An Efficient and Secure Smart Card Based Password Authentication Scheme

Yanjun Liu[1], Chin-Chen Chang[1], and Shih-Chang Chang[2]
(Corresponding author: Chin-Chen Chang)
Department of Information Engineering and Computer Science[1]
Feng Chia University, Taichung 407, Taiwan
Department of Computer Science and Information Engineering[2]
National Chung Cheng University, Chiayi 621, Taiwan
(Email: alan3c@gmail.com)
(Received Sept. 16, 2015; revised and accepted Dec. 7 & Dec. 30, 2015)

Abstract

With the advancement of internet network technologies, remote user authentication schemes using smart cards have been widely adopted. In order to satisfy the requirements of a remote user authentication scheme, the smart card has become an essential device, one that is widely used because of its low computational cost and expedient portability. Recently, Li et al. pointed out some security weaknesses in Chen et al.'s scheme, such as forward secrecy and the wrong password login problem, and proposed an enhanced user password authentication scheme based on smart card. However, we found weaknesses in their scheme. Accordingly, we propose an enhanced scheme to remedy these security weaknesses, and prove that our scheme is more secure and efficient for network application with several positive properties.

Keywords: Authentication, hash function, security, smart card

1 Introduction

With the development of Internet network technologies, remote user authentication schemes using smart cards have been widely adopted. It is generally known that the first proposed remote authentication scheme was based on a password to identify a legitimate user even over an insecure channel [1, 12, 19]. This is the subject of a published research by Lamport in 1981 [8]. It has been claimed that there is a potential security threat caused by a stored verifier table on a remote authentication system, because the verifier table risks being modified by an adversary and has a high maintenance cost, even though all secret passwords can be encrypted against the threat of disclosure. Later, Hwang and Li [4] presented the weakness of Lamport's scheme and proposed a new scheme based on the ElGamal public-key encryption system [3] to solve the corresponding problem. In this novel method, there is no need to maintain a verifier table to achieve remote user authentication. In view of the low cost and capacity of cryptosystems, Sun [18] developed an authentication scheme to enhance the performance efficiency of Hwang and Li's scheme by involving several one-way hash operations, such that the scheme could serve as an ideal substitute for high-cost modular exponentiations. Nevertheless, these two mentioned schemes could not provide users with a free choice of passwords and mutual authentication.

Since a smart card has tamper-resistant properties, it can solve the problem of maintaining the verifier table on the server side. In a smart card based authentication system, only the user is required to hold a smart card, which was issued by the server for more convenient communication and which contained all kinds of stored secret information. Many related studies [5, 6, 7, 11, 16] have investigated smart cards and the smart card has become essential in remote authentication schemes. In 2009, Xu et al. [20] proposed a novel user authentication and claimed that their scheme is secure against various attacks. However, Song [14] and Sood et al. [15] found that Xu et al.'s scheme has some weaknesses and proposed improved schemes. Subsequently, Chen et al. [2] pointed out that there are vulnerabilities on Song and Sood et al.'s schemes. Then, Chen et al. presented an enhanced version to solve the weaknesses. Recently, Li et al. [9] claimed that Chen et al.'s scheme is still insecure and proposed a modified smart card based remote user password authentication scheme. Unfortunately, we find that there are weaknesses in Li et al.'s scheme, such as from a man-in-the-middle attack and an insider attack. Hence, we propose a novel scheme to defend against these security weaknesses. Furthermore, our proposed scheme has better computational efficiency, which has become clear by comparing our work with previous schemes. In addition, our scheme has the following properties:

F1. Mutual authentication: Both the legal user and the remote server can authenticate each other successfully.
F2. Session key agreement: The legal user and the remote server can negotiate a session key and utilize it to process subsequent communication.
F3. Freely chosen and exchanged password: A legal user can freely choose and change the password.
F4. Withstands a man-in-the-middle attack: Our scheme can withstand a man in the middle attack.
F5. Withstands an insider attack: No adversary can present an insider attack.
F6. Withstands a replay attack: No one can perform a replay attack.
F7. Perfect forward secrecy: Even if an adversary can obtain contiguous knowledge of the long-term key, the adversary cannot derive previous session keys.
F8. Satisfying known-key security: No one can utilize the secret information of a legal user to derive the session key.

The rest of this paper is organized as follows. In Section 2, we briefly review Li et al.'s smart-card-based password authentication scheme and Section 3 analyzes its weaknesses. In Section 4, we propose our scheme. Section 5 gives security and performance analyses of the proposed scheme. Finally, we present our conclusions in Section 6.

Table 1: The notations used in both Li et al.'s and our proposed schemes

U_i	The user i
S	The authentication server
ID_i	The identity of the user U_i
PW_I	The password of the user U_i
x	The master secret key of the server S
T_i	The timestamp of the user U_i
T_i'	The time of receiving the login request message
T_s	The timestamp of the server S
T_s'	The time of receiving the mutual authentication message
ΔT	A valid time threshold
$h(\cdot)$	A collision-free one-way hash function
$\|$	The message concatenation operation
$\oplus$	The bitwise XOR operation
sk	The shared session key

2 Review of Li et al.'s Scheme

In this section, we briefly review Li et al.'s smart card based password authentication scheme [9] before demonstrating its weaknesses. Their scheme is an improvement of Chen et al.'s scheme [2] and the security depends on the hardness of solving the discrete logarithm problem [13]. The notations used in both Li et al.'s and our proposed schemes are listed in Table 1.

Their scheme involves two parties, i.e., the user U_i and the server S, to communicate with each other to perform the following four phases: (1) The registration phase; (2) the login phase; (3) the authentication phase; and (4) the password change phase. Since the security basis of their scheme is the discrete logarithm problem, the server S needs to initialize some parameters before the registration phase. The server S selects two large prime numbers p and q that satisfy $p = 2q+1$, the master secret key $x \in Z_q^*$ (Z_q denotes the ring of integers modulo q and Z_q^* denotes the multiplicative group of Z_q) , and a collision-free one-way hash function $h(\cdot)$. Then, the four phases are executed as follows and are illustrated in Figure 1.

2.1 Registration Phase

Step 1. The user U_i selects his/her identity ID_i and password PW_i and submits them to the server S for registration over a secure channel.

Step 2. The server S computes two parameters: $A_i = h(ID_i \| PW_i)^{PW_i} \bmod p$ and $B_i = h(ID_i)^{(x+PW_i)} \bmod p$.

Step 3. The server S stores the data $\{A_i, B_i, h(\cdot), p, q\}$ on a new smart card and issues the smart card to the user U_i over a secure channel.

2.2 Login Phase

Step 1. The user U_i inserts his/her smart card into a card reader and inputs his/her identity ID_i and password PW_i.

Step 2. The smart card computes $A_i^* = h(ID_i \| PW_i)^{PW_i} \bmod p$ and examines whether A_i^* is equal to A_i. If the equation holds, the smart card continues to perform Step 3; otherwise, the smart card terminates this session.

Step 3. The smart card randomly selects a number $\alpha \in_R Z_q^*$ and computes the following parameters:

$$\begin{aligned} C_i &= B_i/h(ID_i)^{PW_i} \bmod p, \\ D_i &= h(ID_i)^{\alpha} \bmod p, \\ M_i &= h(ID_i \| C_i \| D_i \| T_i), \end{aligned}$$

where T_i is the current timestamp of the user U_i.

Step 4. The smart card sends the login request message $\{ID_i, D_i, M_i, T_i\}$ to the server S.

2.3 Authentication Phase

Step 1. The server S verifies whether ID_i is valid and $T_i^{'} - T_i \leq \Delta T$, where $T_i^{'}$ is the time of receiving the login request message and ΔT is a valid time threshold. If both conditions are true, the server S

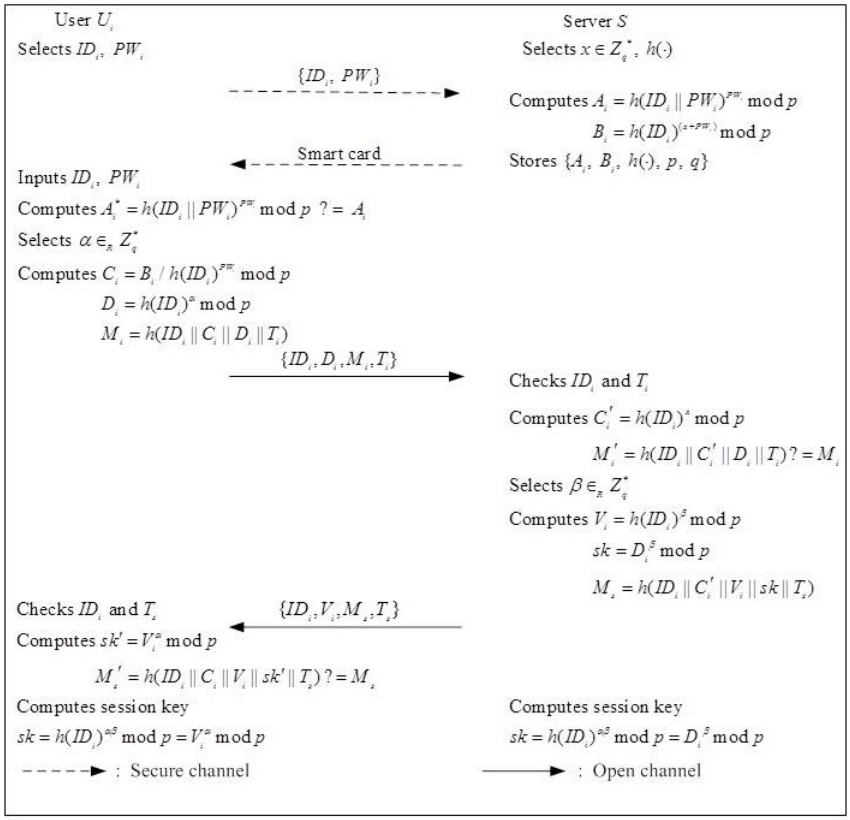

Figure 1: Li et al.'s scheme

continues to execute Step 2; otherwise, the server S rejects the login request.

Step 2. The server S computes two parameters: $C_i^{'} = h(ID_i)^x \bmod p$ and $M_i^{'} = h(ID_i \parallel C_i^{'} \parallel D_i \parallel T_i)$ Then, the server S compares whether $M_i^{'}$ equals M_i. If they are equal, the server S confirms that the user U_i is valid and the login request is accepted; otherwise, the login request is rejected.

Step 3. The server S randomly selects a number $\beta \in_R Z_q^*$ and computes the following parameters:

$$\begin{aligned} V_i &= h(ID_i)^{\beta} \bmod p, \\ sk &= D_i^{\beta} \bmod p, \\ M_s &= h(ID_i \parallel C_i^{'} \parallel V_i \parallel sk \parallel T_s), \end{aligned}$$

where T_s is the current timestamp of the server S.

Step 4. The server S sends the mutual authentication message $\{ID_i, V_i, M_s, T_s\}$ to the user U_i.

Step 5. Upon receiving the message $\{ID_i, V_i, M_s, T_s\}$, the user U_i checks the validity of ID_i and whether $T_s^{'} - T_s \leq \Delta T$, where $T_s^{'}$ is the time of receiving the mutual authentication message. If both of them hold, the user U_i continues to perform Step 6; otherwise, the user U_i terminates this connection.

Step 6. The user U_i computes two parameters: $sk^{'} = V_i^{\alpha} \bmod p$ and $M_s^{'} = h(ID_i \parallel C_i \parallel V_i \parallel sk^{'} \parallel T_s)$. Then, the user U_i checks whether $M_s^{'}$ equals M_s. If they are equal, the validity of the server S is authenticated; otherwise, the session is terminated.

Step 7. The user U_i and the server S construct a shared session key $sk = h(ID_i)^{\alpha\beta} \bmod p$ to ensure the secret communication.

2.4 Password Change Phase

Step 1. The user U_i inserts his/her smart card into a card reader, enters his/her old identity ID_i and password PW_i, and requests to change the password.

Step 2. The smart card computes $A_i^* = h(ID_i \parallel PW_i)^{PW_i} \bmod p$ and checks whether A_i^* equals A_i that is stored in the smart card. If the equation holds, the user U_i submits the new password PW_i^{new}; otherwise, the smart card rejects the password change request.

Step 3. The smart card computes $A_i^{new} = h(ID_i \parallel PW_i^{new})^{PW_i^{new}} \bmod p$ and $B_i^{new} = B_i \cdot h(ID_i)^{PW_i^{new}} / h(ID_i)^{PW_i} \bmod p$. Then, the smart card replaces A_i and B_i with A_i^{new} and B_i^{new}, respectively.

3 Weaknesses of Li et al.'s Scheme

Li et al.'s scheme [9] can correct the design flaws of Chen et al.'s scheme [2], such as by ensuring perfect forward secrecy, quickly detecting the wrong password via the smart card without interacting with the server in the login phase, and provides a friendly and efficient password change. Additionally, Li et al. claimed that their scheme is very secure and can resist various types of attacks. We found, however, that Li et al.'s scheme cannot withstand a man-in-the-middle attack or an insider attack. In addition, the use of modulus exponential operations incurs a considerable computational cost. The details of these weaknesses are discussed below.

3.1 Man-in-the-middle Attack

Li et al.'s scheme is vulnerable to a man-in-the-middle attack. Suppose that there exists an attacker U_E between the user U_i and the server S. The attacker U_E can intercept the login request message and the mutual authentication message transmitted between the user U_i and the server S, and then modify these messages. U_E can act as the user U_i to communicate with the server S and act as the server S to communicate with the user U_i without detection. This type of attack can be described as follows:

Step 1. In the login phase, the user U_i's smart card sends the login request message $\{ID_i, D_i, M_i, T_i\}$ to the server S. The attacker U_E intercepts this message.

Step 2. Since $M_i = h(ID_i \parallel C_i \parallel D_i \parallel T_i)$, the attacker U_E uses the intercepted values of ID_i, D_i, T_i, and M_i to guess C_i Due to the fact that $C_i = B_i/h(ID_i)^{PW_i} \bmod p = h(ID_i)^x \bmod p$ would remain the same in different sessions of the user U_i and the server S, the attacker U_E can easily determine the value of C_i.

Step 3. The attacker U_E generates $D_E = h(ID_i)^e$ and computes $M_E = h(ID_i \parallel C_i \parallel D_E \parallel T_E)$. After that, U_E sends $\{ID_i, D_E, M_E, T_E\}$ to the server S.

Step 4. The server S first checks the validity of ID_i and T_E, and then computes $C_i^{'} = h(ID_i)^x \bmod p$ and $M_E^{'} = h(ID_i \parallel C_i^{'} \parallel D_E \parallel T_E)$ Afterwards, the server S compares whether $M_E^{'}$ equals M_E. If they are equal, the server S believes that the attacker U_E is authenticated as the user U_i.

Step 5. The server S computes $V_i = h(ID_i)^\beta \bmod p$, $sk = D_E^\beta \bmod p = h(ID_i)^{e\beta} \bmod p$, and $M_s = h(ID_i \parallel C_i^{'} \parallel V_i \parallel sk \parallel T_s)$, then sends the mutual authentication message $\{ID_i, V_i, M_s, T_s\}$ to the user U_i. The attacker U_E intercepts this message.

Step 6. The attacker U_E generates $V_E = h(ID_i)^e$, $sk_E = V_E \cdot D_i = h(ID_i)^{e\alpha} \bmod p$ and $M_E^{''} = h(ID_i \parallel C_i^{'} \parallel V_E \parallel sk_E \parallel T_E^{'})$. After that, U_E sends $\{ID_i, V_E, M_E^{''}, T_E^{'}\}$ to the user U_i.

Step 7. The user U_i first checks the validity of ID_i and $T_E^{'}$, and then computes $sk^{'} = V_E^\alpha \bmod p = h(ID_i)^{e\alpha} \bmod p$ and $M_E^{'''} = h(ID_i \parallel C_i \parallel V_E \parallel sk^{'} \parallel T_E^{'})$ Afterwards, the user U_i checks whether $M_E^{'''}$ equals $M_E^{''}$. If they are equal, the user U_i believes that the attacker U_E is authenticated as the server S.

After performing the authentication phase, the user U_i believes that the attacker U_E is the server S and the server S believes that the attacker U_E is the user U_i. Moreover, user U_i and the server S trust that they have established a common session key. However, server S and the attacker U_E share a session key $sk = h(ID_i)^{e\beta} \bmod p$; and user U_i and the attacker U_E share another session key $sk^{'} = h(ID_i)^{e\alpha} \bmod p$. Consequently, Li et al.'s scheme cannot prevent a man-in-the-middle attack.

3.2 Insider Attack

If server S directly obtains user U_i's password PW_i, an insider attack takes place when an intruder steals PW_i from S. In Li et al.'s scheme, the user U_i selects their password PW_i and submits it to the server S for registration over a secure channel. Therefore, server S can obtain the user U_i's password PW_i and cannot withstand an insider attack.

3.3 Computational Inefficiency

From Li et al.'s scheme, we can see that it uses too many modulus exponential operations, which can incur unnecessary overhead. The computational cost in the login and authenticated phases are $3E + 1M + 3H$ and $4E + 4H$, respectively, where E is modulus exponential operations, M is multiplication/division operations, and H is hashing operations. Li et al. claimed that although their scheme requires a higher computational cost, it can achieve higher security and usability compared with other related schemes. Unfortunately, this is not true according to the discussion in Subsections 3.1 and 3.2. In fact, the modulus exponential operations can be replaced with other appropriate operations to reduce the computational cost.

4 Our Proposed Scheme

To overcome the aforementioned weaknesses, we propose a novel smart card based password authentication scheme, which is secure and more efficient. By using the combination of collision-free one-way hash functions, bitwise XOR ($\oplus$) and concatenation ($\parallel$) operations instead of modulus exponential operations, our proposed scheme can significantly enhance computational efficiency while satisfying various security requirements. Our proposed scheme consists of four phases: (1) The registration phase; (2) the login phase; (3) the authentication phase; and (4) the

password change phase. In the following, we will describe the proposed scheme in detail.

4.1 Registration Phase

At the beginning of our proposed scheme, the server S selects the master secret key x and a collision-free one-way hash function $h(\cdot)$. Then, the user U_i registers to the server S by the way below:

Step 1. The user U_i first selects his/her identity ID_i, password PW_i, and a random number r, and then computes $h(r \parallel PW_i)$. U_i submits $\{ID_i, h(r \parallel PW_i)\}$ to the server S for registration over a secure channel.

Step 2. The server S computes the following parameters:

$$\begin{aligned} A_i &= h(ID_i \oplus x) \parallel h(x), \\ B_i &= A_i \oplus h(r \parallel PW_i), \\ C_i &= h(A_i \parallel ID_i \parallel h(r \parallel PW_i)). \end{aligned}$$

Step 3. The server S stores the data $\{B_i, C_i, h(\cdot)\}$ on a new smart card and issues the smart card to the user U_i over a secure channel.

Step 4. The user U_i stores the random number r into the smart card.

The registration phase is depicted in Figure 2.

4.2 Login Phase

This phase is invoked whenever the user U_i wants to login to the server S. The steps of this phase are conducted as follows:

Step 1. The user U_i inserts his/her smart card into a card reader and inputs his/her identity ID_i and password PW_i.

Step 2. The smart card first computes two parameters: $A_i^{'} = B_i \oplus h(r \parallel PW_i)$ and $C_i^{'} = h(A_i^{'} \parallel ID_i \parallel h(r \parallel PW_i))$. Then, the smart card examines whether $C_i^{'}$ is equal to C_i. If the equation holds, the smart card continues to perform Step 3; otherwise, the smart card terminates this session.

Step 3. The smart card randomly selects a number α and computes the following parameters:

$$\begin{aligned} D_i &= h(ID_i \oplus \alpha), \\ E_i &= A_i^{'} \oplus \alpha \oplus T_i, \end{aligned}$$

where T is the current timestamp of the user U_i.

Step 4. The smart card sends the login request message $\{ID_i, D_i, E_i, T_i\}$ to the server S.

4.3 Authentication Phase

After completing this phase, the user U_i and the server S can mutually authenticate each other and establish a shared session key for the subsequent secret communication. The steps of this phase are shown as follows:

Step 1. The server S verifies whether ID_i is valid and $T_i^{'} - T_i \leq \Delta T$, where $T_i^{'}$ is the time of receiving the login request message and ΔT is a valid time threshold. If both conditions are true, the server S continues to execute Step 2; otherwise, the server S rejects the login request.

Step 2. The server S computes the following parameters:

$$\begin{aligned} A_i &= h(ID_i \oplus x) \parallel h(x), \\ \alpha^{'} &= E_i \oplus A_i \oplus T_i, \\ D_i^{'} &= h(ID_i \oplus \alpha^{'}). \end{aligned}$$

Then, the server S compares whether $D_i^{'}$ equals D_i. If they are equal, the server S confirms that the user U_i is valid and the login request is accepted; otherwise, the login request is rejected.

Step 3. The server S randomly selects a number β and computes the following parameters:

$$\begin{aligned} F_i &= h(ID_i \oplus \beta), \\ G_i &= A_i \oplus \beta \oplus T_s, \end{aligned}$$

where T_s is the current timestamp of the server S.

Step 4. The server S sends the mutual authentication message $\{F_i, G_i, T_s\}$ to the user U_i.

Step 5. Upon receiving the message $\{F_i, G_i, T_s\}$, the user U_i checks the validity of T_s. If $T_s^{'} - T_s \leq \Delta T$, where $T_s^{'}$ is the time of receiving the mutual authentication message, the user U_i continues to perform Step 6; otherwise, the user U_i terminates this connection.

Step 6. The user U_i computes $\beta^{'} = G_i \oplus A_i^{'} \oplus T_s$ and $F_i^{'} = h(ID_i \oplus \beta^{'})$, and then checks whether $F_i^{'}$ equals F_i. If they are equal, the validity of the server S is authenticated; otherwise, the session is terminated.

Step 7. The user U_i and the server S construct a shared session key $sk = h(\alpha \parallel \beta^{'} \parallel h(A_i^{'} \oplus ID_i)) = h(\alpha^{'} \parallel \beta \parallel h(A_i \oplus ID_i))$ to ensure the secret communication.

The login and authentication phases are shown in Figure 3.

4.4 Password Change Phase

Step 1. The user U_i inserts his/her smart card into a card reader, enters his/her old identity ID_i and password PW_i, and requests to change the password.

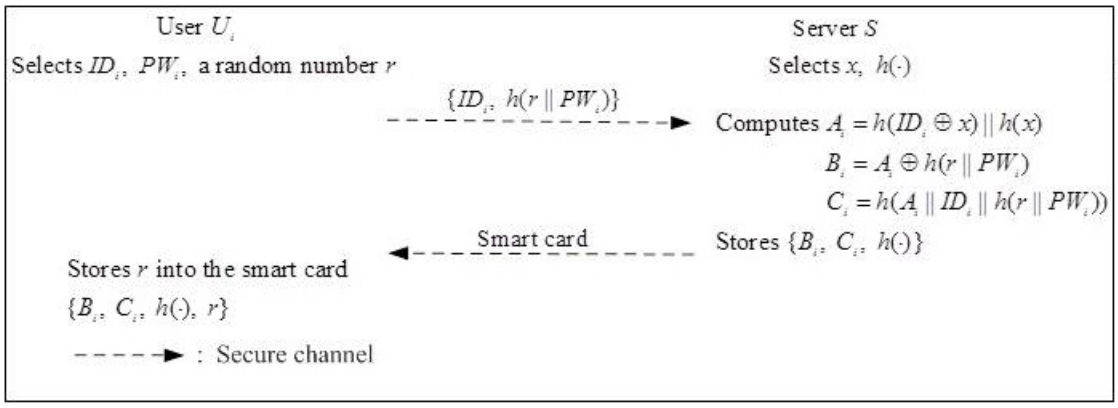

Figure 2: Registration phase of our proposed scheme

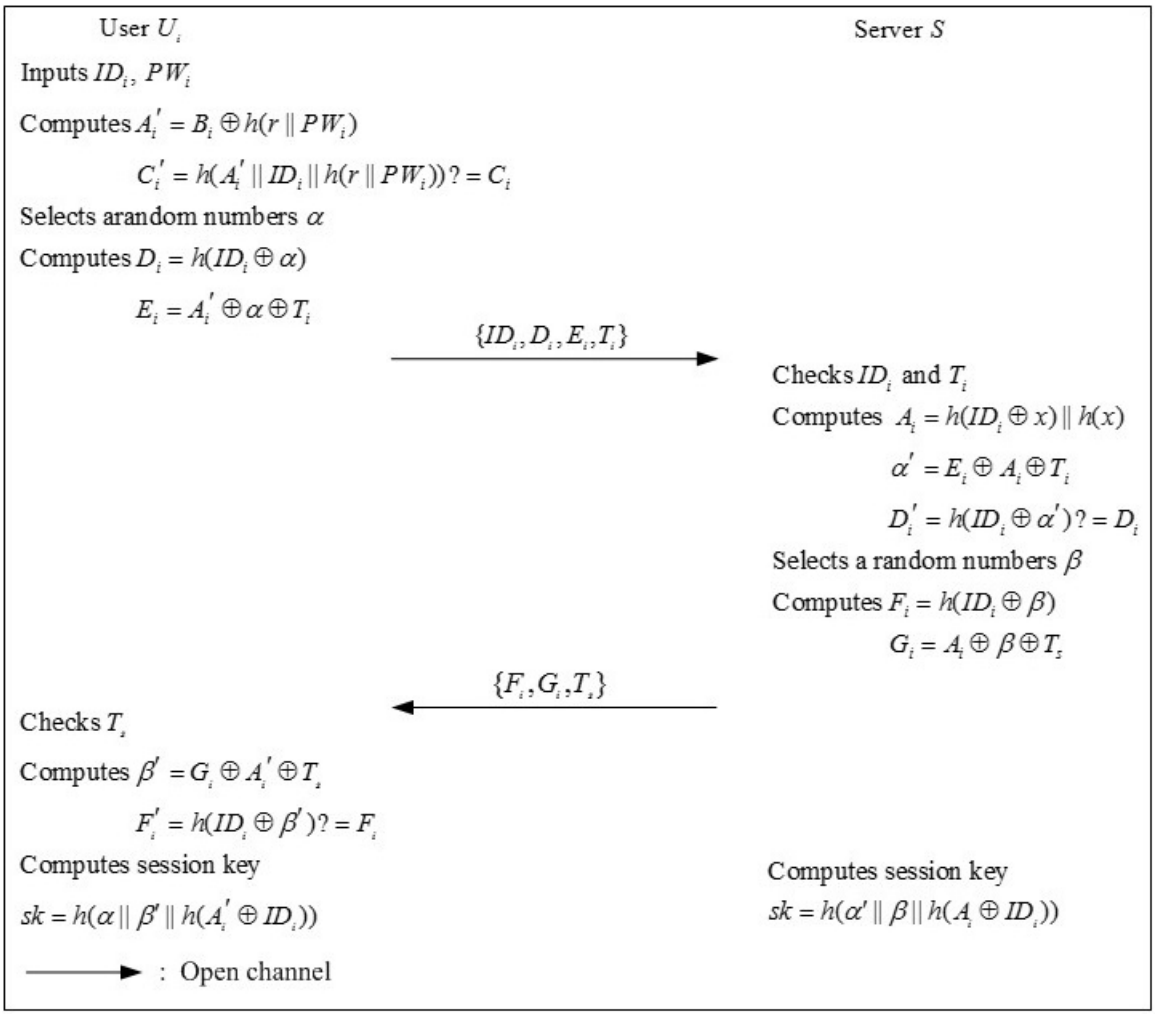

Figure 3: Login and authentication phases of our proposed scheme

Step 2. The smart card computes $A_i^* = B_i \oplus h(r \parallel PW_i)$ and $C_i^* = h(A_i^* \parallel ID_i \parallel h(r \parallel PW_i))$, and then checks whether C_i^* equals C_i that is stored in the smart card. If the equation holds, the user U_i submits the new password PW_i^{new}; otherwise, the smart card rejects the password change request.

Step 3. The smart card computes $B_i^{new} = A_i^* \oplus h(r \parallel PW_i^{new})$ and $C_i^{new} = h(A_i^* \parallel ID_i \parallel h(r \parallel PW_i^{new}))$. Then, the smart card replaces B_i and C_i with B_i^{new} and C_i^{new}, respectively.

5 Analysis of the Proposed Scheme

In this section, we analyze the security and performance of our proposed scheme and make comparisons with other related works.

5.1 Functionality and Security Analyses

5.1.1 Mutual Authentication

Our proposed scheme can achieve mutual authentication such that the user and the server can successfully verify the validity of each other. In Step 2 of the authentication phase, the server S computes $D_i^{'} = h(ID_i \oplus \alpha^{'})$, and then compares whether $D_i^{'}$ equals $D_i^{'}$ that was sent by the user U_i. If they are equal, the server S confirms that user U_i is valid. On the other hand, in Step 6 of the authentication phase, user U_i computes $F_i^{'} = h(ID_i \oplus \beta^{'})$, and then checks whether $F_i^{'}$ equals F_i that was sent by the server S. If they are equal, the validity of the server S is authenticated.

5.1.2 Session Key Agreement

After achieving mutual authentication, the user and the server must negotiate a common session key, which is used to encrypt the data transmitted between the user and the server in the subsequent confidential communications. In our proposed scheme, the user and the server share the session key $sk = h(\alpha \parallel \beta^{'} \parallel h(A_i^{'} \oplus ID_i)) = h(\alpha^{'} \parallel \beta \parallel h(A_i \oplus ID_i))$ at the end of the authentication phase.

5.1.3 Freely Chosen and Exchanged Password

Our proposed scheme allows each user to choose their password in the registration phase so that users can easily remember their passwords. In addition, each user can change their password in the password change phase. If user U_i wants to update their password, the smart card checks the validity of the old password by comparing whether C_i^* equals C_i. If so, user U_i submits the new password PW_i^{new}. The smart card uses PW_i^{new} to compute B_i^{new} and C_i^{new}, and then replaces B_i and C_i with B_i^{new} and C_i^{new}, respectively. The password change phase is friendly and efficient since the smart card can complete both the tasks of verification of old passwords and updating of new passwords. Thus, the user does not need to communicate with the server to change the password.

5.1.4 Withstanding a Man-in-the-middle Attack

Assume that there exists an attacker U_E between the user U_i and the server S. In the login phase, the attacker U_E can intercept the login request message $\{ID_i, D_i, E_i, T_i\}$ and attempts to forge it to act as user U_i. However, U_E cannot get $A_i^{'}$ and α from the intercepted message. So, if U_E generates a fake E_i and sends it to the server S, S can check that $D_i^{'}$ is not equal to the received D_i and concludes that U_E is not a valid user. On the other hand, the attacker U_E can intercept the mutual authentication message $\{F_i, G_i, T_s\}$ and wants to forge it to act as the server S. Similarly, because U_E cannot obtain A_i and β, the user U_i will not be mislead by the forged $F_i^{'}$ and concludes that U_E is not a valid server. Therefore, attacker U_E cannot modify the messages to pass the login and the authentication phases. This indicates that our proposed scheme can prevent a man-in-the-middle attack.

5.1.5 Withstanding an Insider Attack

In the registration phase, the user conceals the password in a ciphertext from the server to resist an insider attack. More specifically, user U_i first selects their password PW_i and a random number r, and then submits $h(r \parallel PW_i)$ to the server S for registration over a secure channel. As a result, server S cannot get the correct password PW_i and an insider attack will not occur.

5.1.6 Withstanding Replay Attack

A replay attack means a malicious intruder repeats or delays valid transmitted messages without detection. Our proposed scheme can resist a replay attack by utilizing timestamps in the login and authentication phases. In Step 4 of the login phase, the smart card adds the timestamp T_i into the login request message $\{ID_i, D_i, E_i, T_i\}$ and sends it to the server S. Meanwhile, in Step 4 of the authentication phase, the server S puts the timestamp T_s into the mutual authentication message $\{F_i, G_i, T_s\}$ and conveys it to the user U_i. Therefore, the user U_i and the server S can verify the occurrence of a replay attack by checking timestamps T_i and T_s.

5.1.7 Providing Perfect Forward Secrecy

Perfect forward secrecy can ensure that any previously established session keys are not disclosed to the attacker even if the server's master secret key is compromised. In our proposed scheme, the shared session key $sk = h(\alpha \parallel \beta^{'} \parallel h(A_i^{'} \oplus ID_i)) = h(\alpha^{'} \parallel \beta \parallel h(A_i \oplus ID_i))$, where $\alpha = \alpha^{'}$, $\beta = \beta^{'}$, $A_i = h(ID_i \oplus x) \parallel h(x)$ and $A_i^{'} = B_i \oplus h(r \parallel PW_i)$. Suppose that an attacker obtained the server's master secret key x. If the attacker wants to derive the previous session key sk, they must know α

Table 2: Functionality comparison of our scheme and other related schemes

	F1	F2	F3	F4	F5	F6	F7	F8
Juang et al. [6]	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
Song [14]	Yes	Yes	Yes	Yes	No	Yes	No	Yes
Chen et al. [2]	Yes	Yes	Yes	Yes	No	Yes	No	Yes
Li et al. [9]	Yes	Yes	Yes	No	No	Yes	Yes	Yes
Sun et al. [17]	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes
Li et al. [10]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Our scheme	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

F1: Mutual authentication; F2: Session key agreement; F3: Freely chosen and exchanged password; F4: Withstanding man in the middle attack; F5: Withstanding insider attack; F6: Withstanding replay attack; F7: Providing perfect forward secrecy; F8: Satisfying known-key security.

and β. However, α and β are not directly transmitted between user U_i and server S via the public channel, but are encrypted into the ciphertext D_i and F_i, respectively. Therefore, α and β cannot be obtained by the attacker, which implies that our proposed scheme provides perfect forward secrecy.

5.1.8 Satisfying Known-key Security

Known-key security guarantees that other session keys will not be derived by the attacker from the compromised session key. Our proposed scheme can satisfy known-key security by allowing user U_i and the server S to establish unique session keys in their different login and authentication phases. Assume that a session key $sk = h(\alpha \parallel \beta \parallel h(A_i \oplus ID_i))$ is compromised. Since α and β are random numbers selected by the user U_i and the server S, respectively, different values of α and β will be selected in different sessions. As a result, even if the attacker gets sk, α, and β, they cannot compute another session key $sk^{'}$ from the compromised sk without knowing $\alpha^{'}$ and $\beta^{'}$ from the other sessions. Therefore, our proposed scheme can satisfy the known-key security problem.

The functionality comparison of our proposed scheme with other related works [2, 6, 9, 10, 14, 17] is summarized in Table 2, which infers that our proposed scheme is more secure and practical than other related works.

5.2 Performance Analysis

In this subsection, we evaluate the performance of our proposed scheme in terms of computational cost. Table 3 compares the computational cost of our proposed scheme and other related schemes [2, 6, 9, 10, 14, 17]. From Table 3, we can see that all of other existing schemes involve some time-consuming operations, such as modulus exponential operations, symmetric encryption/decryption operations or multiplication/division operations. In particular, among these three operations, multiplication/division operations are faster than symmetric encryption/decryption operations while symmetric encryption/decryption operations are faster than modulus exponential operations. Fortunately, our proposed scheme only utilizes one-way hash functions, which are much faster than the mentioned three operations. Therefore, this method can significantly enhance computational efficiency while retaining higher security as shown in Table 2.

6 Conclusions

In this paper, we proposed a smart card based password authentication scheme to overcome the security weaknesses of Li et al.'s scheme. Our proposed scheme can achieve mutual authentication and users can freely choose and change their passwords. We prove that our proposed scheme can resist various types of attack, such as a man-in-the-middle attack, insider attack, and replay attack. Furthermore, our proposed scheme has better computational efficiency than other related works.

References

[1] C. C. Chang, C. Y. Lee and Y. C. Chiu, "Enhanced authentication scheme with anonymity for roaming service in global mobility networks," *Computer Communications*, vol. 32, no. 4, pp. 611–618, 2009.

[2] B. L. Chen, W. C. Kuo and L. C. Wuu, "Robust smart-card-based remote user password authentication scheme," *International Journal of Communication Systems, in press.* (`http://dx.doi.org/10.1002/dac.2368`)

[3] T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.

[4] M. S. Hwang and L. H. Li, "A new remote user authentication scheme using smart card," *IEEE Trans-*

Table 3: Computational cost comparison of our scheme and other related schemes

	C1	C2	C3	C4	C5	C6
Juang et al. [6]	1H	2H+3S	3H+3S	4H+6S+1M	1H+5S	1H+5S
Song [14]	-	2H+E	3H+1S	3H+1S+1E	Null	Null
Chen et al. [2]	-	1H+1E	2H+2M+4E	H+M+4E	3H+2M+2E	3H+2M+3E
Li et al. [9]	-	2H+2E	4H+M+4E	3H+3E	3H+2M+4E	-
Sun et al. [17]	-	2H+1S	4H+2M	4H+1S+2M	2H	-
Li et al. [10]	1H	2H+3S	8H+4S	10H+10S+1M	1H+6S	1H+9S
Our scheme	1H	3H	6H	6H	4H	-

C1: Computational cost of the user in registration phase; C2: Computational cost of the server in registration phase; C3: Computational cost of the user in login and authentication phases; C4: Computational cost of the server in login and authentication phases; C5: Computational cost of the user in password change phase; C6: Computational cost of the server in password change phase; H: Hashing operation; E: Modulus exponential operation; S: Symmetric encryption/decryption operation; M: Multiplication/division operation; Null: Cannot provide this functionality.

actions on Consumer Electronics, vol. 46, no. 1, pp. 28–30, 2000.

[5] W. S. Juang, "Efficient multi-server password authenticated key agreement using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 251–255, 2004.

[6] W. S. Juang, S. T. Chen and H. T. Liaw, "Robust and efficient password-authenticated key agreement using smart card," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 6, pp. 2551–2556, 2008.

[7] S. K. Kim and M. G. Chung, "More secure remote user authentication scheme," *Computer Communications*, vol. 32, no. 6, pp. 1018–1021, 2009.

[8] L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[9] X. Li, J. Niu, M. K. Khan, and J. Liao, "An enhanced smart card based remote user password authentication scheme," *Journal of Network and Computer Applications, in press.* (`http://dx.doi.org/10.1016/j.jnca.2013.02.034.`)

[10] X. X. Li, W. D. Qiu, D. Zheng, K. F. Chen, and J. H. Li, "Anonymity enhancement on robust and efficient password-authenticated key agreement using smart cards," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 2, pp. 793–800, 2010.

[11] J. Y. Liu, A. M. Zhou, and M. X. Gao, "A new mutual authentication scheme based on nonce and smart card," *Computer Communications*, vol. 31, no. 10, pp. 2205–2209, 2008.

[12] M. Peyravian and N. Zunic, "Methods for protecting password transmission," *Computer and Security*, vol. 19, no. 5, pp. 466–469, 2006.

[13] B. Schneier, *Applied Cryptography (2nd Edition)*, New York: Wiley, 1996.

[14] R. Song, "Advanced smart card based password authentication protocol," *Computer Standards and Interfaces*, vol. 32, no. 5, pp. 321–325, 2010.

[15] S. K. Sood, A. K. Sarje, and K. Singh, "An improvement of xu et al.'s authentication scheme using smart cards," in *Proceedings of the Third Annual ACM Bangalore Conference*, pp. 17–22, Bangalore, Karnataka, India, 2010.

[16] D. Z. Sun, J. P. Huai, J. Z. Sun, and J. X. Li, "Cryptanalysis of a mutual authentication scheme based on nonce and smart cards," *Computer Communications*, vol. 32, no. 6, pp. 1015–1017, 2009.

[17] D. Z. Sun, J. P. Huai, J. Z. Sun, J. X. Li, J. W. Zhang, and Z. Y. Feng, "Improvements of juang et al.'s password-authenticated key agreement scheme using smart cards," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 2284–2291, 2009.

[18] H. M. Sun, "An efficient remote user authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 958–961, 2000.

[19] T. C. Wu and H. S. Sung, "Authentication passwords over an insecure channel," *Computer and Security*, vol. 15, no. 5, pp. 431–439, 1996.

[20] J. Xu, W. T. Zhu, and D. G. Feng, "An improved smart card based password authentication scheme with provable security," *Computer Standards and Interfaces*, vol. 31, no. 4, pp. 723–728, 2009.

Yanjun Liu received her Ph.D. degree in 2010, in School of Computer Science and Technology from University of Science and Technology of China (USTC), Hefei, China. She is currently a postdoctor at Feng Chia University, Taichung, Taiwan. Her current research interests include information security and computer cryptography.

Chin-Chen Chang received his Ph.D. degree in computer engineering from National Chiao Tung University. His first degree is Bachelor of Science in Applied Mathematics and master degree is Master of Science in computer and decision sciences. Both were awarded in National Tsing Hua University. Dr. Chang served in National Chung Cheng University from 1989 to 2005. His current title is Chair Professor in Department of Information Engineering and Computer Science, Feng Chia University, from Feb. 2005. Prior to joining Feng Chia

University, Professor Chang was an associate professor in Chiao Tung University, professor in National Chung Hsing University, chair professor in National Chung Cheng University. He had also been Visiting Researcher and Visiting Scientist to Tokyo University and Kyoto University, Japan. During his service in Chung Cheng, Professor Chang served as Chairman of the Institute of Computer Science and Information Engineering, Dean of College of Engineering, Provost and then Acting President of Chung Cheng University and Director of Advisory Office in Ministry of Education, Taiwan. Professor Chang has won many research awards and honorary positions by and in prestigious organizations both nationally and internationally. He is currently a Fellow of IEEE and a Fellow of IEE, UK. And since his early years of career development, he consecutively won Outstanding Talent in Information Sciences of the R. O. C., AceR Dragon Award of the Ten Most Outstanding Talents, Outstanding Scholar Award of the R. O. C., Outstanding Engineering Professor Award of the R. O. C., Distinguished Research Awards of National Science Council of the R. O. C., Top Fifteen Scholars in Systems and Software Engineering of the Journal of Systems and Software, and so on. On numerous occasions, he was invited to serve as Visiting Professor, Chair Professor, Honorary Professor, Honorary Director, Honorary Chairman, Distinguished Alumnus, Distinguished Researcher, Research Fellow by universities and research institutes. His current research interests include database design, computer cryptography, image compression and data structures.

Shih-Chang Chang received his B.S. degree in 2005 and his M.S. degree in 2007, both in Department of Information Engineering and Computer Science from Feng Chia University, Taichung, Taiwan. He is currently pursuing his Ph.D. degree in Computer Science and Information Engineering from National Chung Cheng University, Chiayi, Taiwan. His current research interests include electronic commerce, information security, computer cryptography, and mobile communications.