Expert Systems with Applications 38 (2011) 13863–13870

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A secure dynamic ID based remote user authentication scheme for multi-server environment using smart cards

Cheng-Chi Lee [a,c,*], Tsung-Hung Lin [b], Rui-Xiang Chang [a]

[a] Department of Photonics & Communication Engineering, Asia University, No. 500, Lioufeng Road, Wufeng Shiang, Taichung, Taiwan, ROC
[b] Department of Computer Science and Information Engineering, National Chin-Yi University of Technology, 35, Lane215, Section 1, Chung-Shan Road, Taiping City, Taichung, Taiwan, ROC
[c] Department of Library and Information Science, Fu Jen Catholic University, 510 Jhongjheng Road, Sinjhuang City, Taipei County, Taiwan, ROC

ARTICLE INFO

Keywords:
Authentication
Smart cards
Dynamic ID
Multi-server system
Password

ABSTRACT

Recently, Hsiang et al. pointed out that Liao-Wang's dynamic ID based remote user authentication scheme for multi-server environment is vulnerable to insider attack, masquerade attack, server spoofing attack, registration center attack and is not easily reparable. Besides, Liao-Wang's scheme cannot achieve mutual authentication. For this, Hsiang et al. proposed an improved scheme to overcome these weaknesses and claimed that their scheme is efficient, secure, and suitable for the practical application environment. However, we observe that Hsiang et al.'s scheme is still vulnerable to a masquerade attack, server spoofing attack, and is not easily reparable. Furthermore, it cannot provide mutual authentication. Therefore, in this paper we propose an improved scheme to solve these weaknesses.

1. Introduction

Network security becomes an important issue in the communication environment. Password authentication is one of the mechanisms that were widely used to authenticate a legitimate user. Namely if the network users want to access the remote servers, they must be authenticated. In most password authentication schemes, the remote user needs to send a valid pair of the identity and the password to the remote server to make him/her authenticated when the user wants to access the remote server, In conventional password authentication schemes, the server maintains a password table to verify the user's login request as, for example, in Lamport's (Lamport, 1981) scheme. However, if an adversary can break into the server by someway, the table may be easily modified or corrupted. Then in 1990, Hwang, Chen, and Laih (1990) proposed a non-interactive password authentication scheme using a smart card without maintaining verification table. Many schemes (Hwang & Ku, 1995; Ku & Chen, 2004; Shen, Lin, & Hwang, 2003; Sun, 2000) have subsequently been proposed to make secure the authentication over insecure channels, but all these schemes are designed for the single-server environment.

In conventional user authentication schemes, a network user not only needs to log into various remote servers with repetitive registration, but also needs to remember the various user identities and passwords. Therefore, in 2000, Lee and Chang (2000) proposed a user identification and key distribution scheme based on the difficulty of factorization and hash function for multi-server environment. The user registered at the registration centre once and can use all the permitted services in remote servers. Then many schemes (Chang & Lee, 2004; Hsiang & Shih, 2009; Juang, 2004; Lee & Chang, 2000; Li, Lin, & Hwang, 2001; Liao & Wang, 2009; Lin, Hwang, & Li, 2003; Wu & Hsu, 2004; Yang, Wang, Bao, Wang, & Deng, 2004) have subsequently been proposed for the multi-server environment.

In 2001, Li et al. (2001) proposed a remote user authentication scheme based on the neural networks. However, their scheme was found to waste much time. To remedy this, Lin et al. (2003) proposed an efficient remote user authentication based on discrete logarithm problem for multi-server environments. Later Juang (2004) showed that Lin et al.'s scheme is not efficient in the authentication process, because every user must have a large of memory to store the public parameters for authentication. For this, Juang proposes an efficient multi-server password authenticated key agreement scheme based on the hash function and symmetric key cryptosystem. However, Chang and Lee (2004) pointed out that Juang' scheme still lacks efficiency and is vulnerable to off-line dictionary attack, if the secret value of the smart card is extracted by some way. Therefore, Change-Lee proposed a novel remote user authentication scheme to remedy these weaknesses. In their scheme, the registration center distributed the secret key x to each registered server via secure channel. However, their scheme was found to an insider attack, spoofing attack and registration center spoofing attack.

* Corresponding author at: Department of Library and Information Science, Fu Jen Catholic University, 510 Jhongjheng Road, Sinjhuang City, Taipei County, Taiwan, ROC. Tel.: +886 4 23323456x20059; fax: +886 4 2330 6835.
E-mail addresses: cclee@mail.fju.edu.tw (C.-C. Lee), duke@ncut.edu.tw (T.-H. Lin).

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.04.190

Most of password authentication schemes for multi-server environment are based on static ID, the login ID is sent in the form of plaintext through public networks. An adversary might intercept the login ID from the public network and use it to trace the legal user. Therefore, Liao and Wang (2009) proposed a secure and efficient authentication scheme with anonymity for multi-server environment. Their scheme only uses one way hash functions to implement mutual verification and session key agreement. They claimed that their scheme can resist various attack and achieve mutual authentication. However, Hsiang and Shih (2009) pointed out that Liao-Wang's scheme is vulnerable to an insider attack, masquerade attack, server spoofing attack, registration center spoofing attack, and is not reparable. Beside, Liao-Wang's scheme cannot achieve mutual authentication. To solve these problems, Hsiang et al. proposed an improvement on Liao-Wang's scheme. However, we will find that Hsiang et al.'s scheme is still vulnerable to a masquerade attack, server spoofing attack, and is not easily reparable. Furthermore, Hsiang et al.'s scheme cannot provide mutual authentication. Therefore, in this paper we propose an improved scheme to solve these weaknesses.

The following six requirements should be taken into consideration of the password authentication scheme for multi-server environment:

(1) *No verification table.* A server does not have stored any verification or password table.
(2) *Freely chosen password.* Any user can freely choose and change his/her passwords.
(3) *Mutual authentication and session key agreement.* Severs and users can authenticate each other and establish a session key for protecting their subsequent communications.
(4) *Low computation and communication cost.* The smart card cannot provide a powerful computation capability and high bandwidth. Since the computation ability of the smart card is very limited.
(5) *Single registration.* Any user only must register at the registration centre once and can use all the permitted services in remote servers.
(6) *Security*. The authentication scheme must be able to resist all kinds of attacks.

This paper is organized as follows: in Section 2, we review Hsiang et al.'s remote user authentication scheme. The security flaws of Hsiang et al.'s scheme are shown in Section 3. The Section 4 is our improved scheme. In the Section 5, we discuss the security and performance of our improved scheme. Finally, our conclusion is given in Section 6.

2. Review of Hsiang et al.'s scheme

In this section, we review Hsiang et al.'s remote user authentication scheme. Their scheme contains four phases: registration phase, login phase, verification phase, and password change phase. In this scheme, there are three main participants in Hsiang et al.'s remote user authentication scheme: the user (U_i), the remote server (S_j), and the registration center (RC). RC is assumed to be trustworthy. For the legal server S_j, RC computes $h(SID_j\|y)$ and shares it with S_j in the secure channel. Beside, RC chooses the master secret key x and two secret numbers r and y, and then only RC knows the master secret key x and two secret numbers r and y. Table 1 lists the notations used in Hsiang et al.'s scheme.

Table 1
The notations used in Hsiang et al.'s scheme.

Notations	Descriptions
U_i	The i_{th} user
ID_i	The identity of U_i
PW_i	The password of U_i
S_j	The j_{th} server
RC	The registration center
SC	A smart card
SID_j	The identity of S_j
CID_i	The dynamic ID of U_i
x	The secret key maintained by registration center
$h(.)$	A one-way hash function
$\oplus$	The bitwise XOR operation
$\|$	String concatenation operation
$\Rightarrow$	A secure channel
$\rightarrow$	A common channel

2.1. Registration phase

When the user U_i wants to access the systems, he/she has to submit his/her identity ID_i and PW_i to RC. The steps of the registration phase are as follows:

Step R1. $U_i \Rightarrow RC$: ID_i, $h(b \oplus PW_i)$.
U_i freely chooses his/her identity ID_i and PW_i, and computes $h(b \oplus PW_i)$, where b is a random number generated by U_i. Then U_i sends ID_i and $h(b \oplus PW_i)$ to the registration center RC for registration through a secure channel.

Step R2. RC computes

$$T_i = h(ID_i\|x)$$
$$V_i = T_i \oplus h(ID_i\|h(b \oplus PW_i))$$
$$A_i = h(h(b \oplus PW_i)\|r) \oplus h(x \oplus r)$$
$$B_i = A_i \oplus h(b \oplus PW_i)$$
$$R_i = h(h(b \oplus PW_i)\|r)$$
$$H_i = h(T_i)$$

Step R3. $RC \Rightarrow U_i$: RC issues a smart card to U_i, and the card contains $\{V_i, B_i, R_i, H_i, h(.)\}$.

Step R4. U_i keys b into his/her smart card, then the smart card contains $\{V_i, B_i, R_i, H_i, b, h(.)\}$.

2.2. Login phase

After receiving the smart card from RC, U_i can use it when he/she wants to log into S_j. This phase is depicted in Fig. 1, and the detailed steps are performed as follows.

Step L1. U_i inserts his/her smart card into the smart card reader and then inputs ID_i and PW_i. Then the smart card computes $T_i = V_i \oplus h(ID_i\|h(b \oplus PW_i))$ and $H_i^* = h(T_i)$, and then checks whether the H_i^* is the same as H_i. If they are the same, U_i proceeds to the next step. Otherwise the smart card rejects this login request.

Step L2. The smart card generates a nonce N_i and computes

$$A_i = B_i \oplus h(b \oplus PW_i)$$
$$CID_i = h(b \oplus PW_i) \oplus h(T_i\|A_i\|N_i)$$
$$P_{ij} = T_i \oplus h(A_i\|N_i\|SID_j)$$
$$Q_i = h(B_i\|A_i\|N_i)$$
$$D_i = R_i \oplus SID_j \oplus N_i$$
$$C_0 = h(A_i\|N_i + 1\|SID_j)$$

Step L3. $U_i \rightarrow S_j$: CID_i, P_{ij}, Q_i, D_i, C_0, N_i.

2.3. Verification phase

After receiving the login request sent from U_i, S_j performs the following tasks to authenticate the user's login request. This phase

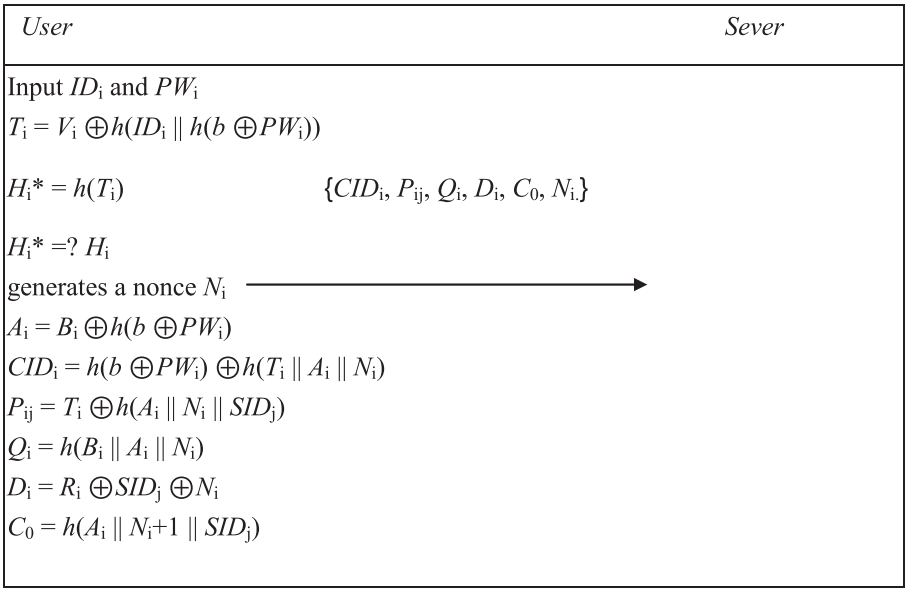

Fig. 1. Login phase of Hsiang et al.'s scheme.

is depicted in Fig. 2, and the detailed steps are performed as follows.

Step V1. Upon receiving the login request, S_j generates nonce N_{jr} and computes

$$M_{jr} = h(SID_j \| y) \oplus N_{jr}.$$

Then he/she sends the message $\{M_{jr}, SID_j, D_i, C_0, N_i\}$ to *RC*.

Step V2. Upon receiving the message $\{M_{jr}, SID_j, D_i, C_0, N_i\}$, *RC* computes

$$N'_{jr} = M_{jr} \oplus h(SID_j \| y)$$
$$R'_i = D_i \oplus SID_j \oplus N_i$$
$$A'_i = R'_i \oplus h(x \oplus r)$$
$$C'_0 = h(A'_i \| N_i + 1 \| SID_j)$$

RC checks if the computed C'_0 is the same as the received C_0. If they are the same, *RC* further generates nonce N_{rj} and computes $C_1 = h(N'_{jr} \| h(SID_j \| y) \| N_{rj})$ and $C_2 = A_i \oplus h(h(SID_j \| y) \oplus N'_{jr})$. Otherwise, *RC* rejects it. Finally, the registration center *RC* responses the message $\{C_1, C_2, N_{rj}\}$ to S_j.

Step V3. Upon receiving the message $\{C_1, C_2, N_{rj}\}$, S_j computes $C'_1 = h(N_{jr} \| h(SID_j \| y) \| N_{rj})$ and checks if the computed C'_1 is the same as the received C_1. If they are the same, S_j authenticates *RC* successfully and computes $A_i = C_2 \oplus h(h(SID_j \| y) \oplus N_{rj})$, $T_i = P_{ij} \oplus h(A_i \| N_i \| SID_j)$, $h(b \oplus PW_i) = CID_i \oplus h(T_i \| A_i \| N_i)$, and $B_i = A_i \oplus h(b \oplus PW_i)$.

Step V4. S_j checks if the computed $h(B_i \| A_i \| N_i)$ is the same as the received Q_i. If they are the same, S_j authenticates U_i successfully. Otherwise, S_j rejects the login request.

Step V5. S_j generates nonce N_j and computes $M'_{ij} = h(B_i \| N_i \| A_i \| SID_j)$, and then responses the message $\{M'_{ij}, N_j\}$ to U_i.

Step V6. Upon receiving the message $\{M'_{ij}, N_j\}$, U_i computes $h(B_i \| N_i \| A_i \| SID_j)$ and checks if the computed $h(B_i \| N_i \| A_i \| SID_j)$ is the same as the received M'_{ij}. If they are the same, U_i authenticates S_j successfully and computes $M''_{ij} = h(B_i \| N_j \| A_i \| SID_j)$, the session key $SK = h(B_i \| A_i \| N_i \| N_j \| SID_j)$, and then U_i responses the message $\{M''_{ij}\}$ to S_j.

Step V7. Upon receiving the message $\{M''_{ij}\}$, S_j computes $h(B_i \| N_j \| A_i \| SID_j)$ and checks if the computed $h(B_i \| N_j \| A_i \| SID_j)$ is the same as the received M''_{ij}. If they are the same, S_j authenticates U_i successfully. S further computes $SK = h(B_i \| A_i \| N_i \| N_j \| SID_j)$ as the session key for securing communications with U.

2.4. Password change phase

In this phase, U_i can change his/her password any time when he/she wants. The steps of the password change phase are as follows:

Step P1. U_i inserts his smart card into the smart card reader and then inputs ID_i and PW_i.

Step P2. The smart card computes $T_i = V_i \oplus h(ID_i \| h(b \oplus PW_i))$ and $H_i^* = h(T_i)$ and then checks if the H_i^* is the same as H_i. If they are the same, U_i chooses a new password PW_{new}.

Step P3. The smart card computes

$$V_{new} = T_i \oplus h(ID_i \| h(b \oplus PW_{new}))$$
$$B_{new} = B_i \oplus h(b \oplus PW_i) \oplus h(b \oplus PW_{new})$$

Finally, the smart card replaces V_i and B_i with V_{new} and B_{new}.

3. Cryptanalysis of Hsiang et al.'s scheme

In this section, we will demonstrate that Hsiang et al.'s scheme is vulnerable to a masquerade attack, server spoofing attack, and is not easily reparable, and then any legal user can masquerade other legal users to log into remote server without knowing users' password. Beside, their scheme cannot achieve mutual authentication.

3.1. Masquerade attack

We assume that the adversary Z is a legal user of the system, and then he/she can obtain a smart card containing $\{V_z, B_z, R_z, H_z, b, h(.)\}$. Then the adversary Z can compute $A_z \oplus R_z = h(x \oplus r)$, where $A_z = B_z \oplus h(b \oplus PW_z)$. When another legal

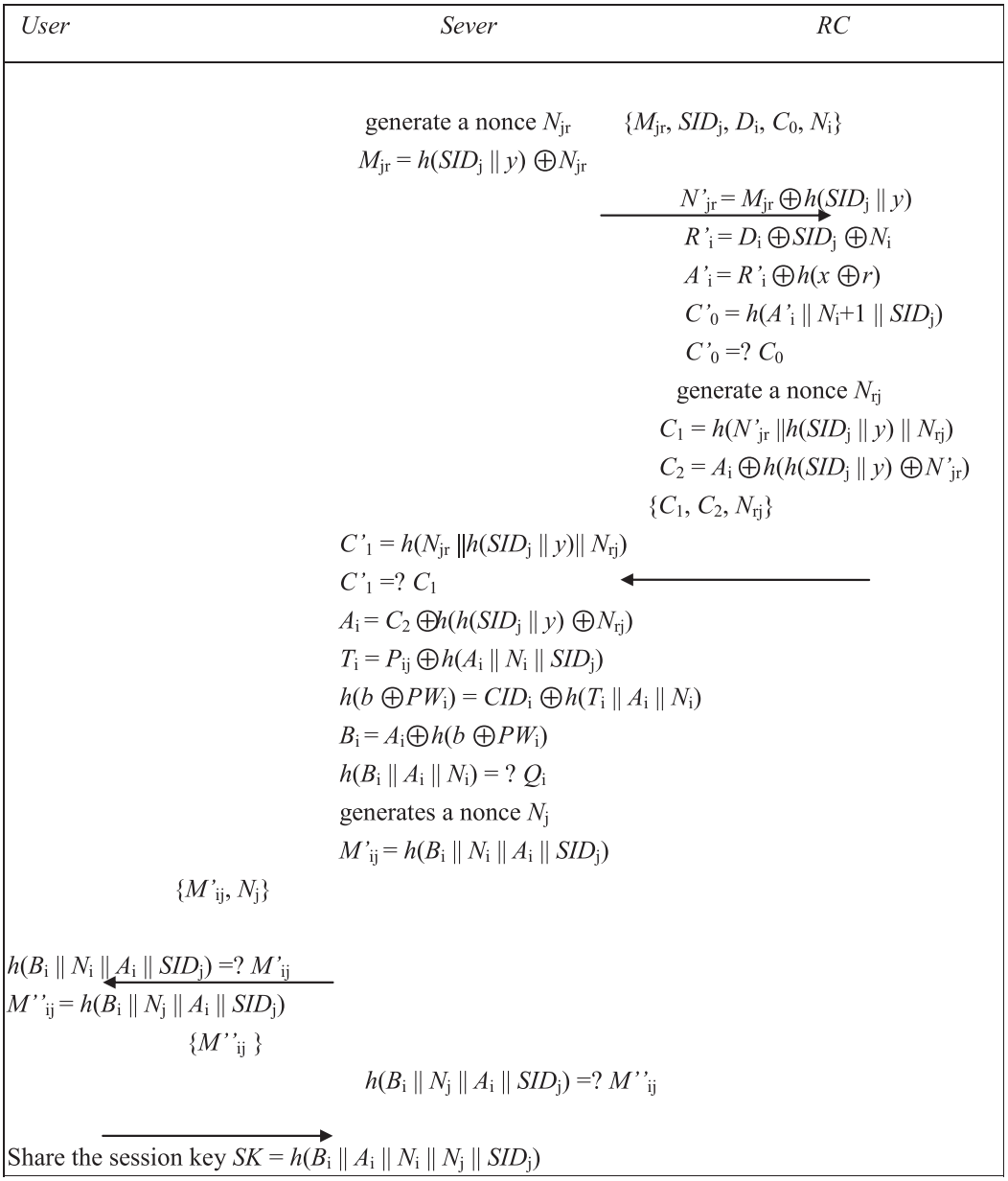

Fig. 2. Verification phase of Hsiang et al.'s scheme.

user U_i communicates with S_j, the adversary can intercept the login message $\{CID_i, P_{ij}, Q_i, D_i, C_0, N_i\}$ between U_i and S_j.

The login request is sent as the form of plaintext through a public networks. Any user, including illegal ones, can intercept it from the public network. Then the adversary can compute

$$R_i = D_i \oplus SID_j \oplus N_i$$
$$A_i = R_i \oplus h(x \oplus r)$$
$$T_i = P_{ij} \oplus h(A_i \| N_i \| SID_j)$$
$$h(b \oplus PW_i) = CID_i \oplus h(T_i \| A_i \| N_i)$$
$$B_i = A_i \oplus h(b \oplus PW_i)$$

When the adversary Z obtains $\{T_i, h(b \oplus PW_i), A_i, B_i, R_i\}$, he/she can masquerade as legal user U_i to log into the remote server without knowing user's password. Then the adversary generates a nonce N_z and computes $CID_i^* = h(b \oplus PW_i) \oplus h(T_i \| A_i \| N_z)$, $P_{ij}^* = T_i \oplus h(A_i \| N_z \| SID_j)$, $Q_i^* = h(B_i \| A_i \| N_z)$, $D_i^* = R_i \oplus SID_j \oplus N_z$, and $C_0^* = h(A_i \| N_z + 1 \| SID_j)$. Finally, the adversary Z sends the forged login message $\{CID_i^*, P_{ij}^*, Q_i^*, D_i^*, C_0^*, N_z\}$ to the remote server S_j.

After receiving these message, S_j generates nonce N_{jr} and computes $M_{jr} = h(SID_j \| y) \oplus N_{jr}$. Then sends the message $\{M_{jr}, SID_j, D_i^*, C_0^*, N_z\}$ to RC.

Upon receiving the message $\{M_{jr}, SID_j, D_i^*, C_0^*, N_z\}$, RC computes $N'_{jr} = M_{jr} \oplus h(SID_j \| y)$, $R'_i = D_i^* \oplus SID_j \oplus N_z$, $A'_i = R'_i \oplus h(x \oplus r)$, and $C'_0 = h(A'_i \| N_z + 1 \| SID_j)$. Then RC checks if the computed C'_0 is the same as the received C_0^*. If they are the same, RC further generates nonce N_{rj} and computes $C_1 = h(N'_{jr} \| h(SID_j \| y) \| N_{rj})$ and $C_2 = A_i \oplus h(h(SID_j \| y) \oplus N'_{jr})$. Finally, RC responses the message $\{C_1, C_2, N_{rj}\}$ to S_j.

Upon receiving the message $\{C_1, C_2, N_{rj}\}$, S_j computes $C'_1 = h(N_{jr} \| h(SID_j \| y) \| N_{rj})$ and checks if the computed C'_1 is the same as the received C_1. If they are the same, S_j authenticates RC successfully and computes $A_i = C_2 \oplus h(h(SID_j \| y)) \oplus N_{rj}$, $T_i = P_{ij}^* \oplus h(A_i \| N_z \| SID_j)$, $h(b \oplus PW_i) = CID_i^* \oplus h(T_i \| A_i \| N_z)$, and $B_i = A_i \oplus h(b \oplus PW_i)$. Then S_j checks if the computed $h(B_i \| A_i \| N_z)$ is the same as the received Q_i^*. If they are the same, S_j will accept the forged login

request. Therefore, any legal user can masquerade other legal users to log into remote server without knowing users' password in Hsiang et al.'s scheme.

3.2. Server spoofing attack

In the previous paragraphs, we demonstrated that Hsiang et al.'s scheme is vulnerable to a masquerade attack. If the adversary Z is a legal user, he/she can masquerade as legal user to log into the remote server. Similarly, the adversary Z can masquerade server to fool any legal user. When another legal user U_i communicates with S_j, the adversary Z generates a nonce N_z and computes $M_{ij}^* = h(B_i||N_z||A_i||SID_j)$, and then sends the message $\{M_{ij}^*, N_z\}$ to U_i. U_i will compute $h(B_i||N_z||A_i||SID_j)$ and compare it with M_{ij}^*. If they are equal, U_i responses the message $h(B_i||N_i||A_i||SID_j)$ and computes the session key $h(B_i||A_i||N_i||N_z||SID_j)$. The adversary Z can decrypt the entire message sent from U_i. Therefore, Hsiang et al.'s scheme is vulnerable to the server spoofing attack.

3.3. Poor reparability

Assume that the adversary has performed successfully the masquerade attack to obtain U_i secret value A_i and B_i. If U_i finds someone to masquerade him/her to log into any remote server S_j, he/she may want to re-register with RC. However, it is no uses for U_i re-register with RC, because any legal users, including illegal ones, can obtain the secret value $h(x \oplus r)$ by computing $A_z \oplus R_z = h(x \oplus r)$. Then when the legal user $U_i$communicates with S_j, the adversary Z can intercept the login message $\{CID_i, P_{ij}, Q_i, D_i, C_0, N_i\}$ and compute $R_i = D_i \oplus SID_j \oplus N_i$, $A_i = R_i \oplus h(x \oplus r)$, $T_i = P_{ij} \oplus h(A_i||N_i||\ SID_j)$, $h(b \oplus PW_i) = CID_i \oplus h(T_i||A_i||N_i)$, and $B_i = A_i \oplus h(b \oplus PW_i)$. The adversary Z also can masquerade as legal user to log into the remote server without knowing the password. Therefore, Hsiang et al.'s scheme is not easily reparable.

3.4. Lack of mutual authentication

In Hsiang et al.'s scheme, when U_i wants to log into the remote server S_j, he/she sends the login request to S_j. Upon receiving the login request $\{CID_i, P_{ij}, Q_i, D_i, C_0, N_i\}$ from U_i, S_j generates nonce N_{jr} and computes $M_{jr} = h(SID_j||y) \oplus N_{jr}$. Then S_j sends the message $\{M_{jr}, SID_j, D_i, C_0, N_i\}$ to RC. Upon receiving the message $\{M_{jr}, SID_j, D_i, C_0, N_i\}$, RC computes

$$N'_{jr} = M_{jr} \oplus h(SID_j||y)$$
$$R'_i = D_i \oplus SID_j \oplus N_i$$
$$A'_i = R'_i \oplus h(x \oplus r)$$
$$C'_0 = h(A'_i||N_i + 1||SID_j)$$

RC checks if the computed C'_0 is the same as the received C_0. If they are the same, RC further generates nonce N_{rj} and computes

$$C_1 = h(N'_{jr}||h(SID_j||y)||N_{rj})$$
$$C_2 = A_i \oplus h(h(SID_j||y) \oplus N'_{jr})$$

Otherwise, RC rejects authentication request. Finally, RC sends the message $\{C_1, C_2, N_{rj}\}$ to S_j. Upon receiving the message $\{C_1, C_2, N_{rj}\}$, S_j computes $C'_1 = h(N_{jr}||h(SID_j||y)||N_{rj})$ and checks if the computed C'_1 is the same as the received C_1. If they are the same, S_j authenticates RC successfully and computes

$$A_i^* = C_2 \oplus h(h(SID_j||y) \oplus N_{rj})$$
$$T_i = P_{ij} \oplus h(A_i||N_i||SID_j)$$
$$h(b \oplus PW_i) = CID_i \oplus h(T_i||A_i||N_i)$$
$$B_i = A_i \oplus h(b \oplus PW_i)$$

S_j checks if the computed $h(B_i||A_i||N_i)$ is the same as the received Q_i. Since $A_i = C_2 \oplus h\ (h(SID_j||y) \oplus N_{jr})$ and $A_i^* = C_2 \oplus h(h(SID_j||y) \oplus N_{rj})$, they are not equal. Their scheme uses a wrong computation in Step V3 of the verification phase. Then any legal user cannot pass S_j's authentication. Therefore, Hsiang et al.'s scheme cannot provide mutual authentication. I think the slip of the pen is very serious problem in this scheme.

4. Our scheme

In this scheme, we propose an improved scheme to avoid various attacks. Our scheme consists of four phases: registration phase, login phase, verification phase, and password change phase. Three entities are involved: the user (U_i), the server (S_j), and the registration center (RC). RC chooses the master key x and secret number y to compute $h(x||y)$ and $h(y)$, and then shares them with S_j in the secure channel. Only RC knows the master secret key x and secret number y.

4.1. Registration phase

When the user U_i wants to access the systems, he/she has to submit his/her identity ID_i and PW_i to RC. The steps of the registration phase are as follows:

Step R1. $U_i \Rightarrow RC$: ID_i, $h(b \oplus PW_i)$.U_i freely chooses his/her identity ID_i and PW_i, and computes $h(b \oplus PW_i)$, where b is a random number generated by U_i. Then U_i sends ID_i and $h(b \oplus PW_i)$ to the registration center RC for registration through a secure channel.

Step R2. RC computes

$$T_i = h(ID_i||x)$$
$$V_i = T_i \oplus h(ID_i||h(b \oplus PW_i))$$
$$B_i = h(h(b \oplus PW_i)||h(x||y))$$
$$H_i = h(T_i)$$

Step R3. $RC \Rightarrow U_i$: RC issues a smart card to U_i, and the card contains $\{V_i, B_i, H_i, h(.), h(y)\}$.

Step R4. U_i keys b into his/her smart card, then the smart card contains $\{V_i, B_i, H_i, b, h(.), h(y)\}$.

4.2. Login phase

After receiving the smart card from RC, U_i can use it when he/she wants to log into S_j. The steps of the login phase are as following:

Step L1. U_i inserts his/her smart card into the smart card reader and then inputs ID_i and PW_i. Then the smart card computes $T_i = V_i \oplus h(ID_i||h(b \oplus PW_i))$ and $H_i^* = (h(T_i)$, and then checks if the H_i^* is the same as H_i. If they are the same, U_i proceeds to the next step. Otherwise the smart card rejects this login request.

Step L2. The smart card generates a nonce N_i and computes

$$A_i = h(T_i||h(y)||N_i)$$
$$CID_i = h(b \oplus PW_i) \oplus h(T_i||A_i||N_i)$$
$$P_{ij} = T_i \oplus h(h(y)||N_i||SID_j)$$
$$Q_i = h(B_i||A_i||N_i)$$

Step L3. $U_i \rightarrow S_j$: CID_i, P_{ij}, Q_i, N_i.

4.3. Verification phase

After receiving the login request sent from U_i, S_j performs the following tasks to authenticate the user's login request. The steps of the verification phase are as follows:

Step V1. Upon receiving the login request $\{CID_i, P_{ij}, Q_i, N_i\}$, S_j computes $T_i = P_{ij} \oplus h(h(y)||N_i||SID_j)$, $A_i = h(T_i||h(y)||N_i)$, $h(b \oplus PW_i) = CID_i \oplus h(T_i||A_i||N_i)$ and $B_i = h(h(b \oplus PW_i)||h(x||y))$ by using received message $\{CID_i, P_{ij}, N_i\}$, $h(y)$ and $h(x||y)$.

Step V2. S_j computes $h(B_i||A_i||N_i)$ and checks it with Q_i. If they are not equal, S_j rejects the login request and terminates this session. Otherwise, S_j accepts the login request and generates a nonce N_j to compute $M'_{ij} = h(B_i||N_i||A_i||SID_j)$. Finally, S_j sends the message $\{M'_{ij}, N_j\}$ to U_i.

Step V3. Upon receiving these message $\{M'_{ij}, N_j\}$ from S_j, $U_i$computes $h(B_i||N_i||A_i||SID_j)$ and checks it with received message M'_{ij}. If they are not equal, U_i rejects these messages and terminates this session. Otherwise, U_i authenticates successfully S_j and computes $M''_{ij} = h(B_i||N_j||A_i||SID_j)$. Finally, $U_i$sends back the message $\{M''_{ij}\}$ to S_j.

Step V4. Upon receiving this message $\{M''_{ij}\}$, S_j computes $h(B_i||N_i||A_i||SID_j)$ and checks it with received message $\{M''_{ij}\}$. If they are equal, S_j authenticates successfully U_i. After finishing verification phase, U_i and S_j can compute $SK = h(B_i||N_i||N_j||A_i||SID_j)$ as the session key for securing communications with authenticator.

The login phase and verification phase are depicted in Fig. 3.

4.4. Password change phase

In this phase, U_i can change his/her password any time when he/she wants. The steps of the password change phase are as follows:

Step P1. U_i inserts his smart card into the smart card reader and then inputs ID_i and PW_i.

Step P2. The smart card computes $T_i = V_i \oplus h(ID_i||h(b \oplus PW_i))$ and $H_i^* = h(T_i)$ and then checks if the H_i^* is the same as H_i. If they are the same, U_i chooses a new password and a new random number b_{new} to compute $h(b_{new} \oplus PW_{new})$ and $V_{new} = T_i \oplus h(ID_i||h(b_{new} \oplus PW_{new}))$. Finally, U_i sends ID_i and $h(b_{new} \oplus PW_{new})$ to RC in the secure channel.

Step P3. RC computes

$$B_{new} = h(h(b_{new} \oplus PW_{new})||h(x||y)).$$

RC sends back B_{new} to U_i.

Step P4. Finally, the smart card replaces V_i and B_i with V_{new} and B_{new}.

5. Analysis of our scheme

This section describes the security analyses of the improved scheme and compares performance with other schemes. To evaluate the security of our improved scheme, we assume that the adversary might execute various attacks to defeat the improved scheme under the smart card based authentication environments. The adversary might perform various attacks as following:

5.1. Masquerade attack

If the adversary tries to masquerade as the legal user to log into the remote server S_j, he/she must enable to forge a valid login request $\{CID_i, P_{ij}, Q_i, N_i\}$ to fool S_j. However, the adversary cannot compute $CID_i = h(b \oplus PW_i) \oplus h(T_i||A_i||N_i)$ and $Q_i = h(B_i||A_i||N_i)$ without the knowledge of A_i, B_i and PW_i.

In addition, if the adversary is a legal user of the system, and then he/she also cannot masquerade as any legal user to log into the remote server S_j. Because he/she cannot compute $B_i = h(h(b \oplus PW_i)||h(x||y))$ from his/her smart card and the intercepted login request $\{CID_i, P_{ij}, Q_i, N_i\}$ without knowing $h(x||y)$.

Beside, if the adversary has obtain the smart card and extract the parameters $\{V_i, B_i, H_i, b, h(.), h(y)\}$ stored in the smart card by some way. He/She also cannot forge a login request to fool S_j, because he/she cannot use the parameter $\{V_i, B_i, H_i, b, h(.), h(y)\}$ to compute the correct values of T_i and A_i without knowing the master key x. Therefore, our improved scheme can withstand the masquerade attack.

5.2. Server spoofing and registration center spoofing

The adversary might try to server spoofing attack in our improved scheme. If the adversary is a legal user of the system, he/she must enable to forge a valid response request $\{M'_{ij}, N_j\}$ to U_i. However, the adversary cannot compute $M'_{ij} = h(B_i||N_i||A_i||SID_j)$ without the knowledge of A_i and $B_i$from his smart card and the intercepted login request $\{CID_i, P_{ij}, Q_i, N_i\}$. It means that the adversary has no way to compute M'_{ij}. Moreover, the adversary cannot compute the session key $SK = h(B_i||N_i||N_j||A_i||SID_j)$ without the knowledge of A_i and B_i. Therefore, our improved scheme can withstand the server spoofing attack.

5.3. Mutual authentication

In Hsiang et al.'s scheme, the mutual authentication cannot achieve since they use a wrong computation in Step V3 of the verification phase. Any legal user cannot pass S_j's authentication in their scheme. Then we propose an improved scheme to provide the mutual authentication. In our scheme, when the user U_i wants to access the systems, he/she sends the login request to S_j. Then S_j will accept the login request in Step V2, and response the message $\{M'_{ij}, N_j\}$ to U_i. Upon receiving these message from S_j, $U_i$computes the hash value $h(B_i||\ N_i||A_i||SID_j)$ to authenticate S_j in Step V3, and then responses the message $\{M''_{ij}\}$ to S_j. Upon receiving this message $\{M''_{ij}\}$, S_j computes the hash value $h(B_i||N_i||A_i||SID_j)$ to authenticate $U_i$in Step V4, and then computes the session key $SK = h(B_i||N_i||N_j||A_i||SID_j)$ to secure communications with U_i. Therefore, the mutual authentication is achieved in the improved scheme.

5.4. Reparability

If the user U_i finds that $B_i = h(h(b \oplus PW_i)||h(x||y))$ has been compromised, he/she can re-register with RC in the secure channel. U_i chooses a new password and a new random number b_{new} to compute $h(b_{new} \oplus PW_{new})$ and $V_{new} = T_i \oplus h(ID_i||h(b_{new} \oplus PW_{new}))$. Finally, U_i sends ID_i and $h(b_{new} \oplus PW_{new})$ to RC in the secure channel. After receiving ID_i and $h(b_{new} \oplus PW_{new})_i$, RC computes $B_{new} = h(h(b_{new} \oplus PW_{new})||h(x||y))$ and then sends back B_{new} to U_i. After receiving B_{new} from RC, the smart card replaces V_i and B_i with V_{new} and B_{new}. Then U_i can securely login by using his/her smart card and PW_{new}. The adversary's login request, which is derived from $B_{i,}$ will be reject. Because S_j will reject the adversary's login request by checking $Q_i = ?Q_{new}$. Since $Q_i = h(B_i||A_i||N_i)$ and $Q_{new} = h(B_{new}||A_i||N_{new})$. Therefore, the improved scheme is easily reparable.

5.5. Performance

This phase compares the performance and functionality of the improved scheme with previously proposed schemes. We mostly

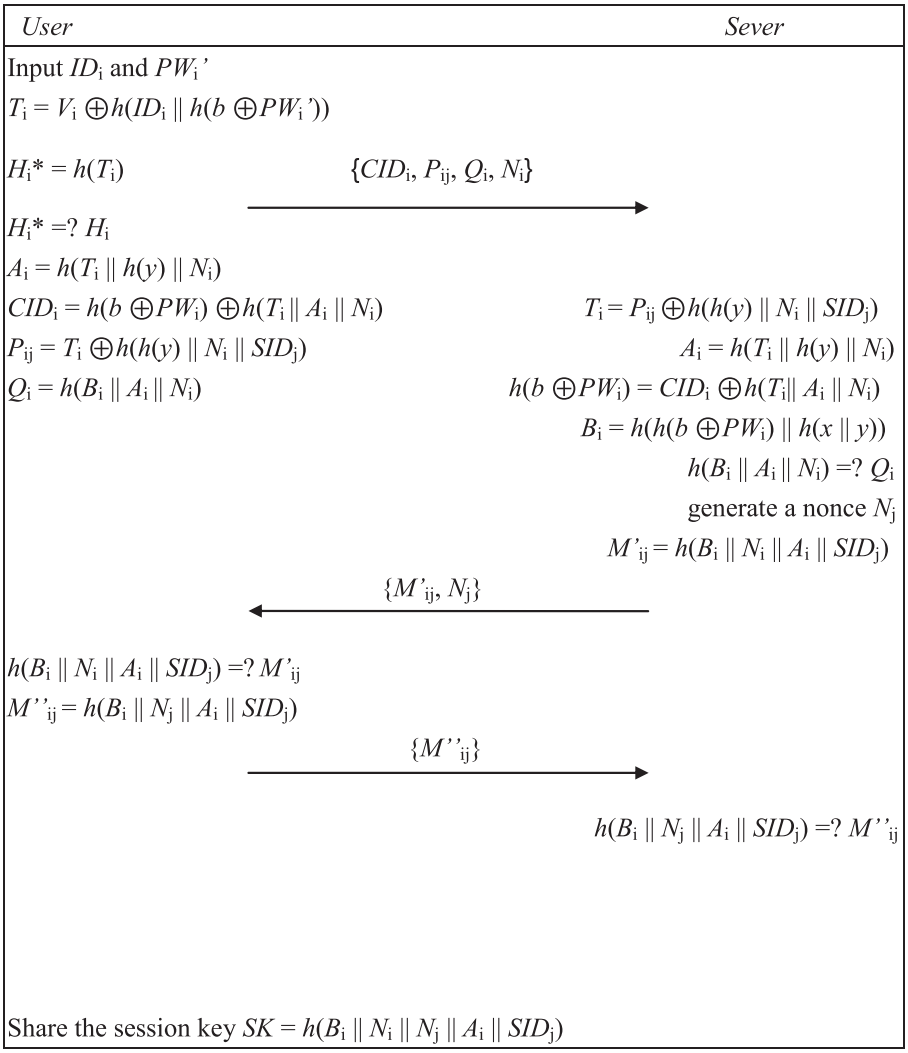

Fig. 3. Login and verification phase of our scheme.

focus on the computations of login and verification phases, because the two phases are the principal part of an authentication scheme. We define the notation T_h is the time for executing a one-way hash function and T_{sym} is the time for executing a symmetric-key encryption/decryption. According to the experimental result of related researches (Argyroudis, Verma, Tewari, & O'Mahony, 2004; Passing & Dressler, 2006; Wong, Fuentes, & Chan, 2001), we know that one-way hash functions are more efficient than symmetric cryptosystems.

Table 2 illustrates the comparison of the improved scheme and previously proposed schemes. Chang-Lee's scheme is using the symmetric cryptosystems that consumes more computation resources than the others. Because Liao-Wang's, Hsiang et al.'s and improved scheme are only using one way hash function. On the other hand, we observe that Hsiang et al.'s scheme has performance level coordinate to those of improved scheme except that the former requires 5 extra one way hash functions.

In Table 2, it is obvious that Liao-Wang's and our improved scheme have the same computation cost. However, Liao-Wang's scheme is vulnerable to insider attack, masquerade attack, server spoofing attack, registration center attack, and is not reparable. Table 3 lists the functionality comparisons between our improved scheme and others. It can be seen that functionality comparisons of our improved scheme is more secure against various attacks.

Table 2
Performance of the improved scheme and previously proposed schemes.

	Communication cost of the login and verification phase
Our proposed scheme	$15T_h$
The Hsiang et al. scheme	$20T_h$
The Liao-Wang scheme	$15T_h$
The Chang-Lee scheme	$8T_h + 6T_{sym}$

Table 3
Comparisons between our proposed scheme and other schemes.

	Ours	Hsiang	Liao-Wang	Chang-Lee
Securely chosen password	O	O	O	X
No verification table	O	O	O	O
Session key agreement	O	O	O	O
Mutual authentication	O	X	X	O
Single registration	O	O	O	O
User's anonymity	O	O	O	X
Two factor security	O	O	X	X
Prevention of registration center spoofing	O	O	X	X
Prevention of server spoofing	O	X	X	X

6. Conclusions

In this paper, we have shown that Hsiang et al.'s secure dynamic ID based remote user authentication scheme is vulnerable to a masquerade attack, server spoofing attack, and is not easily reparable. Furthermore, it cannot provide mutual authentication. Then we propose an improved scheme with anonymity to remedy these weaknesses. We demonstrate that our scheme can satisfy all of the essential requirements for multi-server environment. Our improved scheme is more secure and efficient, compare with Hsiang et al.'s scheme and other schemes. Therefore, our improved scheme will be practicable in the future.

Acknowledgment

This research was partially supported by the National Science Council, Taiwan, ROC, under contract no: NSC99-2221-E-030-022. Our gratitude also goes to Dr. Timothy Williams, Asia University.

References

Argyroudis, P. G., Verma, R., Tewari, H., & O'Mahony, D. (2004). Performance analysis of cryptographic protocols on handheld devices. In *Proceedings of the third IEEE international symposium on network computing and applications (NCA2004), Cambridge, USA* (pp. 169–174).

Chang, C. C. & Lee, J. S. (2004). An efficient and secure multi-server password authentication scheme using smart cards. In *Proceedings of the 2004 IEEE international conference on cyberworlds* (pp. 417–422).

Hsiang, H. C., & Shih, W. K. (2009). Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment. *Computer Standard & Interfaces, 31*(6), 1118–1123.

Hwang, T., Chen, Y., & Laih, C. S. (1990). Non-interactive password authentications without password tables. In *Proceedings of IEEE region 10 conference on computer and communication systems* (pp. 429–431).

Hwang, T., & Ku, W. C. (1995). Reparable key distribution protocols for Internet environments. *IEEE Transaction on Consumer Electronics, 43*(5), 1947–1949.

Juang, W. S. (2004). Efficient multi-server password authenticated key agreement using smart cards. *IEEE Transaction on Consumer Electronics, 50*(1), 251–255.

Ku, W. C., & Chen, S. M. (2004). Weaknesses and improvements of an efficient password based remote user authentication scheme using smart cards. *IEEE Transaction on Consumer Electronics, 50*(1), 204–207.

Lamport, L. (1981). Password authentication with insecure communication. *Communication of ACM, 24*, 770–772.

Lee, W. B., & Chang, C. C. (2000). User identification and key distribution maintaining anonymity for distributed computer network. *International Journal of Computer Systems Science & Engineering, 15*(4), 211–214.

Liao, Y. P., & Wang, S. S. (2009). A secure dynamic ID based remote user authentication scheme for multi-server environment. *Computer Standard & Interfaces, 31*(1), 24–29.

Li, L., Lin, I., & Hwang, M. (2001). A remote password authentication scheme for multi-server architecture using neural networks. *IEEE Transaction on Neural Network, 12*(6), 1498–1504.

Lin, I. C., Hwang, M. S., & Li, L. H. (2003). A new remote user authentication scheme for multi-server architecture. *Future Generation Computer Systems, 1*(19), 13–22.

Passing M., & Dressler, F. (2006). Experimental performance evaluation of cryptographic algorithms. In *Proceedings of the third IEEE international conference on mobile adhoc and sensor systems (MASS), Vancouver, Canada* (pp. 882–887).

Shen, J. J., Lin, C. W., & Hwang, M. S. (2003). A modified remote user authentication scheme using smart cards. *IEEE Transaction on Consumer Electronics, 49*(2), 414–416.

Sun, H. M. (2000). An efficient remote user authentication scheme using smart cards. *IEEE Transaction on Consumer Electronics, 46*(4), 958–961.

Wong, D. S., Fuentes, H. H., & Chan, A. H. (2001). The performance measurement of cryptographic primitives on palm devices. In *Proceedings of the 17th annual computer security applications conference (ACSAC 2001), New Orleans, USA* (pp. 92–101).

Wu, T. S., & Hsu, C. L. (2004). Efficient user identification scheme with key distribution preserving anonymity for distributed computer networks. *Computers & Security, 23*, 120–125.

Yang, Y., Wang, S., Bao, F., Wang, J., & Deng, R. (2004). New efficient user identification and key distribution scheme providing enhanced security. *Computer & Security, 23*(8), 697–704.