An Exquisite Mutual Authentication Scheme with Key Agreement Using Smart Card

Chiu-Hsiung Liao
General Education Center
National Chin-Yi University of Technology
Taichung, Taiwan 411, R.O.C.
E-mail: cliao@ncut.edu.tw

Hon-Chan Chen and Ching-Te Wang
Department of Information Management
National Chin-Yi University of Technology
Taichung, Taiwan 411, R.O.C.
E-mail: {chenhc, ctwang}@ncut.edu.tw

Keywords: Diffie-Hellman scheme, transformed identity, authentication, key agreement, password update

Received: June 25, 2008

To access a network system legally, efficiently and securely, the authentication scheme is essential and very important. In this paper, we propose a nonce-based authentication scheme using smart card. We use Diffie-Hellman scheme to enhance the security of our protocol. To lessen the computation load, the remote system alone proceeds the exponentiation computation and it also implements only once. The other computations are all concerned with simple one-way hash functions or exclusive-or operations. No verification table is needed in our protocol. The protocol provides not only mutual authentication between a user and the remote server but also achievement of key agreement. The protocol also supports convenient password update at the user's terminal. To avoid the identity duplication, we introduce the idea of transformed identity in our protocol.

Povzetek: Opisana je nova shema dostopa do omrežja s pomočjo pametne kartice.

1 Introduction

In recent years, people communicate via networks much more frequently than before. The frequency that network users transmit information and share the computing resources increases very quickly. Moreover, with e-commerce being prosperous, people use computers daily to link with server to ask for service. In these situations, the remote authentication and network security become inevitable and very important.

The authentication scheme is an essential part to assure legitimate, secure and efficient access to a network system. Among authentication schemes, password-based authentication is widely used. But password-based authentication is vulnerable to the dictionary attacks [1,2,3,4], i.e. the password guessing attacks, because people are inclined to choose easy-to-remember identities or meaningful phrases. As a result, a number of protocols have been proposed to overcome the guessing attacks [1,5–7]. Some of the improved protocols [1,8–12] use public key encryption in authentication. The others [6,11,12,14] use nonces and one-way hash functions. The nonce-based protocol is more secure because the nonce is randomly generated. As for one-way hash functions, it is irreversible. Thus, the protocol using hash functions and nonces is safe and secure.

Recently, some authentication protocols using smart card have been proposed [6,11,12,14]. Using smart card has many merits. Not only can it implement computations and store a wealth of useful information, like identification number, password and basic personal data, but also it is portable. Although the protocol using public key encryption is much more secure, it may incur a burdensome computation load. Therefore, we propose an authentication protocol using Diffie-Hellman scheme [15] to enhance the security level and efficiency but to reduce the computation load for a smart card. In our method, the smart card is responsible for simple computations and the server is responsible for complicated ones. The proposed scheme also uses the one-way hash function and the exclusive-or operation to maintain security and convenience. To prevent the replay attacks and the synchronization problem, we adopt the nonces in our scheme instead of using time-stamp. Furthermore, we introduce the design of transformed identity [16] in our scheme to avoid the duplication of identities.

The rest of this paper is organized as follows: Some related schemes are reviewed in Section 2. The proposed authentication scheme is described in Section 3. The security analysis of our scheme is discussed in Section 4. The efficiency and specialities of the proposed scheme are dis-

cussed in Section 5. The functionality and performance of the proposed scheme are compared with related schemes and the result is listed in Table 1. Finally, the conclusions are given in Section 6.

2 Reviews of related schemes

In this section, we review some related schemes briefly and closely.

2.1 Chien and Jan's scheme (ROSI scheme)

Chien and Jan proposed a nonce-based authentication scheme using smart card: Robust and Simple authentication protocol (ROSI) [6], in 2003. The ROSI scheme consists of two phases: "The registration phase" and "The authentication phase".

In the scheme, a prospective user, u, selects his identity, ID_u, password, PW_u, and an initial nonce, N_1. Then, the user transmits these values to the server, S, in registration phase. After accepting the application, the server stores ID_u and $h^2(PW_u||N_1)$ in its database, where the symbol " $||$ " is the string concatenation. The server also uses its secret key to calculate some parameters and stores them in a smart card. Then, the server issues the smart card to the applicant, u. After the authentication phase, the user and the server can mutually authenticate each other. However, in this scheme, it is necessary to set up a verification table and a legitimate user cannot update his password conveniently and freely when the security faces potential threats.

2.2 Juang's password authenticated key agreement scheme

In Juang's authenticated key agreement scheme using smart card [12], two phases are included: "The registration phase" and "The login and session key agreement phase".

A prospective user submits his identity and password to the server for registration. After getting a smart card, the user can use it to access the server. The user applies his smart card to compute a secret key and uses the key to encrypt a message, which includes a random value and an authentication tag. After receiving the message, the server computes the secret key and decrypts the received message to extract the embedded authentication tag. Then, the server verifies the validity of this tag. In order to attain the shared session key, the user's smart card has to encrypt a forwarding message and decrypt the received message from server to perform a nonce-checking. In this scheme, we found that the smart card should encrypt and decrypt several messages by using the cryptographic scheme. In this situation, the smart card has to compute the modular exponential operations, which require a large amount of computations. These computations may overload the capability of the smart card.

2.3 Hwang et al's remote user authentication scheme

The scheme [14] is comprised of three main phases and an additional one. The main phases are "The registration phase", "The log in phase" and "The authentication phase". The additional phase is "The password changing phase" within the user's discretion.

When a prospective user, u, wants to register with a server, S, he submits his identity, ID_u, and a hash value of password, $h(PW_u)$ to the registration center of S. Then, the center uses the server's secret key, x_s, and the hash value of password to compute a shifted password, $PW_{1u} = h(ID_u \oplus x_s) \oplus h(PW_u)$ and stores it with the hash function, $h(\cdot)$, into a smart card, where " $\oplus$ " is the exclusive-or operation. Then, the smart card is issued to the user.

To access the server, the user connects his smart card to a card reader and keys in his identity and password at the user's terminal. The smart card executes the exclusive-or operation on the shifted password and $h(PW_u)$ to attain a crucial parameter, $h(ID_u \oplus x_s)$. The smart card then combines this parameter with a time-stamp to compute an authenticating value. Next, the user transmits these values to the server for authentication. On receiving the messages, the server executes the verification procedures and performs the authentication. However, although the scheme can verify a legitimate user, the user and the server cannot achieve the mutual authentication and the session key agreement. The scheme cannot avoid the time synchronization problem, either.

2.4 Behind the reviews

In reviewing the related schemes, we are motivated to propose an improved scheme. Not only do we supplement the deficiencies, but we also enhance the efficiency and the functionality. In our scheme, the verification table is not required and the mutual authentication can be achieved. Furthermore, a user is allowed to select and update his password freely. Finally, the computation cost is reduced in the proposed scheme.

3 The proposed authentication scheme

Our authentication scheme consists of four phases: the registration phase, the login and authentication phase, the key agreement phase and the password update phase. As mentioned before, for the sake of security, we prefer to adopt modular exponentiation in registration phase. But, it is performed only at the remote server to reduce the computation load for smart card. The login phase is executed at the user's terminal and the authentication is verified mutually between the user and the server. The key agreement is achieved by the user and the server respectively, and is kept temporarily for mutual communication in the session. As

for the password update phase, it is completed only at the user's terminal.

To describe our proposed scheme with ease, we use the following symbols and operations:

1. The operator " $\oplus$ " is the bit-wise exclusive-or operation.
2. The symbol " $||$ " is the string concatenation.
3. The function " h " is a one-way hash function.
4. For the sake of convenience, let the expression " $X \longrightarrow Y : M$ " mean a sender X transmits a message M to a recipient Y.

3.1 The registration phase

The registration phase is performed with the remote server. When a person, u, wants to be a legitimate user to a server, S, he offers an account application to S. The procedure is as follows:

Step 1: $u \longrightarrow S : ID_u, PW_u$. Responding the challenge from the server, the applicant submits his identity, ID_u, and password, PW_u, to the server for registration via a secure communication channel. Both ID_u and PW_u are selected by himself freely.

Step 2: After receiving the response, the server confirms the formats of the submitted identity and password first. Then, the server takes note of the registration time, TS_u and archives the user's ID_u and related TS_u for later authenticating use. Then the server performs the following four processes:

(1) Compute the transformed identity [16], $TID_u = TS_u||ID_u$, automatically by itself. The transformed identity, TID_u, can ensure the uniqueness of the identity. At this stage, the applicant only needs to remember his selected identity, ID_u, and password, PW_u.

(2) Compute $A_u = h(TID_u \oplus x)$, where the parameter x is the secret key of S and is kept confidentially.

(3) Compute $B_u = (g^{A_u} \, mod \, p) \oplus PW_u$, where p is a large prime positive integer and g is a primitive element in Galois field $GF(p)$.

(4) Store the values, TS_u, B_u and $h(\cdot)$, in a smart card and issue the smart card to the applicant.

3.2 The login and authentication phase

When a legitimate user, u, intends to login the server, S, the user's terminal and the server need to mutually authenticate each other.

Step 1: $u \longrightarrow S : M_1 = \{ID_u, NTID_u, C_u\}$.
The user, u, connects his smart card to a reader. The smart card challenges the user for his identity, ID_u, and password, PW_u, which are selected at his application. The smart card automatically performs the following processes:

(1) Generate a nonce, n_u. Store the value, n_u, temporarily until the end of the session.

(2) Retrieve the stored registration time to generate the transformed identity, $TID_u = TS_u||ID_u$.

(3) Compute $NTID_u = TID_u \oplus n_u$.

(4) Compute the value $C_u = h(B_u \oplus PW_u) \oplus n_u$.

(5) Send the message $M_1 = \{ID_u, NTID_u, C_u\}$ to the server, S.

Step 2: $S \longrightarrow u : M_2 = \{D_u, NTID_s\}$.
After receiving the message M_1, S does the following processes:

(1) Retrieve from the database the registration time, TS_u, which is corresponding with the identity, ID_u, of the connecting user. If no such corresponding user matches, the server terminates the connection. Otherwise, it goes on to the next processes.

(2) Compute $TID_u = TS_u||ID_u$, and $n'_u = NTID_u \oplus TID_u$.

(3) Compute $A_u = h(TID_u \oplus x)$ and $g^{A_u} \, mod \, p$, then $h(g^{A_u} \, mod \, p)$.

(4) Compute $n''_u = C_u \oplus h(g^{A_u} \, mod \, p)$. If $n'_u = n''_u$, the received $NTID_u$ is truly sent from u and the parameters n'_u and n''_u should be the same as n_u, which is generated by the smart card at the user's terminal. Hence, the legitimacy of the connecting user is authenticated. See Theorem 1. So, the communication will carry on. On the other hand, if $n'_u \neq n''_u$, the server terminates the connection. Furthermore, the server stores n_u in memory temporarily for later use.

(5) Create a nonce, n_s, randomly. Compute $D_u = C_u \oplus n_u \oplus n_s$ and $NTID_s = TID_u \oplus n_s$. Then the server sends the message, $M_2 = \{D_u, NTID_s\}$, to the connecting user, u.

Theorem 1: If $n'_u = n''_u$, the user, u, is authenticated.

Proof. Since $NTID_u = TID_u \oplus n_u$, thus, $n'_u = NTID_u \oplus TID_u = n_u$.
Also, given $B_u = (g^{A_u} \, mod \, p) \oplus PW_u$, we have

$$\begin{aligned} C_u &= h(B_u \oplus PW_u) \oplus n_u \\ &= h((g^{A_u} \, mod \, p) \oplus PW_u \oplus PW_u)) \oplus n_u \\ &= h(g^{A_u} \, mod \, p) \oplus n_u. \end{aligned}$$

Then,

$$\begin{aligned} n''_u &= C_u \oplus h(g^{A_u} \bmod p) \\ &= h(g^{A_u} \bmod p) \oplus n_u \oplus h(g^{A_u} \bmod p) \\ &= n_u. \end{aligned}$$

It follows that $n'_u = n''_u = n_u$.
The nonce, n_u, is generated at the user terminal when the user, u, inserts his smart card into a card reader. So it is fresh and unique. It is also embedded in $NTID_u$ and never exposed. No one can impersonate it or pry about it. Both TID_u and $NTID_u$ are unique, and $NTID_u$ can be computed by u only. Once $n'_u = n''_u$ is proven, we verify $NTID_u$ is really transmitted by u. Hence, the genuineness of the user, u, is authenticated. □

Step 3: $u \longrightarrow S : M_3 = \{E_u\}$.
When u receives the message M_2, he executes the following processes:

(1) Compute $n'_s = NTID_s \oplus TID_u$ and $n''_s = Cu \oplus n_u \oplus D_u$. If $n'_s = n''_s$, the communication goes on. In this situation, both n'_s and n''_s are equal to n_s, which is generated by the server. Thus, the server is authenticated. See Theorem 2. On the other hand, if $n'_s \neq n''_s$, u ceases the communication. Furthermore, u keeps n_s temporarily at the user terminal for later use.

(2) Compute $E_u = (C_u \oplus n_u)||(n_s + 1)$. Then u sends the message $M_3 = \{E_u\}$ to S. The parameter $n_s + 1$ is the response to the server.

Theorem 2: The server, S, is authenticated if $n'_s = n''_s$.

Proof. Since $NTID_s = TID_u \oplus n_s$, $n'_s = NTID_s \oplus TID_u = n_s$.
Also, since $D_u = C_u \oplus n_u \oplus n_s$, $n''_s = C_u \oplus n_u \oplus D_u = n_s$.
Then, we have $n'_s = n''_s = n_s$.
The nonce, n_s, is immediately generated by S, when S verifies the genuineness of the user, u. So n_s is fresh and unique. The transformed identity, TID_u is also unique. Thus, $NTID_s$ is unique and it can be computed by the server only. Furthermore, D_u is computed with C_u, n_u and n_s. A false server can not forge all of them. Once $n'_s = n''_s$ is proven, the integrity of S is authenticated. □

Step 4: After receiving the message, M_3, the server finds E_u in it. Since $B_u \oplus PW_u = g^{A_u} \bmod p$, $C_u = h(B_u \oplus PW_u) \oplus n_u = h(g^{A_u} \bmod p) \oplus n_u$. Thus, $C_u \oplus n_u = h(g^{A_u} \bmod p)$. So, $E_u = (C_u \oplus n_u)||(n_s + 1) = h(g^{A_u} \bmod p) \| (n_s + 1)$, and it is really the string concatenation of $h(g^{A_u} \bmod p)$ and $n_s + 1$. The server can easily extract $n_s + 1$ from E_u and find n_s in there. At this time, the server ensures that the authenticating user does have the nonce, n_s.

Now, both the user and the server can try for a session key agreement.

3.3 The key agreement phase

After receiving the nonce, n_s, sent from the server, the user creates a session key $SK_u = h((B_u \oplus PW_u)||n_s||n_u)$. Once the server ensures that u has the nonce, n_s, it generates a session key $SK_s = h((g^{A_u} \bmod p)||n_s||n_u)$.

Since $B_u = (g^{A_u} \bmod p) \oplus PW_u$ is computed in the registration phase,

$$h((B_u \oplus PW_u)||n_s||n_u) = h((g^{A_u} \bmod p)||n_s||n_u).$$

Thus, $SK_u = SK_s$. Therefore, the key agreement is achieved and the session key for the session communication is

$$SK = h((B_u \oplus PW_u)||n_s||n_u) = h((g^{A_u} \bmod p)||n_s||n_u).$$

3.4 The password update phase

When a user wants to change his password for personal reasons or for the sake of security. He can do so at the user's terminal by performing the following:

Step 1: Insert the smart card into a reader and announce a password update request at the user's terminal.

Step 2: Key in the original password, PW_u. The smart card calculates $B_u \oplus PW_u$.

Step 3: Responding to the challenge of the smart card, the user gives a new password PW^*_u. The smart card calculates $B^*_u = (B_u \oplus PW_u) \oplus PW^*_u$ and then replaces B_u with this new B^*_u. At this time, the password update phase is completed.

4 Security analysis

Not only do we concern with the efficiency and the specialties of our scheme, but also we ask for security and the computational complexity in our proposed scheme. In this section, we will display the strength of our scheme first, and later we discuss the computational complexity. The security analysis is listed as follows:

(1) Our scheme can overcome the guess attacks:
The user is allowed to select his own identity and password freely in our scheme, so he is apt to choose easy-to-remember or meaningful identity and password. In this situation, it seems easy to guess the identity and the password of a legitimate user. However, the construction of transformed identity in our proposed scheme makes the transformed identity be an independent unity. The uniqueness can prevent the transformed identity from being duplicate and resist the guess attacks. An intruder guesses a legitimate

user's identity. The guessed identity can not be converted into a valid transformed identity without the exact registration time, which is stored in the user's smart card. As a result, a intruder's intent to access a remote server should be rejected without a valid transformed identity.

(2) Our scheme is capable of resisting the man-in-the-middle attacks:
A malicious intruder may intercept or eavesdrop on the communication between a legitimate user, u, and the server, S. After intercepting the message M_1 sent by u, he may impersonate u and replay the message to S. Then, he waits for a response message from S. The intruder can not compute the efficient TID_u from the intercepted $NTID_u$ without the nonce, n_u, which is generated randomly by the smart card and is never exposed on the communication. Even though the intruder has the response message, M_2, from S, he can not extract the nonce, n_s, from $NTID_s$, which is included in M_2, because he has no TID_u at hand. The nonce, n_s, is generated by S and is needed to authenticate the server in Step 3(1) in the login and authentication phase. This nonce is also required to achieve the session key agreement. Furthermore, the intruder must respond the server with $n_s + 1$. Because the nonce, n_s, is unavailable, this response can't be completed, either. Eventually, an illegitimate user should be rejected and the connection is terminated.

On the other hand, when a malicious person intercepts the message M_1, he may pretend to be the server that u is connecting to. Furthermore, he has no TID_u and he can't compute it because he has no TS_u at hand. The intruder has no nonce, n_u, either. Thus, he can not send an available parameters to the user u for authenticating the integrity of the server. The communication terminates when the authentication fails.

(3) An intruder can not achieve session key agreement:
The user's password is never exposed in the transmission. An intruder can not intercept the password or any information about it. Meanwhile, the parameter B_u is stored in the user's smart card, no one can access it. So, the parameter $g^{A_u} \bmod p$ can not be computed from $B_u \oplus PW_u$. On the other hand, if an intruder intends to compute $g^{A_u} \bmod p$ directly. He needs to compute $A_u = h(TID_u \oplus x)$ first. But, the secret key x of the server is kept confidentially. No one can have it. Hence, it is impossible to compute $g^{A_u} \bmod p$ directly. Therefore, no session key agreement can be achieved without all of $g^{A_u} \bmod p$, n_u and n_s at hand.

(4) An intruder will be confronted with the complexity of the discrete logarithm:
The secret key x of the server is protected by the one-way hash function. It is not possible to derive it from $A_u = h(TID \oplus x)$. Trying to solve out A_u from $g^{A_u} \bmod p$ is also impossible, because the adversary will be confronted with the difficulty and the complexity of the discrete logarithm problem. Without secret key x, an adversary can not pretend to be the server, S, in the communication. The parameter, B_u, can't be derived without A_u. Thus, an adversary can not pretend to be the connecting user, u, either.

5 The efficiency and specialties of our scheme

From the procedures of the construction, we point out some merits in our scheme. We concern not only efficiency but also special properties.

(1) No verification table is needed:
Once a prospective user, u, offers his identity, ID_u, and password, PW_u, in registration phase. The server, S, takes note of the registration time, TS_u, to derive the transformed identity, TID_u. Then, S calculates the parameter, B_u, and stores it in a smart card. When the legitimate user wants to access the system, he only gives his selected identity to compute the transformed identity and then transmits it to the remote server. The smart card also generates automatically a nonce, n_u, to compute the authenticating values, C_u and $NTID_u$. Then the values are transmitted to the server. It is not necessary for the remote server to set up any verification table of passwords or other personal information.

(2) The transformed identity is unique:
The construction of transformed identity makes the identity unique. A few users could select the same identities, but the transformed identities should eventually be different since our scheme takes the registration time into account. It prevents the duplication from happening.

(3) The user's identity and password can be selected freely:
Since our proposed scheme uses the transformed identity to discriminate different users, the original identity is allowed to be selected according to the user's preference. Taking into account the registration time, the proposed scheme converts the selected identity into transformed identity. The transformed identities should be different from one another even if the selected identities might be the same. Thus, a user's identity can be selected freely.

The transformed identity is used to compute the parameter A_u. Then, $g^{A_u} \bmod p$ is computed. The parameter B_u is generated by performing exclusive-or operation on PW_u and $g^{A_u} \bmod p$. Because B_u is stored in the user's smart card, no one can pry about it. Therefore, the password can also be selected freely.

(4) Diffie-Hellman scheme is used:
In registration phase, the server calculates the parameter B_u through Diffie-Hellman scheme to enhance

security. Because the computation of modular exponentiation is burdensome for a smart card, the proposed scheme makes the server execute the operation in order to lessen the troublesome implementation for smart card and to speed up the computation.

(5) The computations proceed very quickly and the load is low:
The modular exponentiation is the only burdensome and time-consuming computation. It is used on the Diffie-Hellman scheme and is performed only once at the remote server. The other computations at both the user's terminal and the remote server are just the one-way hash functions, string concatenations and the exclusive-or operations. The computations proceed very quickly, and the load is extremely low for either of them. The Table 1 demonstrates the computational complexity is simple.

(6) The password can be conveniently updated at the user's terminal:
The server needs no password-verification table to check the a user's genuineness. The proposed scheme allows a user to update his password at his terminal. It is convenient and efficient for users.

(7) The mutual authentication is executed:
The scheme can mutually authenticate each other between the user and the server. From the Theorem 1 and 2, the correct methods of the mutually authentication between the user and the remote server are proven.

At the end of this section, we compare our proposed scheme with some other schemes on the computational complexity and the performances.

The comparison on computational complexity is also listed in Table 1.

From an objective point of view about the performance, we include some criteria in the following items:

Item 1. No verification table needed: At the remote server, a password-verification table is not needed to authenticate the users.

Item 2. Using unique transformed identity: Describe whether a user can choose his identity according to his preference and prevent it from duplication.

Item 3. Choosing a password freely: Display whether a scheme allows a user to choose his password freely or not.

Item 4. Mutual authentication: Demonstrate whether a legitimate user and the remote server can mutually authenticate each other or not.

Item 5. Password update conveniently: Discuss whether a user can conveniently update his password at the user's terminal or not.

Item 6. Session key agreement: Show whether a scheme can achieve the session key agreement or not.

Item 7. Avoiding time synchronization problem: Exhibit whether a scheme can avoid the time synchronization problem or not.

The result of the comparisons on the performances is listed in Table 2.

6 The conclusions

We have proposed an exquisite mutual authentication scheme without verification table of passwords and other users' personal information. The proposed scheme includes session key agreement and convenient password update. Our scheme uses the registration time to create the unique transformed identity in order to discriminate a user from the others efficiently, even if they may choose the same value for their identities. Through the storage of important information in the smart card, the proposed scheme can generate necessary parameters without exposing the password in transmission. Our scheme can withstand the replay attacks and resist the man-in-the-middle attacks. Moreover, the security of our scheme relies on the intractability of discrete logarithm because the Diffie-Hellman scheme is used.

References

[1] S.M. Bellovin, M. Merritt, (1993) Augmented encrypted key exchange: A password-based protocols secure against dictionary attacks and password file compromise, *Proceedings of First ACM Conference on Computer & Communications Security,* pp.244–250.

[2] Y. Ding, P. Horster, (1995) Undetectable on-line password guessing attacks, *ACM Operating Syst. Rev.*, pp.77–86.

[3] DV. Klein, (1990) Foiling the cracker: a survey of, and improvements to password security, *Proceedings of the second USENIX UNIX security workshop,* pp.5–14.

[4] R. Morris, K. Thompson, (1979) Password security: a case history. *Communications of the ACM,* 22(11), pp.594–597.

[5] V. Goyal, V. Kumar, M. Singh, A. Abraham, and S. Sanyal, (2006) A new protocol to counter online dictionary attacks, *Computers & Security*, 25, pp.114–120.

[6] H.Y. Chien, J.K. Jan, (2003) Robust and simple authentication protocol, *Computer Journal*, 46, pp.193–201.

[7] C.L. Lin, H.M. Sun, T. Hwang, (2001) Attacks and solutions on strong-password authentication, *IEICE Trans. Commun.* E84-B, No. 9, pp.2622–2627.

[8] S. Halevi, H. Krawczyk, (1998) Public-key cryptography and password protocols, *Proceedings of the 5th ACM Conference on Computer and Communications Security*, San Francisco, CA, pp.122–131.

[9] C.C. Chang, W.Y. Liao, (1994) A Remote Password Authentication Scheme Based upon ElGamal's Signature Scheme, *Computer & Security*, Vol. 13, pp.137–144.

[10] C.C. Chang, L.H. Wu, (1990) A Password Authentication Scheme Based upon Rabin's Public-Key Cryptosystem, *Proceedings of International Conference on Systems Management '90*, Hong Kong, pp.425–429.

[11] M.S. Hwang, L.H. Li, (2000) A new remote user authentication scheme using smart card, *IEEE Transactions on Consumer Electronics*, 46(1), pp.28–30.

[12] W. S. Juang, (2004) Efficient password authenticated key agreement using smart card, *Computer & Security*, 23, pp.167–173.

[13] Y.C. Chen, L.Y. Yeh, (2005) An efficient nonce-based authentication scheme with key agreement, *Applied Mathematics and Computation*, 169, pp.982–994.

[14] M.S. Hwang, C.C. Lee, Y.L. Tang, (2002) A simple remote user authentication scheme, *Mathematical and Computer Modelling*,36, pp.103–107.

[15] W. Diffie, M. Hellman, (1976) New directions in cryptography, *IEEE Trans. Inform. Theory*, 22, pp.476–492.

[16] C.T. Wang, C.C. Chang, C.H. Lin, (2004) Using IC Cards to Remotely Login Passwords without Verification Tables, *Proceedings of the 18th International Conference on Advanced Information Networking and Application*(*AINA*), Fukoka, Japan, pp.321–326.

Phase	Registration	Login and Authentication	Key Agreement	Password Update
Our scheme	1Co 1Ha 2⊕ 1ME	3Co 2Ha 9⊕ 1ME	Yes	Yes
Chien et al [6]	2Co 3Ha 1⊕	5Co 17Ha 10⊕	No	No
Hwang and Li [11]	1ME	2Ha 2⊕ 5ME 2MM	No	No
Juang's [12]	1Ha 1⊕	1Co 4Ha 1⊕ 3En 3De	Yes	No
Hwang et al [14]	2Ha 2⊕	1Ha 2⊕	No	Yes

Co: concatenation; Ha: one-way hash function; ⊕: exclusive-or;
ME: modular exponentiation; MM: modular multiplication;
En: encryption; De: decryption

Table 1: Comparison on Computational Complexity

Criterion item	No verification table	Using transformed ID	Choosing PW freely	Mutual authentication	PW update	Session key agreement	Avoiding synchronization
Our scheme	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Chien et al's [6]	No	No	Yes	Yes	No	No	Yes
Hwang and Li's [11]	Yes	No	No	No	No	No	No
Juang's [12]	Yes	No	Yes	Yes	No	Yes	Yes
Hwang et al's [14]	Yes	No	Yes	No	Yes	No	No

Table 2: The result of the comparisons on performancesamong schemes