ISSN 1392 – 124X, ISSN 2335 – 884X (online) INFORMATION TECHNOLOGY AND CONTROL, 2013, Vol. 42, No. 2

# An Improved Password-Based Remote User Authentication Protocol without Smart Cards

**Qi Jiang[1], Jianfeng Ma[1], Guangsong Li[2], Zhuo Ma[1]**

[1] *School of Computer Science and Technology, Xidian University*
*2nd South Taibai Road, Xi'an 710071, P.R. China*
*e-mail: jiangqixdu@gmail.com*

[2] *Department of Information Research, Information Engineering University,*
*Zhengzhou, P.R. China*

crossref http://dx.doi.org/10.5755/j01.itc.42.2.2079

**Abstract**. Authentication is one of the fundamental mechanisms to enable a legitimate user to log into a remote server in an insecure environment. Many authentication protocols have been proposed in the literature for preventing unauthorized parties from access resources. Recently, Chen et al. proposed a password-based remote user authentication and key agreement scheme using common storage devices, such as USB sticks. They claimed that the scheme can withstand off-line dictionary attacks even if the authentication information stored in the device is obtained by the adversary. However, we observe that Chen et al.'s scheme is insecure against off-line dictionary attacks in this case. To remedy this security flaw, we propose an improved authentication protocol without using smart cards. Compared with the previous schemes, our scheme not only provides more security guarantees, but also is more efficient both in computation and communication cost..

**Keywords**: mutual authentication, password, remote access, off-line dictionary attack.

## 1. Introduction

Authentication between the users and the server is essential to prevent unauthorized service and resource access and remove the potential security threats over the insecure networks. Password-based user authentication is one of the simplest, most convenient authentication schemes, in which, password is the secret data which the user and the server agree in advance and is used to verify the identity of the two parties. Since Lamport [1] proposed the first password authentication scheme over an insecure channel in 1981, password authentication (PA) protocols have been extensively investigated in the literature [1-31]. They can be classified into three classes: the password-only PA protocols, the dedicated device-aided PA protocols (e.g., smart card based PA) and the memory device-aided PA protocols [29].

In the password-only PA protocols, no extra devices are used, the user only requires presenting the password that can be easily memorized by the human beings, and the server maintains a password file to verify the user's authentication request. While widely deployed, the maintenance of the password file introduces a risk of tampering and maintenance cost.

In order to reduce the risk and maintenance cost, many dedicated device-aided PA protocols have been proposed. In these schemes, a user requires to remember a short password and to hold a specialized device (i.e., a smart card) to complete a successful authentication. The authentication information is stored in the smart card, which is issued by the server and is accessed via inserting the smart card into the specialized readers. Although smart cards come with a tamper-resistant property, the content of the smart card can be extracted by monitoring its power consumption and analyzing the leaked information [32, 33]. Therefore, the use of tamper-resistant devices does not guarantee that an authentication scheme is secure against all risks; most of them are still subject to some traditional attacks if the smart card is stolen. In addition, the required infrastructure for smart card-based schemes, such as the cards and readers, greatly increases the cost of deployment.

To reduce the deployment cost, the memory device-aided PA protocols come into being. Each user uses a common memory device without the property of tamper resistance, such as universal serial bus (USB) sticks, portable HDDs, mobile phones, PDAs and PCs, to store some authentication information

issued by the server. This kind of PA scheme is also called PA without using smart cards. It often saves the deployment cost in the real world due to the low cost of the devices used and less dependence on the supported infrastructure.

In this work, we investigate the third kind of PA schemes, i.e., the PA without using smart cards. The authentication information issued from the server is stored on a common storage device without tamper-resistance. Once the device is stolen, all the information in it may be revealed. Our goal is to ensure the security of the scheme in the case that the memory device is stolen but the password of the device owner is unknown to the adversary. More specifically, even if the memory device is stolen, the adversary will not be able to mount off-line dictionary attacks.

In 2009, Rhee et al. [24] pointed out that the schemes using smart cards cannot be directly converted into schemes using a common storage device. Specifically, they analyzed the security of Fan et al.'s [7] and Khan et al.'s [8] authentication schemes when the tamper-resistant property is eliminated from the smart card. They showed that Fan et al.'s authentication scheme [7] is vulnerable to impersonation attacks, and that Khan et al.'s authentication scheme [8] becomes vulnerable to impersonation attacks and off-line dictionary attacks, in this case. Then they proposed a practical and secure user authentication scheme using the common storage device based on Khan et al.'s scheme using smart cards. They claimed that their scheme achieves mutual authentication and enjoys all advantages of authentication schemes using smart cards. However, Rhee et al.'s scheme is insecure against impersonation and man-in-the-middle attacks [25, 26].

Recently, Chen et al. [28] found that Rhee et al.'s scheme is insecure because there are much redundant information and the identification-related information is absent in the login request. Then they proposed the enhanced version of Rhee et al.'s scheme based on the computational Diffie-Hellman Problem [34]. They claimed that their scheme provides mutual authentication and is secure against off-line dictionary and well-known on-line attacks, such as replay attacks, forgery attacks, and impersonation attacks.

However, we identify that Chen et al.'s scheme is insecure against off-line dictionary attacks if the device is stolen. To fix this security problem, we propose an improved PA scheme without using smart cards. Compared with the previous schemes, our scheme not only provides more security guarantees, but also is still efficient both in computation and communication cost. Therefore, our scheme is more suitable for practical applications.

The rest of the paper is organized as follows. Section 2 gives a brief review and cryptanalysis of Chen et al.'s scheme. We present our proposed scheme and its analysis in Sections 3 and 4, respectively. Finally, we give the conclusion in Section 5.

## 2. Review of Chen et al.'s Scheme [28]

### 2.1. Description

The notations used throughout the paper are defined as follows.

$ID_i$: the user $U_i$'s identity;

$PW_i$: the user $U_i$'s password;

$x$, $X$: the remote server $S$'s secret key and public key;

$p$, $q$: two large prime numbers such that $p = 2q + 1$;

$g$: a generator with order $q$ in $GF(p)$;

$H$: a secure one-way hash function;

$T$: timestamp;

$\Delta T$: maximum transmission delay;

$Z_q$: a ring of integers modulo $q$;

$Z_q^*$: multiplicative group of $Z_q$;

$||$: the concatenation operation.

We now briefly review Chen et al.'s scheme [28], which consists of three phases: registration, login and authentication, and one activity: password change, as is shown in Figure 1.

#### 2.1.1. Registration Phase

In this phase, the remote server $S$ selects large prime numbers $p$ and $q$ such that $p = 2q + 1$. The server also chooses a generator $g$ of $Z_q^*$, its secret key $x \in Z_q^*$, and a secure one-way hash function $H$. When $U_i$ wants to become a new legal user, he proceeds with the following steps through a secure channel:

(1) $U_i$ selects a unique identity $ID_i$ and a password $PW_i$, and submits them to $S$.

(2) Upon receiving the registration information, $S$ computes $Y_i = H(ID_i)^{x+PW_i} \bmod p$. Then $S$ sends the authentication information $\{Y_i, H, p, q\}$ back to $U_i$.

(3) After receiving the authentication information, $U_i$ stores it locally on his memory device, i.e., his USB drive.

Since the authentication information issued from the server is stored on a device without tamper-resistance, it is possible that the information may be altered carelessly or maliciously. In these cases, a timeout threshold is set to ensure the correctness of the authentication information. The user has to re-register to obtain the new authentication information when he does not receive $S$'s response in the threshold time.

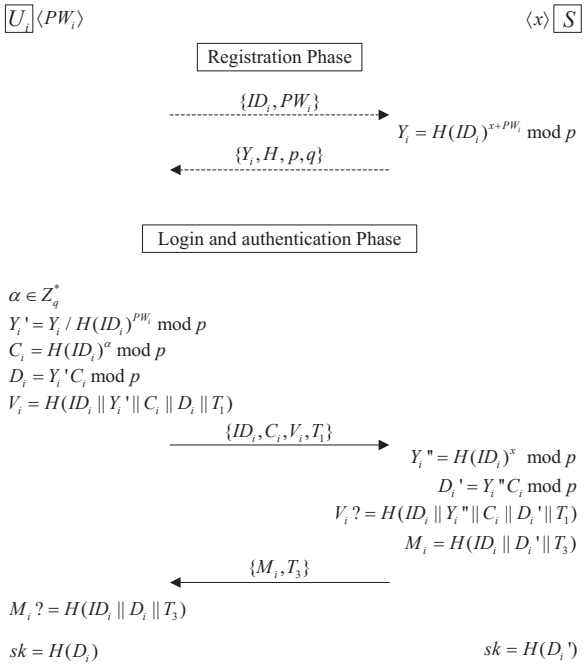


**Figure 1.** Chen et al.'s scheme [28]

2.1.2. Login and Authentication Phase

When a legal user $U_i$ wants to access system resources provided by $S$, he first retrieves the authentication information stored on his USB stick, and inputs his password $PW_i$. Then the login and authentication procedure proceeds as follows.

(1) $U_i$ chooses a random number $\alpha \in Z_q^*$, and computes $Y_i' = Y_i / H(ID_i)^{PW_i} \bmod p$, $C_i = H(ID_i)^{\alpha} \bmod p$, $D_i = Y_i' C_i \bmod p$, and $V_i = H(ID_i \| Y_i' \| C_i \| D_i \| T_1)$, where $T_1$ is the current time of $U_i$. Next, $U_i$ sends his login request $\{ID_i, C_i, V_i, T_1\}$ to $S$.

(2) On receiving the login request from $U_i$, $S$ checks the validity of $ID_i$ and $(T_2 - T_1) < \Delta T$, where $T_2$ is the current time of $S$. If either does not hold, $S$ drops the request and terminates the session. Otherwise, $S$ computes $Y_i'' = H(ID_i)^x \bmod p$ and $D_i' = Y_i'' C_i \bmod p$, and then compares $V_i$ with $H(ID_i \| Y_i'' \| C_i \| D_i' \| T_1)$. If they are not equal, $S$ rejects the request. Otherwise, $S$ authenticates $U_i$ and the login request is accepted. Then, $S$ computes $M_i = H(ID_i \| D_i' \| T_3)$, where $T_3$ is the current time of $S$, and sends $\{M_i, T_3\}$ to $U_i$.

(3) Upon receiving the message from $S$, $U_i$ checks if $T_3$ is valid and $M_i$ is equal to $H(ID_i \| D_i \| T_3)$. If both hold, $S$ is authenticated and mutual authentication between $S$ and $U_i$ is achieved. Otherwise, $S$ is not authenticated.

(4) After the mutual authentication has finished, $U_i$ and $S$ compute the symmetric session key $sk = H(D_i)_{user-side} = H(D_i')_{server-side}$ and use the key to establish a secure communication channel.

2.1.3. Password Change Activity

If a legal user $U_i$ wants to change his password, $U_i$ selects the new identity $ID_i^*$ and password $PW_i^*$, goes back to the registration phase, and re-obtains his new authentication information from $S$.

### 2.2. Weaknesses of Chen et al.'s Scheme

The authentication information issued from the server is stored on a common storage device without tamper-resistance. Once the device is stolen, all the information in it may be exposed. Specifically, to analyze the security of password based authentication without using the smart card, we suppose that an adversary $A$ has the following capabilities [10]. First, the adversary has total control over the communication path between the server and users. That is, the adversary may intercept, insert, delete, or modify any message through the insecure channel. Second, the adversary may extract the secret parameters from the common memory device. Third, the adversary may know passwords and all the information stored in the common memory device of all the users except those of the user who is under attack from the adversary.

Passwords play a critical role in the PA schemes. Password protection is the key point to ensure the security of the scheme. Since the users tend to choose easily remembered short passwords for their convenience, the sample space of passwords may be small enough to be enumerated by an adversary. That is, these passwords are potentially vulnerable to dictionary attacks. The dictionary attacks can be further classified into three classes [31].

Detectable on-line dictionary attacks: an adversary tries to verify the correctness of a guessed password in an on-line manner by observing the response from the server. In this case, a failed guess can be detected and logged by the server.

Undetectable on-line dictionary attacks: an adversary also attempts to verify a password guess in an on-line manner. However, a failed guess cannot be detected or logged by the server. In this case, the server cannot distinguish between an honest request of a legitimate user and a malicious request of an adversary.

Off-line dictionary attacks: an adversary guesses a password and verifies his guess in an off-line manner by using the eavesdropped authentication messages and the authentication information stored in the device.

Most of the password-based authentication schemes are insecure against detectable on-line dictionary attacks. In general, the possible number of password guesses is limited to prevent detectable on-line dictionary attack in these schemes; an adversary

can hardly succeed to find the correct password within the limitation. All the password-based user authentication schemes should be designed to prevent undetectable on-line and off-line dictionary attacks. It is critical to ensure that even if the device is stolen, the adversary will not be able to mount an off-line dictionary attack.

In [28], Chen et al. claimed that the scheme is secure against off-line dictionary attacks even if the common storage device is stolen. However, on the contrary, we observe that it is not true. $A$ can mount the off-line dictionary attacks. The following assumptions are made.

First, an adversary $A$ eavesdrops the messages transmitted between the user and the server and stores the message $\{ID_i, C_i, V_i, T_1\}$ into the database according to $ID_i$, where $Y_i' = Y_i / H(ID_i)^{PW_i} \bmod p$, $C_i = H(ID_i)^{\alpha} \bmod p$, $D_i = Y_i' C_i \bmod p$, and $V_i = H(ID_i \| Y_i' \| C_i \| D_i \| T_1)$.

Second, $A$ has stolen $U_i$'s device and extracted the stored authentication information $\{Y_i, H, p, q\}$, where $Y_i = H(ID_i)^{x+PW_i} \bmod p$.

Then $A$ can find out $U_i$'s password $PW_i$ through the following procedure.

1) $A$ retrieves a message $\{ID_i, C_i, V_i, T_1\}$ from the database according to the identity $ID_i$.

2) $A$ guesses a candidate password $PW_i^*$, and computes $Y_i^* = Y_i / H(ID_i)^{PW_i^*} \bmod p$ and $D_i^* = Y_i^* C_i \bmod p$.

3) $A$ checks whether $H(ID_i \| Y_i^* \| C_i \| D_i^* \| T_1)$ and $V_i$ are equal. If they are equal, $A$ has found the correct password. Otherwise, $A$ repeats the steps 2) and 3) until the correct password is found.

It is easy to see that the adversary can obtain the password through the above procedure. Therefore, Chen et al.'s scheme is vulnerable to the off-line dictionary attack with the stolen device.

## 3. Our Improved Scheme

Our improved scheme consists of 5 ph ases: initialization, registration, login, authentication, and password change, as is shown in Figure 2.

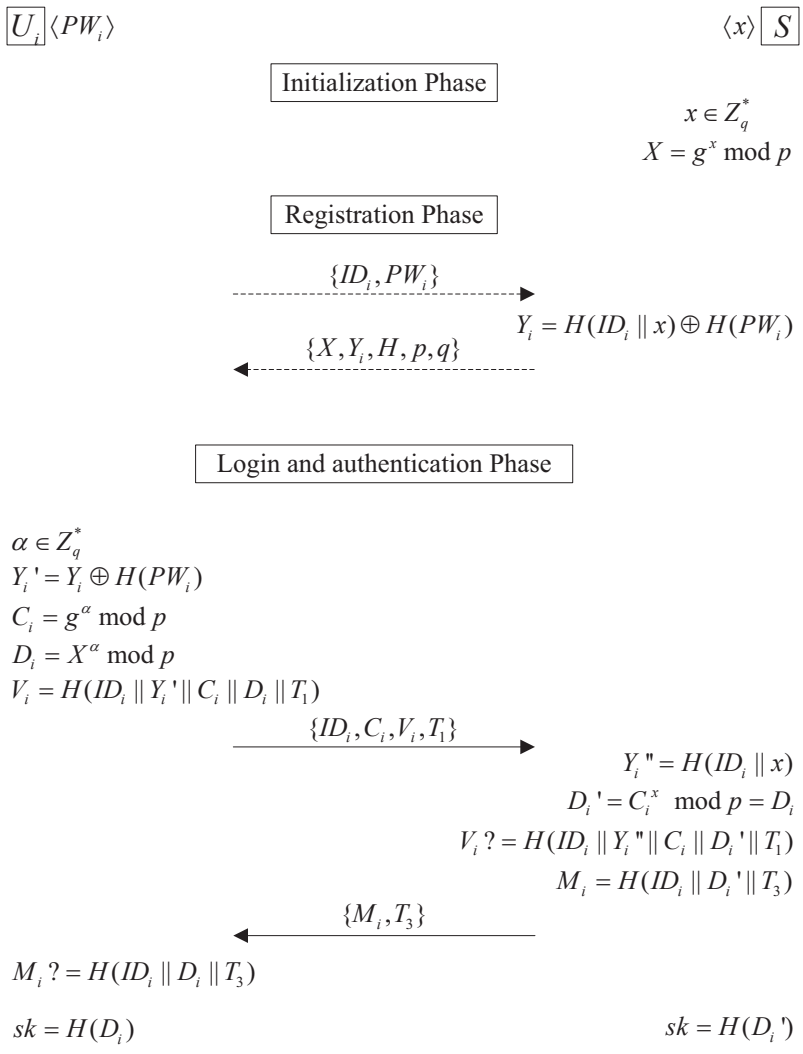


**Figure 2.** Our improved scheme

### 3.1. Initialization Phase

In this phase, the remote server $S$ selects large prime numbers $p$ and $q$ such that $p = 2q + 1$. The server also chooses a generator $g$ of $Z_q^*$, its secret key $x \in Z_q^*$, a secure one-way hash function $H$, and computes its public key $X = g^x \bmod p$.

### 3.2. Registration Phase

When $U_i$ wants to become a new legal user, he proceeds with the following steps:

(1) $U_i$ selects a unique identity $ID_i$ and a password $PW_i$, and submits them to $S$ through a secure channel.

(2) Upon receiving the registration information, $S$ computes $Y_i = H(ID_i \| x) \oplus H(PW_i)$. Then $S$ sends the authentication information $\{X, Y_i, H, p, q\}$ to $U_i$ via the secure channel. In our scheme, $S$ only maintains the ID table which includes $U_i$'s identity $ID_i$.

(3) After receiving the authentication information, $U_i$ stores it locally on his memory device, e.g., his USB stick.

As in Chen et al.'s scheme [28], a timeout threshold is set.

### 3.3. Login and Authentication Phase

When a legal user $U_i$ wants to access system resources provided by $S$, he first retrieves the authentication information stored on his USB stick, and inputs his password $PW_i$. Then the login and authentication procedure proceeds as follows.

(1) $U_i$ chooses a random number $\alpha \in Z_q^*$, and computes $Y_i' = Y_i \oplus H(PW_i)$, $C_i = g^\alpha \bmod p$, $D_i = X^\alpha \bmod p$, and $V_i = H(ID_i \| Y_i' \| C_i \| D_i \| T_1)$, where $T_1$ is the current time of $U_i$. Next, $U_i$ sends his login request $\{ID_i, C_i, V_i, T_1\}$ to $S$.

(2) On receiving the login request from $U_i$, $S$ checks the validity of $ID_i$ according to the ID table and $(T_2 - T_1) < \Delta T$, where $T_2$ is the current time of $S$. If either does not hold, $S$ drops the request and terminates the session. Otherwise, $S$ computes $Y_i'' = H(ID_i \| x)$ and $D_i' = C_i^x \bmod p = g^{x\alpha} \bmod p = X^\alpha \bmod p = D_i$, and then compares $V_i$ with $H(ID_i \| Y_i'' \| C_i \| D_i' \| T_1)$. If they are not equal, $S$ rejects the request. Otherwise, $S$ authenticates $U_i$ and the login request is accepted. Then, $S$ computes $M_i = H(ID_i \| D_i' \| T_3)$, where $T_3$ is the current time of $S$, and sends $\{M_i, T_3\}$ to $U_i$.

(3) Upon receiving the message from $S$, $U_i$ checks if $T_3$ is valid and $M_i$ is equal to $H(ID_i \| D_i \| T_3)$. If both hold, $S$ is authenticated and mutual authentication between $S$ and $U_i$ is achieved. Otherwise, $S$ is not authenticated.

(4) After the mutual authentication has finished, $U_i$ and $S$ compute the symmetric session key $sk = H(D_i)_{user-side} = H(D_i')_{server-side}$ and use the key to establish a secure communication channel.

### 3.4. Password Change Activity

If a legal user $U_i$ wants to change his password, $U_i$ selects the new password $PW_i^*$, then computes $Y_i^* = Y_i \oplus H(PW_i) \oplus H(PW_i^*)$ and replaces $Y_i$ with $Y_i^*$.

## 4. Security and Efficiency Analysis

### 4.1. Security Analysis

We first introduce some hard problems, which form the basis of security of our improved scheme.

Discrete logarithm problem (DLP) [35]: Given two elements $g$ and $h$, it is computationally infeasible to find an integer $a \in Z_q^*$ such that $h = g^a \bmod p$ whenever such an integer exists.

Computational Diffie-Hellman Problem (CDHP) [34]: Given $g$, $g^a$, $g^b$ for $a, b \in Z_q^*$, it is computationally intractable to compute $g^{ab} \bmod p$.

One-way property: for any given $h$, it is computationally infeasible to find $y$ so that $h = H(y)$.

Under the attack model defined in Section 2.2, even if the authentication information stored in the memory device is exposed, our improved scheme can resist various malicious attacks, including stolen verifier attacks, on-line and off-line dictionary attacks, replay attacks, user impersonation attacks and server impersonation attacks. Our scheme also achieves mutual authentication and establishes a secure channel between the user and the server.

Following the analysis in [28], the security analysis of our improved scheme is presented as follows.

#### 4.1.1. Security of the Server *S*'s Secret Key *x* [28]

As in Chen et al.'s scheme, only the server $S$ has the knowledge of the secret key $x$. An adversary $A$ may eavesdrop the network traffic and collect the authentication messages, i.e., $U_i$'s login request $\{ID_i, C_i, V_i, T_1\}$ and $S$'s response $\{M_i, T_3\}$. However,

$A$ cannot recover $x$ from $C_i$, since $C_i = g^{\alpha} \bmod p$ contains no information of $x$. Besides, owing to the irreversibility of hash function $H$, it is impossible for $A$ to obtain $x$ based on $V_i$ and $M_i$. Hence, there is no way for $A$ to obtain $S$'s secret key $x$ using the eavesdropped authentication messages.

Suppose $A$ steals $U_i$'s authentication information $\{X, Y_i, H, p, q\}$, and tries to retrieve $x$ from $X = g^x \bmod p$ and $Y_i = H(ID_i \| x) \oplus H(PW_i)$. To retrieve $x$ from $X = g^x \bmod p$, he faces to break the DLP problem. To retrieve $x$ from $Y_i = H(ID_i \| x) \oplus H(PW_i)$, even for a malicious user $U_i$ who can extract $Y_i' = H(ID_i \| x)$, it is impossible to get $x$, since $H$ is a secure one-way hash function.

4.1.2. Stolen Verifier Attacks

Stolen verifier attacks mean that an adversary stealing the password verifier (e.g., plaintext passwords, hashed passwords) stored in verification table from the server can use it directly to impersonate as a legitimate user during the authentication phase [17]. In our scheme, the server $S$ only stores the user identity in an ID table. Even if the adversary stole the identity from the ID table, he or she has no way of obtaining $Y_i' = H(ID_i \| x)$ to compute a valid login request without the secret key $x$. Clearly, no sensitive information, such as passwords or verifiers derived from $PW_i$, is stored in the verification table. Therefore, our scheme is secure against the stolen verifier attack.

4.1.3. Undetectable On-line Dictionary Attacks

In our scheme, on-line dictionary attack will be detected immediately. Suppose that an adversary attempts to find the password of a legal user. He would guess a possible password to perform the login and authentication phase. According to the scheme, the server can detect the attack by confirming whether $V_i$ is equal to $H(ID_i \| Y_i'' \| C_i \| D_i' \| T_1)$ or not. Generally, when the third attempt goes wrong, the identity attacked by the adversary would be locked and no further attempts would be allowed. As a result, our scheme does not suffer from the undetectable on-line dictionary attacks.

4.1.4. Off-line Dictionary Attacks

There is no way for an adversary $A$ to retrieve $U_i$'s password $PW_i$ based on the eavesdropped authentication messages, such as $U_i$'s login request $\{ID_i, C_i, V_i, T_1\}$ and $S$'s response $\{M_i, T_3\}$, because these messages do not contain any information of $PW_i$.

Suppose that $A$ steals $U_i$'s authentication information $\{X, Y_i, H, p, q\}$, and tries to retrieve $PW_i$ from $Y_i = H(ID_i \| x) \oplus H(PW_i)$. Since there is no information that can be used as a verifier, it is impossible for $A$ to guess $PW_i$.

Further assume that $U_i$'s authentication information $\{X, Y_i, H, p, q\}$ is stolen by an adversary $A$, and $A$ has also collected the authentication messages $\{ID_i, C_i, V_i, T_1\}$ and $\{M_i, T_3\}$.

In the improved scheme, for each guessed password $PW_i^*$, $A$ can compute $Y_i^* = Y_i \oplus PW_i^*$. Since $V_i = H(ID_i \| Y_i' \| C_i \| D_i \| T_1)$ and $D_i = X^{\alpha} \bmod p$, $A$ has to compute $D_i$ to verify whether $Y_i^*$ is valid. However, due to the hardness of the CDHP, it is impossible for $A$ to compute $D_i = X^{\alpha} \bmod p$ with $g$, $C_i = g^{\alpha} \bmod p$ and $X = g^x \bmod p$. Then except a negligible probability (guess the value $\alpha$ or obtain the server's secret key $x$), $A$ cannot verify whether $Y_i^*$ is valid without on-line interaction with the server. Due to hardness of DLP, it is impossible for $A$ to compute $\alpha$ with $C_i = g^{\alpha} \bmod p$. In addition, as discussed in Section 4.1.1, it is impossible for $A$ to recover the Server $S$'s secret key $x$ from the user's authentication information or eavesdropped authentication messages.

As a result, our improved scheme is secure against off-line dictionary attacks.

4.1.5. Replay Attacks

Assume that an adversary $A$ pretends to be a user $U_i$ or the server $S$ by replaying the eavesdropped messages, such as $U_i$'s login request $\{ID_i, C_i, V_i, T_1\}$ or $S$'s response $\{M_i, T_3\}$. For $U_i$'s login request $\{ID_i, C_i, V_i, T_1\}$, $S$ can easily detect a replay attack by checking the timestamp $T_1$. For $S$'s response $\{M_i, T_3\}$, $U_i$ can easily detect a replay attack by checking the timestamp $T_3$. Therefore, the adversary cannot circumvent the timestamp checking and complete the authentication phase.

The man in the middle attack, a special case of replay attack, can also be detected by checking the timestamp [28].

4.1.6. User Impersonation Attacks

Assume that there is an adversary $A$ who wants to impersonate a valid user $U_i$ to the server $S$. $A$ has to compute $Y_i' = H(ID_i \| x)$ to prove its legitimacy. However, $A$ is unable to compute $Y_i' = H(ID_i \| x)$ directly if he does have secret key $x$, since $H$ is a secure hash function. In addition, $A$ still cannot get $Y_i' = H(ID_i \| x)$ even if he could get the stored data

$Y_i = H(ID_i \| x) \oplus H(PW_i)$ in the memory device, since it is protected by the password. $A$ cannot compute the correct value of $V_i$. As a result, our improved scheme is secure against user impersonation attacks.

4.1.7. Server Impersonation Attacks

If an adversary $A$ wants to impersonate $S$ and spoof user $U_i$, $A$ needs to generate a valid response $\{M_i, T_3\}$, where $M_i = H(ID_i \| D_i' \| T_3)$. However, $A$ has no knowledge of the server's secret key $x$ and cannot recover it from the eavesdropped communication messages and authentication information as presented in Section 4.1.1. $A$ cannot correctly compute $D_i'$ and generate a valid response. Therefore, our improved scheme is secure against server impersonation attacks.

4.1.8. Mutual Authentication

Mutual authentication means that not only the server but also a user can verify the identity of the communicating party. It is a critical requirement for most real-world applications where one's private information should not be released to anyone until mutual confidence is established.

Our scheme provides a mechanism that allows the user and the server to authenticate each other. On the one hand, from the discussion in Section 4.1.6, we can see that only the legitimate user with the correct password and authentication information can pass the verification of the server. On the other hand, from Section 4.1.7, we can observe that only the server with correct secret key can pass the verification of the user. As a result, the scheme achieves mutual authentication between a legal user and the server.

4.1.9. Secure Channel

After a user logins the remote server successfully, another criticasl security issue arises, that is, to ensure data confidentiality and integrity during transmission. The major concern is to safeguard the confidential data from exposure, modification or deletion during their transmission.

In our scheme, after mutual authentication has completed, both $U_i$ and $S$ can compute the session key $sk$, which is used to protect the subsequent communications. The session key is generated from $g^{x\alpha} \bmod p$, which is unknown to any other parties except $U_i$ and $S$. All of the data communications will be encrypted by the session key, and no adversary can eavesdrop, modify, or delete the transmitting data.

In addition, the session key $sk$ is generated independently and is different for each login session. Hence, even if some session keys are revealed, the previous and future session keys are still secure. The session key in our scheme will be invalid whenever the session between the user and the server goes to the end. That is, the key will be revoked and cannot be used any more when its period of usage expires. When the user enters the system again, a new session key will be established to protect his information during the current session. Therefore, a secure channel is established in our scheme.

4.1.10. Functionality Comparisons

The functionality comparisons between our scheme and two previous schemes are summarized in Table 1. Our scheme achieves more security functionalities than these schemes in [24, 28].

**Table 1.** Functionality Comparison

| **Functionality** | **Rhee et al.'s scheme [24]** | **Chen et al.'s scheme [28]** | **Our scheme** |
|---|---|---|---|
| Resisting off-line dictionary attacks | No | No | Yes |
| Resisting stolen authentication information attacks | No | Yes | Yes |
| Undetectable On-line Dictionary Attacks | Yes | Yes | Yes |
| Stolen Verifier Attacks | Yes | Yes | Yes |
| Resisting man in the middle attacks | No | Yes | Yes |
| Resisting replay attacks | No | Yes | Yes |
| Resisting user impersonation attacks | No | Yes | Yes |
| Resisting server impersonation attacks | No | Yes | Yes |
| Mutual authentication | No | Yes | Yes |
| Secure channel | No | Yes | Yes |

**4.2. Efficiency Analysis**

Furthermore, we evaluate the efficiency of our scheme in terms of computation and communication cost. Table 2 summarizes the computation and communication cost comparison between our scheme and the previous schemes, respectively. The following notations are used in Table 2.

$t_e$ : the time complexity of exponentiation operation.

$t_m$ : the time complexity of multiplication/division operation.

$t_h$ : the time complexity of hash operation.

Note that we ignore the computational complexity of other operations, such as exclusive-OR and comparison, because they require very limited computation resources.

As is in [28], we also assume that the length of the identity and the timestamp are 64 bits, the length of

the prime number $p$ is 128 bits, and the one-way hash function is SHA-1 with 160 bits.

From Table 2, it is obvious to see that our scheme is more efficient than these schemes in [24, 28] in computation cost. As for communication cost, a successful user authentication in our scheme requires two message exchanges between the user and the server. As a result, our scheme is as efficient as these schemes in [24, 28] in round efficiency. The number of communication bits of Rhee et al.'s scheme is 832(=64+128+128+128+128+64+128+64) bits, and that of Chen et al.'s scheme and our improved scheme is 640(=64+128+160+64+160+64) bits. It is demonstrated that our scheme is as efficient as Chen et al.'s scheme and is more efficient than Rhee et al.'s scheme in communication cost.

Above all, our proposed scheme exceeds these schemes in [24, 28] regarding both security and performance. Therefore, it is a s ecure and efficient authentication scheme which is suitable for practical applications.

**Table 2.** Comparison

| Cost\Scheme | | | Rhee et al.'s scheme [24] | Chen et al.'s scheme[28] | Our scheme |
|---|---|---|---|---|---|
| Computation | Registration | User side | - | - | - |
| | | Server side | $2t_e + t_m + t_h$ | $t_e + t_h$ | $2t_h$ |
| | Authentication | User side | $4t_e + 2t_m + 3t_h$ | $2t_e + 2t_m + 4t_h$ | $2t_e + 4t_h$ |
| | | Server side | $3t_e + t_m + 2t_h$ | $t_e + t_m + 4t_h$ | $t_e + 4t_h$ |
| Communication | Rounds | | 2 | 2 | 2 |
| | Bits | | 832 | 640 | 640 |

## 5. Conclusions

We have analyzed a password based authentication scheme without using smart cards and pointed out that the scheme suffers from off-line dictionary attacks if the authentication stored in the memory device is exposed. In order to overcome the defects in these schemes, we have proposed an improved authentication scheme based on CDHP. We have demonstrated that our improved scheme can withstand various attacks and achieves mutual authentication between the user and the server. Compared with the previous schemes, our scheme not only provides more security guarantees, but also is more efficient both in computation and communication cost. Therefore, our scheme is secure and efficient for practical applications.

### Acknowledgments

This work is supported by the Key Program of NSFC-Guangdong Union Foundation (Program No. U1135002), Major national S&T program (2011ZX03005-002), National Natural Science Foundation of China (Program No. 61072066, 61173135, 61100233, 61100230, 61100153), Natural Science Basic Research Plan in Shaanxi Province of China (Program No. 2012JQ8043), Fundamental Research Funds for the Central Universities (Program No. JY10000903001, K50511030004). The authors would like to thank the anonymous reviewers and the editor for their constructive comments that have helped us to improve this paper.

## References

[1] **L. Lamport**. Password authentication with insecure communication. *Communications of the ACM*, 1981, Vol. 24, No. 11, pp. 770–772.

[2] **M. S. Hwang, L. H. Li**. A new remote user authentication scheme using smart cards. In: *IEEE Transactions on Consumer Electronics*, 2000, Vol. 46, No. 1, pp. 28–30.

[3] **H. M. Sun**. An efficient remote use authentication scheme using smart cards. In: *IEEE Transactions on Consumer Electronics*, 2000, Vol. 46, No. 4, pp. 958-961.

[4] **H. Y. Chien, J. K. Jan, Y. M. Tseng**. An efficient and practical solution to remote authentication: smart card. *Computers and Security*, 2002, Vol. 21, No. 4, pp. 372–375.

[5] **C. C. Yang, R. C. Wang**. Cryptanalysis of a u ser friendly remote authentication scheme with smart cards. *Computers and Security*, 2004, Vol. 23, No.5, pp. 425–427.

[6] **W. S. Juang**. Efficient password authenticated key agreement using smart cards. *Computers & Security*, 2004, Vol. 23, No. 2, pp. 167–173.

[7] **C. I. Fan, Y. C. Chan, Z. K. Zhang**. Robust remote authentication scheme with smart cards. *Computers & Security*, 2005, Vol. 24, No. 8, pp. 619–628.

[8] **M. K. Khan, J. Zhang.** Improving the security of 'a flexible biometrics remote user authentication scheme'. *Computer Standards & Interfaces*, 2007, Vol. 29, No. 1, pp. 82–85.

[9] **X. M. Wang, W. F. Zhang, J. S. Zhang, M. K. Khan**. Cryptanalysis and improvement on two efficient remote user authentication scheme using smart cards. *Computer Standards & Interfaces*, 2007, Vol. 29, No. 5, pp. 507–512.

[10] **G. Yang, D. Wong, H. Wang, X. Deng**. Two-factor mutual authentication based on smart cards and

passwords. *Journal of Computer and System Sciences*, 2008, Vol. 74, No. 7, pp. 1160–1172.

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

[12] **K. H. Yeh, C. H. Su, N. W. Lo, Y. Li, Y. X. Hung**. Two robust remote user authentication protocols using smart cards. *Journal of Systems and Software*, 2010, Vol. 83, No. 12, pp. 2556–2565.

[13] **T. H. Chen, H. C. Hsiang, W. K. Shih**. Security enhancement on an improvement on two remote user authentication schemes using smart cards. *Future Generation Computer Systems*, 2011, Vol. 27, No. 4, pp. 377-380.

[14] **M. L. Das, A. Saxena, V. P. Gulati**. A dynamic ID-based remote user authentication scheme. In: *IEEE Transactions on Consumer Electronics*, 2004, Vol. 50, No. 2, pp. 629–631.

[15] **Y. Y. Wang, J. Y. Liu, F. X. Xiao, J. Dan**. A more efficient and secure dynamic ID based remote user authentication scheme. *Computer Communications*, 2009, Vol. 32, No. 4, pp. 583–585.

[16] **M. K. Khan, S. K. Kim, K. Alghathbar**. Cryptanalysis and security enhancement of a "more efficient & secure dynamic ID- based remote user authentication scheme". *Computer Communications*, 2011, Vol. 34, pp. 305–309.

[17] **R. Madhusudhan, R. C. Mittal**. Dynamic ID-based remote user password authentication schemes using smart cards: a review. *Journal of Network and Computer Applications*, 2012, Vol. 35, No. 4, July 2012, pp. 1235–1248, DOI:10.1016/j.jnca.2012.01.007.

[18] **C. T. Li**. Secure smart card based password authentication scheme with user anonymity. *Information Technology and Control*, 2011, Vol. 40, No. 2, pp. 157–162.

[19] **C. T. Li, C. C. Lee**. A robust remote user authentication scheme using smart card. *Information Technology and Control*, 2011, Vol. 40, No. 3, pp. 236–245.

[20] **D. He, J. Chen, R. Zhang**. A more secure authentication scheme for telecare medicine information systems. *Journal of Medical Systems*. 2012, Vol. 36, No. 3, pp. 1989-1995, DOI: 10.1007/s10916-011-9658-5.

[21] **Q. Jiang, J. Ma, G. Li, L. Yang**. An Enhanced Authentication Scheme with Anonymity for Roaming Service in Global Mobility Networks. *Wireless Personal Communications*, 2012, Vol. 68, No. 4, pp. 1477-1491. DOI: 10.1007/s11277-012-0535-4.

[22] **D. He, J. Chen, Y. Chen**. A secure mutual authentication scheme for session initiation protocol using elliptic curve cryptography. *Security and Communication Networks*, 2012, Vol. 5, No. 12, pp. 1423–1429, DOI: 10.1002/sec.506.

[23] **W. C. Ku**. A hash-based strong-password authentication scheme without using smart-cards. *ACM Operating Systems Review*, 2004, Vol. 38, No. 1, pp. 29-34.

[24] **H. S. Rhee, J. O. Kwon, D. H. Lee**. A remote user authentication scheme without using smart cards. *Computer Standards & Interfaces*, 2009, Vol. 31, No. 1, pp. 6–13.

[25] **Z. Tan**. Security analysis of two password authentication schemes. Mobile Business. *International Conference on, 296–300, 2009 Eighth International Conference on Mobile Business*, 2009.

[26] **W. G. Shieh, W. B. Horng**. Security analysis and improvement of the remote user authentication scheme without using smart cards. *ICIC Express Letters*, 2010, Vol. 4, No. 6, pp. 2431-2436.

[27] **K. H. Yeh, N. W. Lo**. A novel remote user authentication scheme for multi-server environment without using smart cards. *International Journal of Innovative Computing, Information and Control*, 2010, Vol. 6, No. 8, pp. 3467-3478.

[28] **B. L. Chen, W. C. Kuo, L. C. Wuu**. A secure password-based remote user authentication scheme without smart cards. *Information Technology and Control*, 2012, Vol. 41, No. 1, pp. 53-59.

[29] **H. Qian, J. Gong, Y. Zhou**. Anonymous password-based key exchange with low resources consumption and better user-friendliness. *Security and Communication Networks*, 2012, Vol. 5, No. 12, pp. 1379–1393, DOI: 10.1002/sec.501.

[30] **Y. Wang**. Password protected smart card and memory stick authentication against off-line dictionary attacks. In: *D. Gritzalis, S. Furnell, and M. Theoharidou (Eds.): SEC 2012, IFIP AICT 376*, 2012, pp. 489-500.

[31] **C. Lv, M. Ma, H. Li, J. Ma, Y. Zhang**. A novel three-party authenticated key exchange protocol using one-time key. *Journal of Network and Computer Applications*, 2013, Vol. 36, No. 1, pp. 498-503, DOI:10.1016/j.jnca.2012.04.006.

[32] **P. Kocher, J. Jaffe, B. Jun**. Differential power analysis. *Advances in Cryptology*, *CRYPTO'99*, 1999, pp. 388–397.

[33] **T. S. Messerges, E. A. Dabbish, R. H. Sloan**. Examining smart card security under the threat of power analysis attacks. *IEEE Transactions on Computers*, 2002, Vol. 51, No. 5, pp. 541–552.

[34] **W. Diffie, M. E. Hellman**. New Directions in Cryptography. In: *IEEE Transactions on Information Theory*, 1976, Vol. 22, No. 6, pp. 644–654.

[35] **T. ElGamal**. A public-key cryptosystem and a signature scheme based on discrete logarithms. In: *IEEE Transactions on Information Theory*, 1985, Vol. 31, No. 4, pp. 469–472.

Received August 2012.