authentication schemes to identify the users with smart cards [2, 4, 7, 11, 13, 14, 17, 18, 21-23]. The remote users can access to the server for services after they are authenticated.

The general requirements of a smart card based authentication scheme are:

(1) The scheme can resist variety of attacks such as the insider attack, replay attack, guessing attack, stolen-verifier attack and impersonation attack, *etc.*

Received February 20, 2013; accepted April 19, 2013.
Communicated by Ruay-Shiung Chang and Sheng-Lung Peng.
* This work was partially supported by the National Science Council of Taiwan under the Contract No. NSC 01-2632-E-274-001-MY3.

(2) The scheme provides mutual authentication and ensures forward/backward secrecy.
(3) The user anonymity is ensured in some scenarios.
(4) The user can choose his/her identity and password freely.
(5) The user can update password freely.
(6) The server does not need to maintain a verification table.
(7) Due to power constraints of smart cards, the computational overhead should be low.
(8) The scheme provides session key agreement.

In 1981, Lamport [12] proposed the first remote password authentication scheme by using smart cards. However, Lamport's scheme has drawbacks such as high hash overhead and vulnerable to the stolen-verifier attack. Many schemes use one-way hash functions and exclusive-or operations to reduce the computing complexity in smart cards [3, 16, 19]. Hwang and Li [7] proposed a smart card based user authentication scheme in 2000. However, their scheme can not withstand the masquerade attack. In 2002, Chien *et al.* [4] presented a scheme with the merits of mutual authentication and freely updating password. But Ku and Chen [10] showed that Chien *et al.*'s scheme is vulnerable to the reflection attack and insider attack. They proposed an improved scheme to fix the flaws. However, Yoon *et al.* [23] indicated that the improved scheme was also susceptible to the parallel session attack and presented an improved scheme to enhance the security.

Chien and Chen [3] proposed an improved scheme to preserve user anonymity; however, Bindu *et al.* [1] showed that the scheme is vulnerable to the insider attack and the man-in-the-middle attack. Lin *et al.* [16] presented a strong password authentication protocol with one-way hash function. But the scheme is insufficient of mutual authentication and user anonymity. Juang [8] presents a simple authentication scheme in 2004, but the users cannot change passwords freely and it does not provide mutual authentication. Das *et al.* [5] and Liao *et al.* [15] introduced dynamic identity to achieve user anonymity, but both schemes are vulnerable to the insider attacks and neither scheme really provides user anonymity [17]. Khan *et al.* [9] and Tseng *et al.* [19] proposed remote authentication schemes to provide user anonymity. However, both schemes require time synchronization to resist the replay attack [17].

In 2012, Shin *et al.* [17] proposed a remote user authentication scheme with the merits of mutual authentication and user anonymity. The scheme overcomes the weaknesses of Das *et al.*'s scheme [5] and Liao *et al.*'s scheme [14]. However, in this article, we will show that Shin *et al.*'s scheme is vulnerable to the impersonation attack, denial-of-service attack, off-line guessing attack and stolen-verifier attack. We propose an improved scheme to enhance the security. The improved scheme has the merits of dynamic identity, user anonymity, forward and backward secrecy, mutual authentication, and low computation overhead. Moreover, the scheme resists variety of attacks such as the replay attack, off-line guessing attack, smart card loss attack, impersonation attack and insider attack.

The remainder of the article is organized as follows. All notations used throughout this article are described in section 2. Shin *et al.*'s smart card based remote user authentication scheme is briefly described in next Section. The security analysis of their scheme is shown in section 4. In section 5, we propose an improved scheme to enhance the security. The security analysis of the improved scheme is described in section 6. Finally, we make conclusions in section 7.

2. PRELIMINARIES AND NOTATIONS

For the remote authentication scheme, a user should register himself/herself to the server in advance if he/she wants to join the system. When a user intends to access the system for service, he/she has to login the system. The user's terminal or smart card sends login request to the server after the identity and password are keyed into the device. Upon receiving the login request, the server verifies the request information to authenticate the user. The server authorizes the user to access the system after the user is authenticated.

All notations used throughout this article are listed in Table 1.

Table 1. The notations used in Shin *et al.*'s scheme and the improved scheme.

Notations	Description
U_i	A legitimate user.
ID_i	The identity of the user U_i.
S	The server.
TID_i	The transform identity of the user U_i.
PW_i	The password of the user U_i.
K_S	The server's secret key.
K_U	The common key of user for S.
$h(\cdot)$	A one-way hash function.
$E_{SK}(M)$	To encrypt message M with secret key SK by using a symmetric cryptosystem.
T	A timestamp.
$\oplus$	An exclusive-or (XOR) operation.
$\|\|$	The concatenation operation.
$A \Rightarrow B$: $\{M\}$	The entity A sends message M to the receiver B via a secure channel.
$A \rightarrow B$: $\{M\}$	The entity A sends message M to the receiver B through a public channel.

3. SHIN *ET AL.*'S REMOTE USER AUTHENTICATION SCHEME

Recently, Shin *et al.* [17] proposed a remote user authentication scheme with user anonymity. Their scheme comprises four phases: registration phase, login phase, key agreement phase and password updating phase as follows.

3.1 Registration Phase

If the legitimate user U_i wants to join the system, the steps of the registration phase are as follows.

Step R-1: $U_i \Rightarrow S$: $\{ID_i, h(PW_i)\}$.

Firstly, the user U_i chooses identity ID_i and password PW_i. Then he/she submits $\{ID_i, h(PW_i)\}$ to the server S via a secure channel.

Step R-2: The server computes the user's TID_i, A_i and B_i.

After receiving $\{ID_i, h(PW_i)\}$, the server computes TID_i, A_i and B_i by:

$$TID_i = h(ID_i||h(PW_i)) \quad (1)$$
$$A_i = h(K_U) \oplus K_S \quad (2)$$
$$B_i = (g^{A_i} \bmod p) \oplus h(PW_i) \quad (3)$$

Where g is a primitive element in Galois field $GF(p)$, p is a large prime number. The server stores TID_i in the verification table.

Step R-3: $S \Rightarrow U_i$. Smart card.
The server stores $\{TID_i, B_i, h(\cdot), K_U\}$ in a smart card and sends it to the user.

3.2 Login Phase

If the user wants to log into the system, the login steps are as follows.

Step L-1: The user attaches smart card to a card reader and then keys in ID_i and PW_i.
Step L-2: $U_i \rightarrow S$: $\{DID_i, CTID_i, C_i, k_i\}$.
The smart card computes $\{CTID_i, C_i, M_i, DID_i\}$ as follows after generating two nonces n_i and k_i:

$$CTID_i = TID_i \oplus n_i, \quad (4)$$
$$C_i = h(B_i \oplus h(PW_i)) \oplus n_i, \quad (5)$$
$$M_i = K_U \bmod k_i \quad (6)$$
$$DID_i = h^{M_i}(TID_i \oplus h(B_i \oplus h(PW_i))). \quad (7)$$

The user sends $\{TID_i, CTID_i, C_i, k_i\}$ along with the login request information to the server.

Step L-3: The server checks the transform identity TID_i.
After receiving $\{DID_i, CTID_i, C_i, k_i\}$, the server computes A_i by:

$$A_i = h(K_U) \oplus K_S. \quad (8)$$

Since $C_i = h(B_i \oplus h(PW_i)) \oplus n_i = h(g^{A_i}) \oplus n_i$, the nonce n_i can be recovered by:

$$n_i = C_i \oplus h(g^{A_i} \bmod p) . \quad (9)$$

With n_i and $CTID_i$, the user's transform identity TID_i is obtained by:

$$TID_i = CTID_i \oplus n_i. \quad (10)$$

Then S checks whether the transform identity TID_i is in the database. If it isn't, the server terminates the connection; otherwise, continue the next steps.

Step L-4: The server authenticates the legitimate user.

The server computes $M_i = K_U \bmod k_i$ and obtains DID'_i by:

$$DID'_i = h^{M_i}(TID_i \oplus h(g^{A_i} \bmod p)). \tag{11}$$

Then the server checks whether DID'_i is equal to the received DID_i. If $DID'_i = DID_i$, S authenticates the user U_i. Otherwise, S will terminate the connection.

Step L-5: $S \rightarrow U_i$: $\{DID_S, CTID_S\}$.

After the user is authenticated, the server generates a nonce n_S and computes $\{DID_S, CTID_S\}$ by:

$$DID_S = h\{DID_i \oplus n_i \oplus n_S\}. \tag{12}$$
$$CTID_S = CTID_i \oplus n_S. \tag{13}$$

The server forwards $\{DID_S, CTID_S\}$ to U_i.

Step L-6: The user U_i authenticates the server S.

On receiving $\{DID_S, CTID_S\}$, the user obtains n'_S by:

$$n'_S = CTID_S \oplus CTID_i. \tag{14}$$

Thereby, U_i computes DID'_S with:

$$DID'_S = h(DID_i \oplus n_i \oplus n'_S). \tag{15}$$

If $DID'_S = DID_S$, the user authenticates the remote server. Otherwise, U_i will terminate the login steps.

Step L-7: $U_i \rightarrow S$: $\{DIS_{iS}\}$.

After S is authenticated, U_i sends DIS_{iS} to S, where:

$$DID_{iS} = DID_S \oplus n_i \oplus (n_S + 1). \tag{16}$$

Step L-8: The server S authenticates the user U_i.

Upon receiving DIS_{iS}, the server obtains $(n_S + 1)'$ by:

$$(n_S + 1)' = DID_{iS} \oplus DIS_S \oplus n_i. \tag{17}$$

The server S computes $(n_S + 1)$ and compares it with $(n_S + 1)'$. If $(n_S + 1)' = (n_S + 1)$, the mutual authentication is obtained. Otherwise, S will terminate connection with U_i.

3.3 Key Agreement Phase

After mutual authentication is obtained, the user and the server compute common session keys SK_i and SK_S, respectively, by:

$$SK_i = h(B_i \oplus h(PW_i) \oplus n_i \oplus n_S), \tag{18}$$

$$SK_S = h((g^{A_i} \bmod p) \oplus n_i \oplus n_S). \tag{19}$$

The generated common session keys of SK_i and SK_S are the same since $B_i \oplus h(PW_i) = g^{A_i}$.

3.4 Password Updating Phase

When the user wants to change his/her password, the steps are as follows.

Step U-1: $U_i \rightarrow S$: $\{DID_i, CTID_i, C_i, k_i$, Password updating Request$\}$
Similar to the login steps, the user attaches the smart card to a reader and forwards $\{DID_i, CTID_i, C_i, k_i$, Password updating Request$\}$ to the server.

Step U-2: The user and the server obtain mutual authentication.
Similar to the steps in the login phase, U_i and S obtain mutual authentication.

Step U-3: $U_i \rightarrow S$: $\{E_{SKi}(TID_i^*)\}$
After the mutual authentication is obtained, U_i chooses and keys in a new password PW_i^*. The smart card computes new transform identity TID_i^* by $TID_i^* = h(ID_i||h(PW_i^*))$. Following, the smart card encrypts TID_i^* by using the session key SK_i and sends $E_{SK_i}(TID_i^*)$ to the server.

Step U-4: The server replaces TID_i with TID_i^* in the database.
After receiving $E_{SK_i}(TID_i^*)$, S decrypts it by using SK_S and replaces TID_i with TID_i^*. Next, S sends response message to U_i.

Step U-5: U_i replaces TID_i and B_i with TID_i^* and B_i^*, respectively.
After receiving the response message from S, U_i computes $B_i^* = B_i \oplus h(PW_i) \oplus h(PW_i^*)$. Then the user replaces TID_i and B_i with TID_i^* and B_i^*, respectively.

4. SECURITY ANALYSIS OF SHIN *ET AL.*'S SCHEME

In Shin *et al.*'s scheme, the smart card computes $DID_i = h^{M_i}(TID_i \oplus B_i \oplus h(PW_i))$ at the login session, where $M_i = K_U \bmod k_i$. Thus, their scheme suffers high hash overhead if k_i is very large. In Step L-4, the server authenticates the legitimate user with verifying DID_i, and in Step L-6, the user authenticates the server by checking DID_S. That is, the server and the user obtain mutual authentication after the Step L-6 is completed. Thus, the Steps L-7 and L-8 are redundant in authentication. In the login Step L-3, the server computes $A_i = h(K_U) \oplus K_S$, where K_U is the common key of user for S. Since K_U is different for each user, the server has difficulty to find the correct common key for verification if the system has a huge amount of users.

Moreover, their scheme is vulnerable to the following attacks:

(1) It is vulnerable to the impersonation attack.

Suppose that an adversary Eve (*E*, for short) wants to impersonate as a legitimate user U_i to login the system. Firstly, Eve intercepts $CTID_i$ from Step L-2 and $CTID_S$ from Step L-5. Then, with Eq. (13), n_S can be obtained by $n_S = CTID_S \oplus CTID_i$.

Next, Eve intercepts DID_{iS} from Step L-7. Then, with Eq. (16), n_i also can be recovered by $n_i = DID_{iS} \oplus DID_S \oplus (n_S + 1)$. With n_i, the user's $h(g^{A_i})$ and TID_i will be obtained with Eqs. (9) and (10). By using TID_i and $h(g^{A_i})$, Eve can impersonate as the legitimate user U_i with the following steps.

Step I-1: $E \rightarrow S$: $\{DID_i, CTID_i, C_i, k_i\}$

Eve selects two integers for nonces n_i and k_i, and chooses a small integer for M_i. Thereby she computes $\{CTID_i, C_i, DID_i\}$ by $CTID_i = TID_i \oplus n_i$, $C_i = h(g^{A_i}) \oplus n_i$ and $DID_i = h^{M_i}(TID_i \oplus h(g^{A_i}))$. Next, Eve sends $\{DID_i, CTID_i, C_i, k_i\}$ along with the login request to *S*.

Step I-2: $S \rightarrow E$: $\{DID_S, CTID_S\}$.

After receiving $\{DID_i, CTID_i, C_i, k_i\}$, the server computes A_i and obtains $\{n_i, TID_i, M_i\}$ as the steps in login phase. Note that $M_i = K_U \bmod k_i$ and Eve doesn't know K_U, so correct M_i is also unknown by Eve. Thus the verification probably fails.

However, if Eve chooses a very small k_i such that M_i will be small enough, then the forwarded DID_i will pass the verification with a very high probability. That is, in Step I-1, Eve choose a very small k_i and selects another very small integer for M_i, where $M_i < k_i$. Then Eve will pass the verification with a very high probability of $P = 1/2^{|k_i|}$. If Eve is authenticated, the server generates a nonce n_S, computes $\{DID_S, CTID_S\}$ and sends it to Eve.

Step I-3: $E \rightarrow S$: $\{DID_{iS}\}$.

After receiving $\{DID_S, CTID_S\}$, Eve obtains n_S by Eq. (14). Then DIS_{iS} can be obtained by Eq. (16). Next, Eve sends DIS_{iS} to the server.

Step I-4: *S* and Eve obtain a common session key.

After receiving $\{DID_S, CTID_S\}$, *E* and U_i obtain a common session key.

Hereafter, the adversary can successfully impersonate as a legitimate user to communicate with the server by using the common session key. Thus, Shin *et al.*'s scheme is vulnerable to the impersonation attack.

(2) It cannot resist the off-line guessing attack.

Similar to the cryptanalysis steps in the impersonation attack, Eve obtains n_S by $n_S = CTID_S \oplus CTID_i$ and knows n_i by $n_i = DID_{iS} \oplus DID_S \oplus (n_S + 1)$. With n_i, the user's transform identity TID_i will be obtained by $TID_i = CTID_i \oplus n_i$. Since $TID_i = h(ID_i \| h(PW_i))$ and user's identity ID_i is public, the password PW_i can be easily guessed. Thus, Shin *et al.*'s scheme cannot resist the off-line guessing attack.

(3) It suffers the denial-of-service attack.

In Step U-3 of the password updating phase, if Eve sends a random message *X* to the server. The server will decrypted it to *Y* and replace the old transform identity TID_i

with Y, where $Y = D_{SK_S}(X)$. Hereafter, the legitimate user cannot successfully login the system for services since $Y \neq TID_i^*$. Thus, Shin *et al.*'s scheme cannot withstand the denial-of-service attack.

(4) It cannot withstand the stolen-verifier attack.

The server stores the transformed user's identity TID_i in the database. Since $TID_i = h(ID_i||h(PW_i))$ and ID_i is public, an adversary can easily guess the password PW_i if he/she obtains the verify table. With the password, the adversary can impersonate as the legitimate user to login the system if he/she obtains the smart card. Thus, Shin *et al.*'s scheme cannot withstand the stolen-verifier attack.

5. THE IMPROVED SMART CARD BASED REMOTE USER AUTHENTICATION SCHEME WITH ANONYMITY

In this section, we propose an improved scheme to fix the flaws. Suppose that the smart card is tamper-free, the stored information and the computation data in process should not be leaked out. The improved scheme also comprises registration phase, login phase, key agreement phase and password updating phase as follows.

5.1 Registration Phase

If a user wants to join the system, he/she should register himself/herself to the server. The server sends a smart card to the user after registration steps are completed.

Step IR-1: $U_i \Rightarrow S$: $\{ID_i, h(PW_i), r\}$.

Firstly, the user U_i chooses identity ID_i and password PW_i. Then he/she sends $\{ID_i, h(PW_i), r\}$ along with the registration request to the server S via a secure channel, where r is a random number.

Step IR-2: The server computes user's TID_i and A_i.

After receiving $\{ID_i, h(PW_i), r\}$, the server checks the uniqueness of the ID_i. The server asks the user to select a new identity if ID_i is already existed in the database. Then S computes U_i's transform identity TID_i and A_i by:

$$TID_i = h(ID_i||r||h(PW_i)) \tag{20}$$

$$A_i = h(K_S) \oplus h(PW_i) \tag{21}$$

Where K_S is the server's secret. The server stores TID_i in the database.

Step IR-3: $S \Rightarrow U$: Smart card.

The server installs $\{TID_i, h(), A_i, r\}$ into a smart card and sends it to the user.

5.1 Login Phase

If the user wants to acquire services from the server, he/she should log into the system. The login steps are as follows.

Step IL-1: $U_i \rightarrow S$: $\{CTID_i, B_i, C_i, T_i, R_i\}$.

The user attaches his/her smart card to a card reader and keys in ID_i and PW_i. The smart card computes $TID_i' = h(ID_i||r||h(PW_i))$ and checks whether $TID_i' = TID_i$ holds. If $TID_i' = TID_i$, the smart card obtains $h(K_S) = A_i \oplus h(PW_i)$ and computes $\{CTID_i, B_i, C_i\}$ as follows after generating two nonces n_i and R_i.

$$CTID_i = TID_i \oplus h(n_i) \tag{22}$$
$$B_i = h(R_i \oplus h(K_S)) \oplus n_i \tag{23}$$
$$C_i = h(TID_i \oplus h(K_S) \oplus T_i) \tag{24}$$

Where T_i is the current timestamp. Then the user sends $\{CTID_i, B_i, C_i, T_i, R_i\}$ along with the login request to the server.

Step IL-2: The server authenticates the user.

Upon receiving $\{CTID_i, B_i, C_i, T_i, R_i\}$, the server checks whether T_i is in a valid time interval to ensure the freshness. If T_i is not fresh, the server rejects the login request; otherwise, the server obtains n_i by:

$$n_i = B_i \oplus h(R_i \oplus h(K_S)). \tag{25}$$

Thereby the transform identity TID_i can be recovered by:

$$TID_i = CTID_i \oplus h(n_i). \tag{26}$$

The server checks whether the transform identity TID_i is in the database. If it isn't, the server terminates the login steps; otherwise, the server computes C_i' by:

$$C_i' = h(TID_i \oplus h(K_S) \oplus T_i). \tag{27}$$

The server checks whether C_i' is equal to C_i. If $C_i' = C_i$, the server will authenticate the user. Otherwise, the server rejects the user's login request.

Step IL-3: $S \Rightarrow U_i$: $\{D_i, T_S\}$.

After the user U_i is authenticated, the server computes D_i by:

$$D_i = h(TID_i \oplus n_i \oplus T_S). \tag{28}$$

Where T_S is the current timestamp. Next, the server sends $\{D_i, T_S\}$ to the user.

Step IL-4: The user authenticates the server.

After receiving $\{D_i, T_S\}$, the user computes $D_i' = h(TID_i \oplus n_i \oplus T_S)$. The server is authenticated if $D_i' = D_i$. Thereby, the mutual authentication between the server and the user is obtained.

5.2 Key Agreement Phase

After mutual authentication is obtained, the user's smart card and the server com-

pute the common session key SK_i and SK_S, respectively, by:

$$SK_i = h(h(K_S) \oplus TID_i \oplus n_i), \quad (29)$$
$$SK_S = h(h(K_S) \oplus TID_i \oplus n_i). \quad (30)$$

Hereafter, the user can use the common session key to communicate with the server securely.

5.3 Password Updating Phase

When the user wants to change his/her password, the steps are as follows.

Step IU-1: The smart card checks ID_i and PW_i.

The user keys in ID_i and PW_i to the smart card. The smart card computes $TID_i' = h(ID_i||r||h(PW_i))$ and compares it with the stored transform identity TID_i. The password updating steps continued if $TID_i' = TID_i$; otherwise, the server denies the password updating request.

Step IU-2: $U_i \rightarrow S$: $\{F, G, T_i'\}$

The user selects a new password PW_{i_new}. Then, the smart card obtains $h(K_S)$ as the Step IL-1 and computes TID_{i_new} and A_{i_new} by:

$$TID_{i\_new} = h(ID_i||r||h(PW_{i\_new})) \quad (31)$$
$$A_{i\_new} = A_i \oplus h(PW_i) \oplus h(PW_{i\_new}) \quad (32)$$

Next, U_i sends $\{F, G, T_i'\}$ to the server after computing F and G as follows, where T_i' is the current timestamp.

$$F = E_{SK_i}(TID_{i\_new}||h(K_s)) \quad (33)$$
$$G = h(TID_{i\_new} \oplus h(h(K_s) \oplus T_i')) \quad (34)$$

Step IU-3: $S \rightarrow U_i$: $\{H, T_S'\}$.

Upon receiving $\{F, G, T_i'\}$, the server checks the freshness of the timestamp T_i'. If T_i' is in a valid time interval, the server obtains TID_{i_new} and $h(K_S)$ by decrypting F with the session key SK_S. Then the server computes $G' = h(TID_{i\_new} \oplus h(h(K_S) \oplus T_i'))$. The server checks whether $G = G'$ holds. If $G = G'$, the server replaces TID_i with TID_{i_new}. Next, the server computes H by:

$$H = h(h(TID_{i\_new}) \oplus h(K_S) \oplus T_S'). \quad (35)$$

Where T_S' is the current timestamp. The server sends $\{H, T_S'\}$ to the user.

Step IU-4: The user updates TID_{i_new} and A_{i_new}.

Upon receiving $\{H, T_S'\}$, the user checks the freshness of the timestamp T_S'. If T_S' is in a valid time interval, the user computes $H' = h(h(TID_{i\_new}) \oplus h(K_S) \oplus T_S')$. If $H' = H$, the user replaces TID_i and A_i with TID_{i_new} and A_{i_new}, respectively.

6. DISCUSSIONS AND SECURITY ANALYSIS

The improved scheme provides dynamic identity, user anonymity, forward secrecy and mutual authentication. The computation overhead is quite low. Moreover, the scheme can withstand variety of attacks such as the replay attack, off-line guessing attack, smart card loss attack, impersonation attack and insider attack. The merits of the improved scheme are described as follows.

(1) The proposed scheme obtains dynamic identity and user anonymity.

In the login phase, the user sends $\{CTID_i, B_i, C_i, T_i, R_i\}$ to the server, where $CTID_i = TID_i \oplus h(n_i)$, $B_i = h(R_i \oplus h(K_S) \oplus n_i$, $C_i = h(TID_i \oplus h(K_S) \oplus T_i)$. Due to the nonce n_i and timestamp T_i are varied, thus $\{CTID_i, B_i, C_i\}$ are always changed on each login session. That is, in the login phase, the user sends dynamic information $\{CTID_i, B_i, C_i\}$ rather than fixed identity to the server. If an adversary wants to obtain TID_i from Eq. (22), it is infeasable due to n_i is unknown. The dynamic identity property also ensures the user anonymity.

(2) The proposed scheme provides forward and backward secrecy.

Forward secrecy means that a compromise of the current key should not compromise any future key. While backward secrecy implies any earlier key cannot be revealed even if the current key is disclosed. In the proposed scheme, the common session key is computed by $h(h(K_S) \oplus TID_i \oplus n_i)$, it is infeasible to obtain the session key since $h(K_S)$, TID_i and n_i are unknown. If an adversary wants to obtain $\{ h(K_S), TID_i, n_i\}$ from intercepted information $\{CTID_i, B_i, C_i, T_i, R_i\}$ in Step IL-1, it is infeasible due to the irreversible one-way hash function is adopted. The session key is always changed since n_i is updated on each login session. The adversary cannot know any future or earlier session key even if he/she obtains the current session key. Therefore, the proposed scheme provides forward and backward secrecy.

(3) The proposed scheme obtains mutual authentication.

The mutual authentication is essential for many authentication schemes. In Step IL-2, upon receiving $\{CTID_i, B_i, C_i, T_i, R_i\}$, the server authenticates the user by checking C_i. On the other hand, in Step IL-4, the user authenticates the server if D_i is verified. Thus, the proposed scheme provides mutual authentication for the user and server.

(4) The computation overhead is very low.

For the widely used cheap smart card, the computation complexity and memory space should be considered on system implementation. In Shin *et al.*'s scheme, the exponential operation is required in the registration phase and login phase, thus the computation overhead is quite high. Moreover, since the multi-hash function $h^{M_i}(\cdot)$ is required in their scheme, the computation complexity is also very high if M_i is large.

In the proposed improved scheme, only bitwise exclusive-or (XOR) operation and one-way hash function are adopted for computation. Thus, the computation overhead is quite low.

(5) It withstands the replay attack.

In the login phase, the user sends $\{CTID_i, B_i, C_i, T_i, R_i\}$ to the server for authentication. Because of $C_i = h(TID_i \oplus h(K_S) \oplus T_i)$ which comprises the timestamp, the replay message will be detected if the adversary replays the intercepted message. Similarly, if an adversary replays the message $\{D_i, T_S\}$ to the user in Step IL-3, the user also will detect the attack by checking the freshness of the timestamp. So the proposed scheme resists the replay attack.

(6) It resists the off-line guessing attack.

Suppose that an adversary wants to guess the password by using the intercepted information $\{CTID_i, B_i, C_i, T_i, R_i\}$ in Step IL-1. We discuss the attack with the following three cases: (1) If the adversary wants to guess the password by using $CTID_i$, due to $CTID_i = (h(ID_i)||h(PW_i)) \oplus h(n_i)$ and n_i is unknown, the attack fails since the adversary cannot verify his/her guessing; (2) If the adversary intends to obtain n_i with B_i, the intention will fail due to $B_i = h(R_i \oplus h(K_S) \oplus n_i)$ and K_S is unknown; (3) If the adversary wants to guess the password by using C_i, due to $C_i = h(TID_i \oplus h(K_S) \oplus T_i)$ which is equal to $h((h(ID_i)||r||h(PW_i)) \oplus h(K_S) \oplus T_i)$, the attack also will fail since $h(K_S)$ and r are unknown.

If the adversary wants to obtain the password by the information $\{D_i, T_S\}$ intercepted in Step IL-3, the attack also will fail due to $D_i = h(TID_i \oplus n_i \oplus T_S)$ and n_i is unknown. Thus, the improved scheme can resist the off-line guessing attack.

(7) It resists the smart card loss attack.

Smart card loss attack means an attacker can launch various attacks such as the off-line guessing attack if he/she obtains a legitimate user's smart card. In the login phase, the user's smart card compute B_i and C_i by using two random numbers n_i and R_i, thus the forward message $\{CTID_i, B_i, C_i, T_i, R_i\}$ is updated on each login session. If the adversary obtains the smart card and trying to guess the password, the try will fail due to the ever changed information makes the adversary cannot verify his/her guessing. Thus, the smart card loss attack can be avoided even if the adversary obtains the smart card. Moreover, the improved schemes can also limit the number of unsuccessful attempts of password guessing to avoid the attack.

(8) It resists the impersonation attack.

Suppose that an adversary wants to impersonate as a user to login the system. The attempt will fail due to $\{n_i, h(K_S), TID_i\}$ is unknown by the adversary such that the correct login information $\{CTID_i, B_i, C_i, T_i, R_i\}$ cannot be obtained. If the adversary wants to recover n_i, $k(K_S)$ and TID_i with the intercepted message $\{CTID_i, B_i, C_i, T_i, R_i\}$, it is infeasible due to the one-way hash function is adopted. Similarly, if the adversary wants to masquerade as the server, the attempt also will fail due to correct D_i cannot be obtained without knowing n_i and TID_i. Thus, the improved scheme can resist the impersonation attack.

(9) It resists the insider attack.

A malicious insider is a legitimate user whose actions are counter to the system policy, or is a masquerader who owns a legitimate user's identity and impersonates as another user for malicious purposes. In the improved scheme, the secret parameters TID_i and A_i are stored in the tamper-free smart card, any malicious insider cannot obtain the secrets even if he/she owns the card. It is infeasible to find the information $h(K_S)$ and

TID_i with $\{CTID_i, B_i, C_i, T_i, R_i\}$ or $\{D_i, T_S\}$ due to irreversible one-way hash function. That is a malicious insider cannot compute correct messages to impersonate as a legitimate user to login the system. Therefore, the scheme can resist the insider attack.

7. CONCLUSIONS

Recently, Shin *et al.* proposed a remote authentication scheme. In this article, we first show that their scheme is vulnerable to the impersonation attack, denial-of-service attack, off-line guessing attack and stolen-verifier attack. Their scheme also has drawbacks such as high hash overhead and validations steps redundancy. Then we propose an improvement scheme to fix the flaws. The improved scheme has the merits as follows:

(1) Providing dynamic identity and user anonymity.
(2) Obtaining forward/backward secrecy and mutual authentication.
(3) Low computation overhead.
(4) Withstanding the replay attack, off-line guessing attack, smart card loss attack, impersonation attack and insider attack.

REFERENCES

1. C. S. Bindu, P. C. S. Reddy, and B. Satyanarayana, "Improved remote user authentication scheme preserving user anonymity," *International Journal of Computer Science and Network Security*, Vol. 8, 2008, pp. 62-65.
2. C. C. Chang and T. C. Wu, "Remote password authentication with smart cards," *IEE Proceedings*-E138, Vol. 3, 1993, pp. 65-168.
3. H. Y. Chien and C. H. Chen, "A remote authentication scheme preserving user anonymity," in *Proceedings of the 19th International Conference on Advanced Information Networking and Applications*, Vol. 2, 2005, pp. 245-248.
4. H. Y. Chien, J. K. Jan, and Y. M. Tseng, "An efficient and practical solution to remote authentication: smart card," *Computer and Security*, Vol. 4, 2002, pp. 372-375.
5. M. L. Das, A. Saxena, and V. P. Gulati, "A dynamic ID-based remote user authentication scheme," *IEEE Transactions on Consumer Electronics*, Vol. 50, 2004, pp. 629-631.
6. B. T. Hsieh, H. M. Sun, and T. Hwang, "On the security of some password authentication protocols," *Informatica*, Vol. 14, 2003, pp. 195-204.
7. M. S. Hwang and L. H. Li, "A new remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, Vol. 46, 2000, pp. 28-30.
8. W. S. Juang, "Efficient password authentication key agreement using smart cards," *Computer and Security*, Vol. 23, 2004, pp. 167-173.
9. M. K. Khan, S. K. Kim, and K. Alghathbar, "Cryptanalysis and security enhancement of a more efficient and secure dynamic ID-based remote user authentication schemes," *Computer Communications*, Vol. 34, 2011, pp. 306-309.
10. W. C. Ku and S. M. Chen, "Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, Vol. 50, 2004, pp. 204-207.

11. M. Kumar, "A new secure remote user authentication scheme with smart cards," *International Journal of Network Security*, Vol. 11, 2010, pp. 88-93.
12. L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, Vol. 24, 1981, pp. 770-772.
13. C. T. Li and C. C. Lee, "A robust remote user authentication scheme using smart card," *Information Technology and Control*, Vol. 40, 2011, pp. 231-238.
14. C. H. Liao, H. C. Chen, and C. T. Wang, "An exquisite mutual authentication schemes with key agreement using smart card," *Informatica*, Vol. 33, 2009, pp. 125-132.
15. Y. P. Liao and S. S. Wang, "A secure dynamic ID based remote user authentication scheme for multi-server environment," *Computer Standards and Interfaces*, Vol. 31, 2009, pp. 24-29.
16. C. W. Lin, C. S. Tsai, and M. S. Hwang, "A new strong password authentication scheme using one-way hash functions," *International Journal of Computer and Systems Sciences*, Vol. 45, 2006, pp. 623-626.
17. S. Shin, K. Kim, K. H. Kim, and H. Yeh, "A remote user authentication scheme with anonymity for mobile devices," *International Journal of Advanced Robotic Systems*, Vol. 9, 2012, pp. 1-7.
18. H. M. Sun, "An efficient remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, Vol. 46, 2000, pp. 958-961.
19. H. R. Tseng, R. H. Jan, and W. Yang, "A bilateral remote user authentication scheme that preserves user anonymity," *Journal of Security and Communication Networks*, Vol. 1, 2008, pp. 301-308.
20. C. S. Tsai, C. C. Lee, and M. S. Hwang, "Password authentication schemes: current status and key issues," *International Journal of Network Security*, Vol. 3, 2006, pp. 101-115.
21. D. Wang, C. G. Ma, Q. M. Zhang, and S. Zhao, "Secure password-based remote user authentication scheme against smart card security breach," *Journal of Networks*, Vol. 8, 2013, pp. 148-155.
22. S. Wu, Y. Zhu, and Q. Pu, "Robust smart-cards-based user authentication scheme with user anonymity," *Security and Communication Networks*, Vol. 5, 2012, pp. 236-248.
23. E. J. Yoon, E. K. Ryu, and K. Y. Yoo, "Further improvement of an efficient password based remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, Vol. 50, 2004, pp. 612-614.

Yung-Cheng Lee received his Ph.D. degree at the Department of Electrical Engineering, National Cheng Kung University, Taiwan, in 1999. He was the Secretary of Chinese Cryptology and Information Security Association, the Chairman of the Department of Electrical Engineering and Department of Computer Science and Information Engineering, National Formosa University; and also was the Dean of Security and Engineering School, WuFeng University, Taiwan. Currently, he is the Dean of the Academic Affair, WuFeng University. His current research interests include network security, artificial intelligence and cryptography.