A Remote User Authentication Scheme with Anonymity for Mobile Devices

Soobok Shin 1,* Kangseok Kim 2, Ki-Hyung Kim 3 and Hongjin Yeh 1

1 Graduate school of Information and Communication at Ajou University, Suwon, Korea
2 Department of Knowledge Information Security at Ajou University, Suwon, Korea
3 Department of Information and Computer Engineering at Ajou University, Suwon, Korea
* Corresponding author E-mail: watermel@ajou.ac.kr

Received 18 Jan 2012; Accepted 09 Feb 2012

© 2012 Shin et al.; licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/3.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract With the rapid growth of information technologies, mobile devices have been utilized in a variety of services such as e-commerce. When a remote server provides such e-commerce services to a user, it must verify the legitimacy of the user over an insecure communication channel. Therefore, remote user authentication has been widely deployed to verify the legitimacy of remote user login requests using mobile devices like smart cards. In this paper we propose a smart card-based authentication scheme that provides both user anonymity and mutual authentication between a remote server and a user. The proposed authentication scheme is a simple and efficient system applicable to the limited resource and low computing performance of the smart card.

The proposed scheme provides not only resilience to potential attacks in the smart card-based authentication scheme, but also secure authentication functions. A smart card performs a simple one-way hash function, the operations of exclusive-or and concatenation in the authentication phase of the proposed scheme. The proposed scheme also provides user anonymity using a dynamic identity and key agreement, and secure password change.

Keywords Authentication Scheme, User Anonymity, Mobile Device, Smart Card

1. Introduction

The main aim of the remote authentication scheme using smart cards is to identify and verify the smart card holder with valid access rights and access to the remote server. It has been widely accepted that the smart card-based remote user authentication is one of the most reliable and secure forms of electronic identification for authentication. Therefore, a variety of password-based authentication schemes have been proposed for remote authentication using smart cards. In a traditional remote authentication scheme, a user has to register her own identity and password to the server in advance, and she has to submit the identity and password information to the server during a login process. On receiving a login request, the remote server authorizes the user to access facilities provided by the remote server, if the pair of identity and password is equivalent to the one stored in the server’s password table. Otherwise, the access request is rejected.
The desirable security requirements of an authentication scheme using smart cards are as follows. The scheme should resist malicious insider, replay, guess, stolen-verifier and impersonation attacks. Also, the scheme should provide both forward secrecy and known-key security, guaranteeing user anonymity.

The desirable functionality requirements of an authentication scheme are as follows. A user chooses her identity and password freely and changes her password securely, and a remote server does not maintain a verification table to authenticate the user in the server. Due to the power constraints of smart cards, the computational cost of the scheme has to be low and the scheme should provide mutual authentication, user anonymity and session key agreement between a user and a server without requesting time-synchronization between user and server.

In this paper we propose an enhanced authentication scheme using smart cards. Our scheme satisfies the security requirements and functions.

The remainder of this paper is organized as follows: section 2 reviews related works; section 3 details the proposed authentication scheme; section 4 analyses its security; section 5 analyses its performance and functionality. Finally, Section 6 draws brief conclusions.

2. Related Works

A number of remote authentication schemes have been suggested by researchers from time to time. When a server authenticates a user, the server verifies the user with the entered user’s identity and password, and the corresponding values in the verification table of the server. In 1981, Lamport [1] presented a remote authentication scheme using password tables, but security holes and maintenance responsibilities for the verification table exist because it must always maintain the verification table. Also, a stolen verification table may cause many security threats, therefore, in 1990, Hwang [2] presented a remote authentication scheme without password tables. Many other studies have also proposed a scheme that does not maintain the verification table to authenticate the remote user.

Numerous studies [7-16] have proposed a scheme using the one-way hash function and exclusive-or operation due to the constrained resources and computing power of a smart card.

After the user’s authentication is completed, the communication channel for a message may be insecure. If the message is not encrypted, then the message is revealed to an adversary. Therefore, previous studies [11-14, 16] have proposed session key generation schemes using key agreement between user and server, and researchers have also pointed to the weaknesses of other schemes [3, 4].

Because the user’s anonymity is very important in many e-commerce applications, several schemes [5, 6, 9, 12-13, 15, 16] have been proposed to achieve user anonymity in the authentication phase.

<table>
<thead>
<tr>
<th>Notations</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(U_i)</td>
<td>User (i)</td>
</tr>
<tr>
<td>(PW_i)</td>
<td>The password of (U_i)</td>
</tr>
<tr>
<td>(S)</td>
<td>The remote server</td>
</tr>
<tr>
<td>(K_s)</td>
<td>The secret key of server</td>
</tr>
<tr>
<td>(K_u)</td>
<td>The common key of user for (S)</td>
</tr>
<tr>
<td>(TID_i)</td>
<td>The transformed identity of (U_i)</td>
</tr>
<tr>
<td>(CTID_i)</td>
<td>The changed identity of (U_i)</td>
</tr>
<tr>
<td>(DID_i)</td>
<td>The dynamic identity of (U_i)</td>
</tr>
<tr>
<td>(DID_s)</td>
<td>The dynamic identity of (S)</td>
</tr>
<tr>
<td>(SK_i)</td>
<td>The generated session key of (U_i)</td>
</tr>
<tr>
<td>(SK_s)</td>
<td>The generated session key of (S)</td>
</tr>
<tr>
<td>(h(\cdot))</td>
<td>A one-way hash function</td>
</tr>
<tr>
<td>(h(\cdot)^k)</td>
<td>Perform hash function of (k) times</td>
</tr>
<tr>
<td>(\oplus)</td>
<td>Bitwise exclusive-or operation</td>
</tr>
<tr>
<td>(|)</td>
<td>The string concatenation</td>
</tr>
<tr>
<td>(A \rightarrow B : M)</td>
<td>A sends M to B through a secure channel</td>
</tr>
<tr>
<td>(A \rightarrow B : M)</td>
<td>A sends M to B through a common channel</td>
</tr>
<tr>
<td>(E_{SK_i}[M])</td>
<td>Encrypted message by the session key, (SK_i)</td>
</tr>
</tbody>
</table>

Table 1. The notations used in the proposed scheme.

Bindu et al. [6] showed the possibility of insider attack, Main-in-Middle attack, in the scheme of Chien et al. [5] and presented an improved scheme preserving user anonymity. However, the scheme does not provide a password change phase and uses time-stamp to resist replay attack. Lin et al. [8] presented a new strong-password authentication protocol that can withstand a stolen-verifier attack and other possible attacks, but the scheme cannot change passwords and does not provide mutual authentication, session key agreement and user anonymity. Juang [10] presents a simple authentication scheme, but the scheme cannot change passwords and does not provide mutual authentication. Recently, Khan et al. [13] and Tseng et al. [16] presented authentication schemes providing user anonymity and mutual authentication. With the scheme proposed by Tseng et al. [16] the user can freely choose the password and securely change the password, however, both schemes require time synchronization to protect from replay attack.

Here, we point to security issues related to Das et al.’s scheme [15] and Liao et al.’s scheme [14].
1) Both schemes submit a user password directly to the server in the registration phase. Thus, their schemes are vulnerable to insider attacks. If a malicious insider obtains a user password in the registration phase, she may be able to access the other server over a network, because generally, a user is apt to use the same password for convenience, even for most other servers.

2) Neither scheme provides user anonymity. Although Das et al.’s scheme uses a dynamic identity in the verification phase as the user login in the remote server, she always sends the user specific value, $N_i$. Therefore, the user’s location is revealed to the adversary. In Liao et al.’s scheme, a user sends her identity directly via a common channel. Thus, an adversary can know the user’s location.

3) Neither scheme provides secure password change as their password change phase is insecure. When a user inputs the wrong password by mistake, the smart card’s password is changed to the wrong password.

4) Das et al.’s scheme does not provide mutual authentication. The server can authenticate a user, but a user cannot authenticate the remote server.

5) Das et al.’s scheme does not have a session key agreement. After mutual authentication between user and server, important messages have to be protected from adversaries, thus, session key agreement is needed. However, Das et al.’s scheme does not provide session key agreement.

6) Das et al.’s scheme requires time synchronization and uses time-stamp for replay attack. Thus, it requires time synchronization between user and server.

3. Proposed Authentication Scheme

In this section, we propose an enhanced security scheme for mutual authentication and user anonymity using a smart card. The proposed scheme overcomes the weaknesses of Das et al.’s scheme and Liao et al.’s scheme, while it enhances security compared to the existing schemes. The proposed scheme is composed of four phases: registration, login and authentication, key agreement and secure password update. Table 1 is the notation used in the proposed scheme.

3.1 Registration Phase

When user $U_i$ wants to access a remote server for a service legitimately, $U_i$ should perform the following registration steps before the access. The procedure is as follows:

1. $U_i \Rightarrow S$: $ID_i$, $h(PW_i)$ $U_i$ chooses her identity, $ID_i$, and password, $PW_i$, for registration and submits $ID_i$ and $h(PW_i)$ hashed value of $PW_i$, to the remote server via a secure communication channel. Both $ID_i$ and $PW_i$ are selected by the user freely.

Step 2. After receiving $ID_i$ and $h(PW_i)$ from $U_i$, the remote server, $S$ performs the following steps:

1. $S$ generates transform identity $TID_i = h(ID_i \| h(PW_i))$, and checks the existence of the transformed identity in the database. If the identity already exists in the database, $S$ requests $U_i$ to re-initiate the registration procedure with a different $ID_i$ or $PW_i$. Otherwise, $S$ stores $TID_i$ in the database. This process ensures the uniqueness of the user’s transformed identity.

2. Compute $A_i = (h(K_s) \oplus K_s)$, where $K_s$ is a secret key of $S$ and $K_s$ is a common key of user for $S$. $K_s$ is used to generate a dynamic identity, $DID_i$ in the login and authentication phase.

3. Compute $B_i = (q^h \mod p) \oplus h(PW_i)$, where $g$ is a primitive element in Galoisfield $GF(p)$ and $p$ is a large prime positive integer.

4. Store the values, $DID_i$, $B_i$, $h(\cdot)$ and $K_s$ in a smart card and issue the smart card to $U_i$.

![Figure 1. Registration phase.](image)

3.2 Login and Authentication Phase

After $U_i$ registers to $S$, when $U_i$ wants to log into the server, $U_i$ will send a login message to $S$. The login message contains a dynamic identity, $DID_i$ to guarantee user anonymity. After successful verification of the login message, $U_i$ can authenticate $S$ and $S$ can authenticate $U_i$. That is, our scheme provides mutual authentication. The login and the authentication phases work as follows:

Login phase: $U_i \rightarrow S$: $DID_i$, $CTID_i$, $C_i$, $k_i$ User, $U_i$ connects her smart card to a reader. She inputs her identity, $ID_i$ and password, $PW$. The smart card performs the following processes:

1. Generate nonces, $n_i$ and $k_i$.
2. Compute $CTID_i = TID_i \oplus n_i$.
3. Compute $C_i = h(B_i \oplus h(PW_i)) \oplus n_i$.
4. Compute $M_i = K_s \mod k_i$.
5. Compute $DID_i = h^{k_i}(TID_i \oplus B_i \oplus h(PW_i))$.
6. $U_i$ sends $DID_i$, $CTID_i$, $C_i$ and $k_i$ with the login request message to $S$. 
Authentication phase:
The proposed scheme provides mutual authentication. \( U_i \) and \( S \) perform the following processes to achieve this, after \( U_i \) sends the request message to \( S \).

Step 1. \( S \rightarrow U_i : DID_i, C_{TID_i} \).

\( S \) does the following processes to authenticate \( U_i \):
1. Compute \( A_i = h(K_{iS} \oplus K_s) \).
2. Compute \( g^s \mod p \) and execute the hash operation for this value, \( h(g^s \mod p) \).
3. Execute exclusive-or operation the received \( C_i \) and the value, \( h(g^s \mod p) \) for the value \( n_i = C_i \oplus h(g^s \mod p) \).
4. To compute \( TID_i \), \( S \) executes exclusive-or operation with the received value, \( C_{TID_i} \) and the generated value \( n_i \). \( TID_i = CTID_i \oplus n_i \). Then, \( S \) checks that \( TID_i \) is the registered transform identity in the database. If the value is not valid, terminate the connection, otherwise, continue the process.
5. Compute \( M_i = K_s \mod k_i \).
6. \( S \) computes \( DID_i = h^{TID_i}(CTID_i \oplus h(g^s \mod p)) \) and compares the received value, \( DID_i \) and the generated value, \( DID_i \) by the server. If \( DID_i = DID_i \), \( S \) authenticates the legitimate user, \( U_i \). Otherwise, \( S \) fails authentication of \( U_i \) and \( S \) terminates the connection with \( U_i \).
7. Generate nonce, \( n_i \).
8. Compute \( DID_i = h(DID_i \oplus n_i) \) and \( CTID_i = CTID_i \oplus n_i \).
9. \( S \) sends \( DID_i \) and \( CTID_i \) to \( U_i \).

Step 2. \( U_i \rightarrow S : DID_i \).

\( U_i \) authenticates \( S \) and mutual authentication is completed according to the following processes:
1. Compute \( n_i = CTID_i \oplus CTID_i \).
2. \( U_i \) computes \( DID_i = h(DID_i \oplus n_i) \) and compares the received value, \( DID_i \) and the generated value, \( DID_i \). If \( DID_i = DID_i \), the user, \( U_i \) authenticates the remote server, \( S \). Otherwise, \( U_i \) fails server authentication and terminates connection with \( S \).
3. \( U_i \) computes the value, \( DID_i = DID_i \oplus n_i \oplus (n_i + 1) \) and sends \( DID_i \) to \( S \).
4. \( S \) computes \( (n_i + 1) = DID_i \oplus n_i \oplus (n_i + 1) \) and compares the value, \( (n_i + 1) \) and the generated value, \( (n_i + 1) \). If \( (n_i + 1) = (n_i + 1) \), mutual authentication is complete. Otherwise, \( S \) terminates connection with \( U_i \).

3.3 Key Agreement Phase

After \( U_i \) and \( S \) bring mutual authentication to completion, they generate the session keys for secure transmission of messages to each other. Then the session key is generated using information (values) in the authentication phase.

Step 1. User, \( U_i \) generates session key, \( SK_i \). \( U_i \) computes \( SK_i = h(B_i \oplus h(PW_i) \oplus n_i) \). \( SK_i \) is the session key generated by the user. Then, values, \( B_i, h(PW_i) \), \( n_i \) are not revealed in transmission via a common channel.

Step 2. Remote server, \( S \), generates session key, \( SK_r \). \( S \) computes \( SK_r = h(g^s \mod p) \oplus n_i \). \( SK_r \) is the session key generated by the remote server. In addition, values \( g^s \mod p, n_i \) and \( n_i \) are not revealed in transmission via a common channel.

\( SK_i \) and \( SK_r \) are the same values, since \( (g^s \mod p) = B_i \oplus h(PW_i) \). Thus, the session key is created safely between the user, \( U_i \) and the remote server, \( S \).

3.4 Secure Password Update Phase

When user, \( U_i \) wants to change her password for personal reasons or for the sake of security, \( U_i \) can change her password freely. The proposed scheme provides secure password change. The procedure is as follows:
1. \( U_i \rightarrow S : DID_i, C_{TID_i}, C_i, K_s, M_{request-change-PW} \).
   \( U_i \) inserts the smart card into a reader and sends \( DID_i, C_{TID_i}, C_i \) and \( K_s \) with the request message, \( M_{request-change-PW} \) to \( S \).
2. Mutual authentication is performed between \( U_i \) and \( S \), as in the login and authentication phase, mentioned earlier.
3. \( U_i \) generates new password, \( PW_i \) and computes \( TID_i = h(ID_i \oplus h(PW_i)) \).
4. \( U_i \rightarrow S : E_{SK_i}(TID_i) \).
   \( U_i \) encrypts new transform identity \( TID_i \) using session key, \( SK_i \) and sends the encrypted message to the server.
5. \( S \) decrypts the received message using \( SK_i \) and then replaces the value, \( TID_i \) with the received value \( TID_i \). \( S \) sends the response message to \( U_i \).
6. After receiving the response message from \( S \), \( U_i \) computes \( B_i' = B_i \oplus h(PW_i) \oplus h(PW_i) \) and replaces stored values in the smart card, \( TID_i \) and \( B_i \) with \( TID_i \) and \( B_i' \) with each other.

Figure 2. Login and authentication phase.

Figure 3. Secure password update phase.
4. Security Analysis

In this section, we analyse the security of the proposed scheme. The proposed scheme can resist insider, replay, guessing, stolen-verifier and impersonation attacks, and provide user anonymity, forward secrecy, known-key security and mutual authentication for enhanced security.

4.1 The proposed scheme can resist an insider attack

In the registration phase, \( U_i \) submits her identity, \( ID_i \) and the hashed value of password, \( h(PW_i) \) instead of \( PW_i \) for remote-access services. Due to the employment of the one-way hash function \( h() \), it is impossible for an insider to derive the user’s password \( PW_i \) from the hashed value, \( h(PW_i) \). That is, even the server does not know \( PW_i \). Therefore, the proposed scheme can prevent insider attack.

4.2 The proposed scheme can resist a replay attack

Assume that an adversary eavesdrops on the login message sent by \( U_i \) when logging into the server in a later session. However, the replay of \( U_i \)’s previous login message will be detected by the server. In Step 2 of login and authentication, the adversary computes the value \( DIF = DIF \oplus n_i \oplus (n+1) \) for mutual authentication and the value is sent to the server. The adversary generates \( n_i \) and computes \( DIF = DIF \oplus n_i \oplus (n+1) \), because she does not know the value, \( n_i \) and sends \( DIF \) to the server. The server will derive the value \( (n+1) \) from the value \( DIF \) sent by the adversary. However, it is \( (n+1) \neq (n+1) \), since \( DIF \oplus DIF \neq DIF \oplus DIF \). Therefore, the adversary cannot launch a replay attack.

4.3 The proposed scheme can resist a guessing attack

Suppose an adversary finds out the identity and the password of a legitimate user by guessing. She can compute valid values, \( TID_i = h(ID_i \oplus PW_i) \) and \( CTID_i = TID_i \oplus n_i \). However, she does not know \( B_i \) and \( k_i \). Thus, she cannot compute valid values, \( C_i \) and \( DID_i \). Hence, the adversary cannot generate a valid login message.

4.4 The proposed scheme can resist a stolen-verifier attack

The server stores only the transformed user’s identity in the database and does not store the user’s other secret information corresponding to her transformed identity in the proposed scheme. Thus, only the malicious insider or intruder gets the table of the user’s transformed identity. Hence, the adversary cannot launch a stolen-verifier attack.

4.5 The proposed scheme can resist impersonation attack

If an adversary wants to impersonate \( U_i \), she has to create a valid login message: \( DID_i, CTID_i, C_i \) and \( k_i \). First, she has to choose a nonce \( n_i \) and compute \( CTID_{i^{'}} = TID_i \oplus n_i \), \( C_i = h(B_i \oplus h(PW_i)) \oplus n_i \), \( M_i = y \mod k_i \), \( DID_{i^{'}} = h^{M_i}(TID_i \oplus B_i \oplus h(PW_i)) \). Next, she sends the login message \( DID_i, CTID_i, C_i, k_i \) to the server. The adversary cannot forge a valid login message as she has no idea about \( B_i, PW_i \) and \( y \). Hence, she cannot launch an impersonation attack.

4.6 The proposed scheme can provide user anonymity

Consider an adversary eavesdrops on the login message, \( DID_i, CTID_i, C_i \) and \( k_i \). Here, \( DID_i \) is the dynamic identity and as \( CTID_i \) and \( C_i \) are computed by nonce, \( n_i \) they are different in the login phase. In addition, the value \( k_i \) is not the same in each login phase. Thus, the login messages submitted to the server are different in the login sessions. Hence, it is difficult for the adversary to discover a user’s identity.

4.7 The proposed scheme can provide forward secrecy

Suppose, the long-term secret key material (e.g. server’s secret key \( k_i \) and user’s password \( PW_i \)) is revealed to an adversary. Although the adversary knows the secret key material, she cannot compromise the secrecy of the agreed keys in earlier runs because the session key is computed using the long-term secret key material and nonce \( h(PW_i) \) and \( h(B_i) \). Thus, if the adversary does not know the values \( h(PW_i) \) and \( h(B_i) \), she cannot derive the j-th session key. Hence, the proposed scheme provides forward secrecy.

4.8 The proposed scheme can provide known-key security

Suppose that in the j-th session, the session key \( SK_{j-th} \) is compromised by an adversary. Then the adversary cannot further compromise other secret keys or session keys because the session key \( SK_{j-th} \) (\( j \neq k \)) uses nonce \( n_{j-th} \) and \( n_{k-th} \). Hence, the proposed scheme can achieve known-key security.

4.9 The proposed scheme can provide mutual authentication

In the login and authentication phase, \( U_i \) and the server securely exchange a nonce of user, \( n_i \) and a nonce of server, \( n_s \) respectively. Thus, \( U_i \) generates session key \( SK_{i} = h(B_i \oplus h(PW_i)) \oplus n_i \) and the server generates session key \( SK_s = h(g_s \mod p) \oplus n_i \). \( g_s \mod p = B_i \oplus h(PW_i) \) and the values \( n_i \) and \( n_s \) are not revealed in a common channel. Thus, the values \( SK_{i} \) and \( SK_{s} \) are the same and are secure.

5. Performance and Functionality Analysis

The proposed scheme achieves mutual authentication using only one a-way hash function and bitwise exclusive-or operation in a smart card. We prefer to adopt modular exponentiation, a relatively expensive operation,
in the registration phase. However, it is performed only at the remote server. Thus, the proposed authentication in this paper is pertinent to using a practical smart card. In addition, it provides session key agreement and a secure password change. Table 2 compares performance.

<table>
<thead>
<tr>
<th>Registratio n</th>
<th>Login and Authentication</th>
<th>Key Agreement</th>
<th>Password Update</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our scheme</td>
<td>1C, 3H, 2A, 1M</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Das et al.[15]</td>
<td>2H, 1B</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Liao et al.[14]</td>
<td>1C, 1H, 2A, 1M</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Lin et al.[8]</td>
<td>3C, 3H, 1A</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Bindu et al.[6]</td>
<td>4H, 3B</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Juang[10]</td>
<td>1H, 1B</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Khan et al.[13]</td>
<td>2C, 2H, 1A</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Tseng et al.[16]</td>
<td>5H, 4B, 1A</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

C : Concatenation  
H : One-way hash function  
A : Bitwise exclusive-or  
M : Modular exponentiation  
E : Encryption  
D : Decryption  
A : Arithmetic operation, such as add, subtraction and absolute value.  
O : Comparison operation  
+: More

Table 2. Performance comparisons of authentication schemes.

We summarize the functionalities of our proposed scheme in this section. The crucial criteria in the user authentication scheme are listed below:

**F1. Freely chosen password**: in the registration phase, a user can choose her identity and password freely for remote-access services.

**F2. Secure password change**: the user can change her password when she wants to change her password for the sake of security. In our scheme, after the user and server are authenticated, the password change is securely accomplished. Then, the generated value $TID^*$ is encrypted by the session key and transmitted to the server.

**F3. No verification table**: if the server maintains the verification table, when the verification table is revealed to an adversary, the overall authentication mechanism breaks down. Our scheme does not maintain the verification table with the user identity and corresponding password for user authentication. Only the server has a transformed identity table for user authentication.

**F4. Low computation**: computation overhead must be low in smart cards due to their constrained resources. Our scheme accomplishes mutual authentication merely by hash operation and bitwise exclusive-or operation.

**F5. Mutual authentication**: a malicious person can disguise herself as the server or can disguise herself as the user. However, our scheme can provide mutual authentication between user and server.

**F6. Session key agreement**: the user and the server communicate via a common channel after mutual authentication is accomplished. The session key agreement is provided for secure transmission of the important messages. The security of the session key is very important. Our scheme provides session key agreement and at the end of the key exchange, the session key is known to nobody but the user and the server.

**F7. Avoiding time synchronization**: our scheme adopts a nonce instead of using a time-stamp to prevent replay attacks and a synchronization problem. Thus, our scheme does not need time synchronization between user and server.

**F8. User anonymity**: our scheme uses dynamic identity for user anonymity. Whenever a user connects to the server for remote-access services, she sends a different identity. Thus, our scheme provides user anonymity.

Table 3 compares functionality. The proposed scheme satisfies the required functionalities.

<table>
<thead>
<tr>
<th></th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
<th>F7</th>
<th>F8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our scheme</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Das et al.[15]</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Liao et al.[14]</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Lin et al.[8]</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Bindu et al.[6]</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Juang[10]</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Khan et al.[13]</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Tseng et al.[16]</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

F1. Freely chosen password  
F2. Secure password change  
F3. No verification table  
F4. Low calculation for authentication  
F5. Mutual authentication  
F6. Session key agreement  
F7. Avoiding time synchronization  
F8. User anonymity

Table 3. Functionality comparisons of authentication schemes.

6. Conclusion

In this paper we proposed a security enhancement scheme of mutual authentication and user anonymity using smart cards. The proposed scheme does not send the user specific value in the login and authentication phase. Thus, it achieves user anonymity. The proposed scheme can resist insider, replay, guessing, stolen-verifier and impersonation attacks, and provides forward secrecy, known-key security and mutual authentication to enhance security. A user can freely choose her password...
and can change the password safely. In addition, our scheme provides the following functionalities: no verification table, avoiding time synchronization, eviction mechanism, session key agreement and low computation.

7. Acknowledgments

This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the Convergence-ITRC (Convergence Information Technology Research Center) support programme (NIPA-2011 C6150-1101-0004) supervised by the NIPA (National IT Industry Promotion Agency).

8. References