Improvement on A Password Authentication Scheme for Multi-server Environments

Zuowen Tan

1. School of Information Technology, Jiangxi University of Finance and Economics, Nanchang 330032, Jiangxi Province, P.R. China
2. Key Lab of Network Security and Cryptology, School of Mathematics and Computer Science, Fujian Normal University, Fuzhou 350007, Fujian Province, P.R. China
tanzyw@163.com
doi:10.4156/jcit.vol6.issue1.26

Abstract

Since each user needs to have different passwords with different remote servers in the multi-server environment, it is impossible to apply the authentication methods in a single server environment to the multi-server environment. Recently, Ren-Chiun Wang, Wen-Shenq Juang, and Chin-Laung Lei proposed an ID based remote user authentication scheme for multi-server environments. However, we demonstrate that their ID based authentication scheme lacks the authenticity of the Registration Center to the server. Moreover, their ID based authentication scheme does not satisfy perfect forward security. In this paper, an improved multi-server password based smart card authentication scheme is proposed. Our scheme can remove their security flaws. Compared with the previous password based smart card authentication schemes for the multi-server environments, our proposed scheme can hold more security advantages.

Keywords: Smart Card, Multi-server, Authentication

1. Introduction

In 1981, Lamport proposed a password authentication scheme [1] to provide authentication between a user and a remote server. Later, some researchers apply a smart card to password based authentication schemes in a single server [2-5]. However, it is infeasible to construct an efficient password based authentication scheme in the multi-server environment by extending the authentication methods in a single server environment. Since each user needs to have different passwords with different remote servers in the multi-server environment.

In 2001, Li et al. constructed remote password authentication scheme for multi-server architecture [6]. However, the remote password authentication scheme is inefficient. Moreover, it is difficult to apply the scheme to a smart card based password authentication scheme. Since a smart card cannot execute the computation of neural networks in a short time. Tsaur et al. proposed a remote user authentication scheme based on RSA cryptosystem for multi-server environments [7]. Juang proposed a multi-server user authentication and key agreement based on hashing function and symmetric-key cryptosystem [8]. Hu et al. proposed an efficient multi-server password authenticated key agreement scheme using smart cards [9]. Some smart card based password authentication schemes for the multi-server environment are proposed [6,9-14,16,17]. However, many schemes are insecure [9,16,19,15,21]. Liao–Wang's scheme [14] is still vulnerable to insider attacks, masquerade attacks, server spoofing attacks and registration center spoofing attacks. Furthermore, it fails to provide mutual authentication [15,16,21]. Chen, Huang and Chou [16] proposed an improvement on Hsiang and Shih's scheme [15].

Recently, Ren-Chiun Wang, Wen-Shenq Juang, and Chin-Laung Lei also proposed a smart card password authenticated key agreement scheme based on the quadratic residue for multi-server environment (hereafter, we call the scheme WJL-scheme) [21].

In this paper, we demonstrate that WJL-scheme [21] cannot prevent an adversary from impersonating the server. WJL-scheme [21] can not provide perfect forward security. Upon Diffie-Hellman assumptions, we propose an improved multi-server password based smart card authentication scheme to withstand these attacks.
Based on the requirements mentioned in [8], we highlight the attributes of a multi-server password based smart card authentication scheme.

C1. No verification table: No verification or password table is stored in a server.

C2. User friendly: It allows the card holder to change his password freely after assuring the legality of the card holder.

C3. Low computation and communication cost: Due to the power constraints and small flash memory of smart cards, they may not provide a powerful computation capability and high bandwidth.

C4. Mutual authentication: Servers and users can authenticate each other.

C5. Single registration: Users only must register at the registration centre once and can use all the permitted services in the eligible servers.

C6. Session key agreement: Servers and users must negotiate a session key to be used for protecting their subsequent communications. The session key should be determined jointly by the server and the user.

C7. Security: The authentication scheme must be able to resist all kinds of attacks which may suffer from in the real world.

(1) Known-key security. After each run of the scheme, the user and the server can generate a unique secret session key. Each session key is independent of other session keys. Moreover, the compromise of one session key should not lead to compromise of other session keys.

(2) Perfect forward security. If long secret keys of the registration center are compromised, the secrecy of previously established session keys should not be revealed.

(3) Two factor security. If a user's smart card and his password are both stolen, there is no means to prevent the adversary from masquerading as the user. So, the security of password authentication schemes are always discussed in the case that the smart card is stolen.

(4) Resistance against server spoofing. Any adversary (even a server) cannot impersonate a server (other servers) to cheat users.

(5) Resistance against registration center spoofing attack. Any adversary cannot impersonate the registration center to authenticate the user or the servers.

(6) Resistance against impersonation attacks. Any adversary cannot impersonate a user to obtain the services of a server.

(7) Resistance against password guessing attacks.

(8) Resistance against the Denning-Sacco replay attacks [23]. Assume that an attacker has recorded a previous session between Alice and Bob, and compromised the connection key used in that one. If an attacker can succeed in Denning-Sacco replay attacks, then the party Bob would believe he shares a fresh secret key with Alice

(9) Resistance against other attacks such as private key leakage attacks, stolen verifier attacks and man-in-the-middle attacks.

The remainder of this paper is organized as follows. In Section II, we review and analysis of WJL-scheme. In Section III, an improved multi-server password based smart card authentication scheme is proposed. In Section IV, we analyze the proposed multi-server password based smart card authentication scheme. Finally, conclusions will be given in Section V.

2. Review and Analysis of WJL-scheme

2.1. Review of WJL-scheme

WJL-scheme [21] is a password authenticated key agreement scheme using smart cards scheme based on the quadratic residue for multi-server environment. The scheme includes the user registration phase, the login phase, the authentication server and registration center phase and the authentication server and user phase.

First, RC selects a public system parameter \( n = p \times q \) where \( p \) and \( q \) are two large prime numbers and are kept by RC. RC holds a master key \( x \).

In the user registration phase, RC calculates \( U_i \)'s secret key \( SK_i = h(ID_i || x) \) and sends it to \( U_i \). \( U_i \) chooses a password, computes \( SK_i' = SK_i \oplus h(PW) \) and stores \( \{ SK_i', ID_i, h(), n \} \) in the smart card. In
Improvement on A Password Authentication Scheme for Multi-server Environments
Zuowen Tan
Journal of Convergence Information Technology, Volume 6, Number 1. January 2011

the same way, $RC$ sends the secret key $SK_j = h(SID_j||x)$ to $S_j$.

The login phase, the authentication server and registration center phase and the authentication server
and user phase are demonstrated in Fig.1, Fig.2 and Fig.3.

**Figure 1. Login phase**

**Server**

1. Generate $N_j$
2. $Req_i = (ID_i||SK_i||SID_j||N_i)^2 \mod n$
3. Decrypt $ID_i||SK_i||SID_j||N_i$
4. $Res_1, Res_2$
5. $Res_3, Res_4$

**Authentication Server and Registration Center Phase**

1. Generate $N_j$
2. $Req_j = (ID_j||SID_j||N_j)^2 \mod n$
3. Decrypt $ID_j||SID_j||N_j$
4. $Res_1, Res_2$
5. $Res_3, Res_4$

**Figure 2. Authentication Phase 1**

**Figure 3. Authentication Phase 2**
2.2. Analysis of Wang-Juang-Lei’s scheme

Wang-Juang-Lei’s scheme has two flaws: RC cannot authenticate the server and the authentication scheme does not hold perfect forward security.

Firstly, in Wang-Juang-Lei’s scheme, the registration center RC authenticates the server $S_j$ only by the information $SID_j$ sent by the server. So any adversary can choose randomly a $Req_j$ and impersonate the server with an identity $SID$. After RC receives the message sent by the adversary, RC believes that the server with an identity $SID$ is communicating with her.

Secondly, Wang-Juang-Lei’s scheme does not satisfy perfect forward security. If the master key of the RC is disclosed, all the previous session key $SK_j$ will be comprised. This is because

$$SK_j=h(SID_j || x), N_j = SK_j \oplus Req_j,$$  

$$N = h(SK_j || N_j) \oplus Res_1, \quad K_{SC} = h(SK_j || N_j) \oplus K_{SC},$$

Thus, the adversary can obtain the session key

$$SK=h(SID_j || h(N_j)) \oplus R_{ij}.$$

3. A proposed SCHEME

In the section, we propose an improved multi-server password based smart card authentication scheme.

In our scheme, RC holds two master keys, $x$ and $y$. All the identities of users and servers have public information, e.g. $ID_i$ and $SID_j$. The public system parameters include a large field $F_q$ of prime order $q$, a generator $g$ and a secure hash function $h()$.

If the login is the first time, the proposed scheme is composed of four phases: (1) registration phase, (2) login phase, (3) authentication server and registration center phase, (4) authentication server and user phase, and (5) password change phase. If the login is not the first time, the proposed scheme is composed of three phases: (1) registration phase, (2) login phase, and (3) authentication and session key agreement phase, and (4) authentication and password change phase. For the two cases above, registration phase and password change phase are the same.

A. Registration phase

In the registration phase, $S_j$ with identity $SID_j$ obtains its secret information $RS_j=h(SID_j || y)$ from $RC$ through a secure channel.

When the user $U_i$ registers to $RC$, $U_i$ and $RC$ performs the following procedures to accomplish the user $U_i$’s registration.

Step 1. $U_i \rightarrow RC: ID_i$.

Step 2. $RC$ computes $h(ID_i || x)$ and sends it to $U_i$.

Step 3. $U_i$ selects a password $PW_i$ and computes $B_i = h(ID_i || x) \oplus h(ID_i || PW_i)$, stores $\{B_i, ID_i, h()\}$ in the smart card.

If $U_i$ logs the server $S_j$ the first time,

B. Case 1: For the login first time

B.1. Login phase

When $U_i$ logs the server $S_j$ the first time, $U_i$ inserts his smart card and keys his identity $ID_i$ and password $PW_i$. The smart card executes the following steps.

Step 1. Compute

$$T_i=h(ID_i || x) \oplus B_i \oplus h(ID_i || PW_i),$$

Step 2. Choose a random nonce $u$ and calculate

$$N_i = g^u, C_i = h(T_i || SID_j) \oplus N_i, C_2 = h(C_i || N_i).$$
Step 3. $U_i \rightarrow S_j$: \{ID$_i$, SID$_j$, C$_1$, C$_2$\}
Upon receiving the message \{ID$_i$, SID$_j$, C$_1$, C$_2$\}, $S_j$ executes the following steps.
Step 1. Choose a random nonce $N_j$ and calculate
$$V_1 = h(RS_j || ID_i) \oplus N_j, \quad V_2 = h(C_1 || C_2 || V_1 || N_j). \quad (5)$$
Step 2. $S_j \rightarrow RC$: \{ID$_i$, SID$_j$, C$_1$, C$_2$, V$_1$, V$_2$\}

The login phase for the first time is depicted as in Fig. 4.

B.2. Authentication Server and Registration Center Phase
After $RC$ receives the message \{ID$_i$, SID$_j$, C$_1$, C$_2$, V$_1$, V$_2$\}, $RC$ executes the following steps.
Step 1. Compute and verify
$$N_i = h(h(ID_i |x|) || SID_j) \oplus C_1, \quad C_2 = h(C_1 || N). \quad (6)$$
If the equality (6) holds, $RC$ authenticates $U_i$.

\begin{itemize}
  \item Step 2. Compute and check
    $$N_j = h(h(SID_j | y |) || ID_i) \oplus V_1, \quad V_2 = h(C_1 || C_2 || V_1 || N_j). \quad (7)$$
    If the equality (8) holds, $RC$ authenticates $S_j$.
  \item Step 3. Calculate
    $$N = (N_j)^{h(ID_i || y)} \odot h(ID_i || SID_j) \oplus N_j \odot N_j,$$
    $$Z_2 = h(Z_1 \odot N_j || h(ID_i || h(SID_j || y)) || N) \odot N_j, \quad Z_3 = h(Z_2 || h(ID_i || h(SID_j || y))) \odot N. \quad (9)$$
  \item Step 4. $RC \rightarrow S_j$: \{Z$_1$, Z$_2$, Z$_3$\}
When $S_j$ obtains \{Z$_1$, Z$_2$, Z$_3$\}, $S_j$ computes
$$N = h(Z_2 || h(ID_i || RS_j)) \odot Z_3. \quad (11)$$
Then $S_j$ checks if the following holds:
$$Z_2 = h(Z_1 \odot N_j || h(ID_i || RS_j)) || N \odot N_j. \quad (12)$$
\end{itemize}

The authentication server and registration center phase is depicted in Fig. 5 as Authentication Phase.
1. B.3. Authentication Server and User Phase

If the equality (12) holds, \( S_j \) continues the following procedures.

Step 1. Choose \( v \in \mathbb{Z}_q \) and compute \( N_{ji} = g^v \).

Step 2. Compute

\[
V_3 = h((Z_1 \oplus N_j) || ID_i || RS_j) \oplus h((Z_1 \oplus N_j) || SID_i),
\]

(13)

\[
V_4 = h((Z_1 \oplus N_j) || ID_i || V_3 || N_{ji}),
\]

(14)

\[
V_5 = h((Z_1 \oplus N_j) || ID_i || SID_i) \oplus N_{ji},
\]

(15)

Step 3. \( S_j \rightarrow U_i: \{V_3, V_4, V_5\}. \)

Step 4. Compute the session key

\[
SK = h((Z_1 \oplus N_j) || N_v || h(ID_i || RS_j)).
\]

(16)

Upon receiving the message \( \{V_3, V_4, V_5\} \), \( U_i \) executes the following steps.

Step 1. Compute and verify

\[
N_{ji} = V_5 \oplus h((Z_1 \oplus N_j) || ID_i || SID_i),
\]

(17)

\[
V_4 =? h((Z_1 \oplus N_j) || ID_i || V_3 || N_{ji}).
\]

(18)

If the equality (18) holds, \( U_i \) authenticates \( S_j \).

Step 2. Calculate the session key

\[
SK = h(h(SID_j || T_i || ID_i || N_j || (N_{ji} \oplus h(N_i)) \oplus V_3 \oplus h(h(SID_j || T_i || ID_i) \oplus N_{ji})).
\]

(19)

The authentication server and user phase is depicted in Fig. 6 as Authentication Phase 2.

---

**Figure 5. Authentication Phase 1**
C. Case 2: For the login not the first time

C.1. Login phase

If $U_i$ wants to obtain the service of the server $S_j$ after the first time of login without the participation of $RC$, $U_i$ stores $h(ID||RS_i) \oplus h(pw_i||ID||T_i)$ to the smart card. Note that a random integer, say $N_i$, should be added to the smart card as $h(ID||RS_i) \oplus h(pw_i||h(N_i)||T_i)$ in case that it suffers from the password guessing attack when a malicious insider of the server $S_j$ steals the smart card. The login and authentication is executed only between the user $U_i$ and the server $S_j$.

$U_i$ inserts his smart card and keys his identity $ID_i$ and password $PW_i$. If $h(ID||RS_i) \oplus h(pw_i||h(N_i)||T_i)$ is stored in the smart card, $U_i$ has to remember $h(N_i)$ and keys it. The smart card executes the following operations.

Step 1. Compute
$$h(ID||RS_i)=B \oplus h(pw_i||h(N_i)||T_i).$$

Step 2. Choose a random nonce $u$ and calculate
$$N_1=g^u, \quad C_1=h(ID||RS_i) \oplus N_1, \quad C_2=h(ID||N_i||SID_i).$$

Step 3. $U_i \rightarrow S_j$: \{ID_i, C_1, C_2, Flag\}, where Flag is set to ‘not the first time login’.

C.2. Authentication and session key agreement phase

After receiving \{ID_i, C_1, C_2, Flag\}, $S_j$ executes the following steps.

Step 1. Compute $N_j=h(ID||RS_j) \oplus C_1$, check $C_2=?h(ID||N_j||SID_j)$. If the equality holds, $S_j$ confirms $U_i$.

Step 2. Generate a random nonce $v$ and calculate
$$N_j=g^v, \quad V_1=h(ID||RS_j) \oplus h(N_j) \oplus N_j, \quad V_2=h(ID||SID_j||T_i)||N_j).$$

Step 3. Calculate the session key
$$SK=h(N_j) \oplus ID_i||h(N_j) \oplus SID_j||h(ID_i)||SID_j)((N_j)^y).$$

Step 4. $S_j \rightarrow U_i$: \{V_1, V_2\}.

After receiving the message from $S_j$, $U_i$ computes the session key $SK=h(N_j) \oplus ID_i||h(N_j) \oplus SID_j||h(ID_i)||SID_j)((N_j)^y)$.

D. Authentication and password change phase

If $U_i$ wants to change his password, $U_i$ inserts his smart card and keys his identity $ID_i$ and password $PW_i$. $U_i$ issues a require of replacing old password with a new password. The smart card executes the
following steps.

Step 1. Compute
\[ T_i = h(ID_i)[x] = B_i \oplus h(ID_i)[PW_i], \]
Step 2. Encrypt \(ID_i||require||t_i\) using \(T_i\) as the secret key,
\[ C = E_{T_i}(ID_i||require||t_i), \]
where \(E_{T_i}(\cdot)\) is a symmetric encryption algorithm, and \(t_i\) is the timestamp.

Step 3. \(U_i \rightarrow RC: \{ID_i, C\}\).
After receiving \(\{ID_i, C\}\), RC decrypts \(C\) and obtains \(ID_i\) and \(t_i\). When RC finishes checking the consistency of the identity \(ID_i\) and verifying the validity of time stamp \(t_i\), RC sends a response to the smart card in the form
\[ \text{C\_RC} = E_{T_i}(ID_i||tag||t_{RC}), \]
where \(t_{RC}\) is the timestamp and \(tag\) denotes Yes or No.

After receiving \(C_{\text{RC}}\), the smart card continues the following.

Step 1. Decrypts \(C_{\text{RC}}\) and obtains \(ID_i\) and \(t_{RC}\). Check if the decrypted identity is consistent with \(ID_i\) and if the time stamp \(t_{RC}\) is valid.

Step 2. If \(tag\) represents No, stop changing the password. Otherwise, the user keys a new password \(PW_{\text{new}_i}\).

Step 3. Compute
\[ B_{\text{new}_i} = B_i \oplus h(ID_i)[PW_i] \oplus h(ID_i)[PW_{\text{new}_i}], \]
and replace \(B_i\) with \(B_{\text{new}_i}\) in the smart card.

4. Analysis on the proposed scheme

It is easily known that our proposed scheme satisfies the attributes C1-C5 mentioned in Section I.

Now, we show that the proposed scheme holds Session key agreement (C6).

At the end of the session key agreement the first time, the session key is computed by
\[ SK = h(Z_i \oplus N_i)[N_j||h(ID_j)||RS_j)] \]  or \[ SK = h(h(SID_j)||T_i||ID_j) \oplus N_j \oplus (N_j)^{\alpha(N_j)} \oplus V_j \oplus h(h(SID_j)||T_i||ID_j) \oplus N_j). \]

On the assumption of Diffie-Hellman, to compute \(N_j^{\alpha(N_j)}\) is infeasible without \(u\) or \(v\). Only the user and the server have secret exponents \(u\) and \(v\), respectively. In order to obtain \(u\) from \(N_j\) or \(v\) from \(N_j\), one has to be faced with Discrete Logarithm problems.

If the session key is generated after the first time, the session key \(SK\) is \(h(h(N_j) \oplus ID_j)[h(N_j) \oplus SID_j][N_j]||h(ID_j)||RS_j))\) or \(h(h(N_j) \oplus ID_j)[h(N_j) \oplus SID_j][N_j]||h(ID_j)||RS_j)).\)

On the assumption of Diffie-Hellman, to compute \(N_j^{\alpha(N_j)}\) or \(N_j^{\alpha(N_j)}\) is infeasible without \(u\) or \(v\). In essence, only the user and the server have the secret exponents \(u\) and \(v\), respectively. Because obtaining \(u\) from \(N_j\) or \(v\) from \(N_j\) means solving Discrete Logarithm problems.

Next, we will discuss the security of our proposed in detail. We will analyze that well-known security threats cannot work in our scheme and compare the satisfaction of some merits with Hsiang et al.'s scheme [27], Chen et al.'s scheme [16], WJL-scheme [26], Liao et al.'s scheme [15] and Tsai's scheme [25] in Table I.

1. (1) Known-key security

Only knowing a compromised session key cannot determine the other used session keys. This is since every session key \(SK = h(Z_i \oplus N_j)[N_j||h(ID_j)||RS_j)).\) When the user logsins the server, \(U_i\) chooses a random \(N_i\) and \(S_i\) chooses a random \(N_j\). Even if the adversary has the previous session key, the adversary is still unable to obtain fresh \(N_i\) and \(N_j\). So the adversary cannot calculate the new session key. Therefore, the proposed scheme can provide known-key security.

2. (2) Perfect forward security

Even RC's master keys \(x\) and \(y\) of RC are disclosed, all previous fresh session keys cannot also be derived. If an adversary has the master key \(x\), he can compute \(N_j = h(RS_j||ID_j) \oplus V_1\) and further \(N_{\text{new}_i} = h(Z_i||h(ID_i)||RS_i) \oplus Z_i).\) However, the adversary cannot still work out \(v\) on the assumption of Discrete Logarithm and the assumption of Diffie-Hellman. Thus, the adversary cannot compute the
session key by the formula \( SK = h((Z_i \oplus N_j)||N^\prime||h(ID)||RS_j)) \).

(3) Two factor security

When an adversary obtains the smart card, he can extract the information stored in the card. Therefore, a secure smart card based scheme must have two-factor security. In the proposed scheme, even if the smart card is stolen but the user password of the device owner is unknown to the adversary, the adversary still can not recover \( h(ID)||x \) from \( B_i = h(ID)||x \oplus h(ID)||PW_i \). Our scheme holds two-factor security.

(4) Server spoofing attacks

Suppose an adversary wants to cheat a user \( U_i \) or RC by masquerading as a server \( S_j \). As the adversary does not have the secret key \( RS_j \) of the server \( S_j \), the adversary cannot produce a valid pair \( (V_1, V_2) \). In other words, the adversary cannot fool RC by masquerading as a server \( S_j \). The adversary cannot also compute \( N_i \) from \( h(ID)||SID_i \oplus C_1 \) (as \( T_i \) is protected by the password). So, it is impossible to compute \( Z_i \oplus N_i \) by the formula \( h(ID)||T_i||ID_i \oplus N_i \). Therefore, while the adversary sends back \( \{V_1, V_2, V_3\} \) to the user, the smart card can identify that they are not sent by the server \( S_j \). This is because the verification equation \( V_2 = h((Z_i \oplus N_i)||ID_i)||V_1||N_j) \) will not hold. Therefore, the proposed scheme can resist server spoofing attacks.

(5) Registration center spoofing attacks

Suppose an adversary wants to cheat a user \( U_i \) or a server \( S_j \) by masquerading as RC. As the adversary does not have the master key \( x \) or \( y \), the adversary cannot compute \( T_i \) or \( RS_j \). So, it is impossible to compute \( N_i \) from \( h(ID)||SID_i \oplus C_1 \) or \( N_j \) from \( h(ID)||SID_j \oplus V_1 \). The adversary cannot generate a valid tuple \( \{Z_i, Z_2, Z_3\} \) to pass the verification equation \( Z_2 =? h(Z_i \oplus N_i)||h(ID)||RS_j)||N_j \oplus N_i \). The server will realize that the information is not from RC. Similarly, the verification equation \( V_2 =? h((Z_i \oplus N_j)||ID_i)||V_1||N_j) \) will not hold. Therefore, the proposed scheme can resist registration center spoofing attacks.

(6) Impersonation attacks

The proposed scheme can withstand impersonation attacks as a legal user \( U_i \). In order to perform an impersonation attack, the attacker requires \( T_i \) to compute authentication messages \( \{C_1, C_2\} \). Even the server \( S_j \) cannot solve \( h(ID)||x \) from information which is sent by the user, since \( h(ID)||x \) is protected by a random number \( N_i \). To avoid \( S_j \) to get \( h(ID)||x \), RC sends \( \{Z_i, Z_2, Z_3\} \) to \( S_j \). \( h(ID)||x \) is mixed with the random nonce \( N_i \) and \( N_j \), so \( h(ID)||x \) is not obtained by the attacker. Therefore, the proposed scheme can resist impersonation attacks.

(7) Password guessing attacks

In the proposed scheme, the undetectable on-line password guessing attack will not work. Since RC can authenticate the user by checking the validity of the information \( \{C_1, C_2\} \). In addition, the off-line password guessing attacks will also fail in our proposed scheme. Since the password is protected in the smart card as \( h(ID)||x \oplus h(ID)||PW \). While \( h(ID)||x \) is contained in \( C_1 \) or \( C_2 \) and \( C_1 \) or \( C_2 \) are provided with a random nonce \( N_i \). It is infeasible to extract \( h(ID)||x \) from \( C_1 \) or \( C_2 \). Therefore, an attacker can not obtain a verification function about the password from the stored information \( h(ID)||x \oplus h(ID)||PW \) or the transmitted information. That is, the proposed scheme can resist guessing attacks.

(8) Denning-Sacco Replay attacks

In the proposed scheme, the freshness of the messages transmitted is provided by the random nonces \( N_i \) and \( N_j \), and the freshness of the shared session key \( SK \) is provided by the random nonce \( N^\prime \). Only the user and the server can access the shared secret value \( N^\prime \). Moreover, all the random nonce and the shared secret value \( N^\prime \) are embedded into the hash function to produce a fresh session key. Therefore, the proposed scheme can resist replay attacks.

(9) Private key leakage attacks

In the multi-server environment, the Private key leakage attack means that an attacker compromises an old session key and tries to find a long-term private key, e.g. other session keys, the user’s password, the registration center’s secret keys. The first two cases are discussed in Section 4.1.3 and Section 4.5.2. Here, we show that the proposed scheme can prevent an attacker from obtaining the registration center’s secret key. The secret key \( x \) is hashed in the form \( h(ID)||x \) and the secret key \( y \) is hashed in the form \( h(SID)||y \). During the transmission of the above values, they are hashed by adding some random
nones and other information. Therefore, an attacker cannot extract $x$ or $y$ from all the transmission information and the stored information in the smart card.

(10) Stolen verifier attacks

Since the proposed scheme does not maintain a verification table or password table in the servers or the registration center, nobody obtain any verifiable information from the servers and $RC$. So the proposed scheme can resist against the stolen verifier attack.

(11) Man-in-the-middle attacks

In the proposed scheme, the user and the server can achieve mutual authentication with the help of the registration center. Both the user and the server can authenticate the registration center. The registration center also can authenticate the user. Thus, the adversary cannot launch the man-in-the-middle attack to cheat either the user or the server.

Table I. Comparisons of satisfaction of the criteria

<table>
<thead>
<tr>
<th>criteria</th>
<th>Ours</th>
<th>[15]</th>
<th>[16]</th>
<th>[21]</th>
<th>[14]</th>
<th>[13]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perfect forward security</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Against password guessing attacks</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Against impersonation attacks</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Against server spoofing attacks</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

5. Conclusion

In the paper, we analyze an ID based remote user authentication scheme for multi-server environments. The authentication scheme lacks the authenticity of the registration center to the server. WJL-scheme does not provide perfect forward security. We present a new password based smart card authentication scheme in the multi-server environment. The security analyses prove that our scheme can withstand various possible attacks, and the functionality comparison can provide some advantages in contrast to the previous password based smart card authentication schemes.

6. Acknowledgment

This work is partially supported by a grant from the National Natural Science Foundation of China (10961013). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

7. References