An ID-based remote user authentication scheme without using smart cards for multi-server environment

一種多伺服器環境下以身份為基礎且不需智慧卡的遠端使用者驗證方案

廖一評

大同大學通訊工程研究所

聖約翰科技大學資訊工程系

Email: newsun87@mail.sju.edu.tw

汪順祥

大同大學通訊工程研究所

Email: sswang@ttu.edu.tw

Abstract- The issue of remote user authentication scheme using smart cards for multi-server environment has been received much attention recently. Smart cards however are far from ubiquitous since some obstacles have restricted their practical applications. In this paper, we first propose an ID-based remote user authentication scheme without using smart cards for multi-server environment. The proposed scheme uses one-time password authentication to enhance the security of password. Furthermore, self-certified public key (SCK) is introduced to reduce the cost of public key management. The proposed scheme makes security analysis and compares functionality with other schemes. The results show that our scheme not only retains all advantages of robust authentication scheme for multi-server environment but also offers several nice properties such as user's identity protection and forward secrecy.

Key words: Smart cards; Multi-server environment; ID-based; Self-certified public key.

摘要-針對多伺服器環境的遠端使用者驗證方案的議題近來受到相當的重視。然而一些實際障礙限制了智慧卡的應用範圍，使得智慧卡無法普及。在本篇文章我們首度提出一個針對多伺服器環境，以身份為基礎且不需智慧卡參與的遠端使用者驗證方案。本方案使用一次密碼驗證來加強密碼的安全性。此外，本方案也引入自我驗證公鑰來減輕公鑰管理的負擔。該方案經過了安全的分析並與其他的方案作功能的比較，結果顯示不僅可以保有多伺服器環境強固的安全特性而且也提供一些額外的功能，例如身份隱藏及向前的祕密。

關鍵字: 智慧卡；多伺服器環境；身份基礎；自我驗證公鑰。

I. Introduction

Remote user authentication becomes an important issue for accessing the remote server's sources securely. Password authentication is one of the simplest and the most common authentication mechanism over an insecure channel since it allows people to choose and remember their own passwords without any assistance device. In 1981, Lamport proposed a novel password authentication scheme using cryptography hash functions [1]. A common feature of conventional password authentication schemes is that a verification cable,

which contains the verifiers of user's password. Under this situation, verification table is vulnerable to some risks, such as tampering and stolen-verifiers. To reduce these risks and maintenance cost, many password-based remote user authentication schemes using smart cards have been proposed without the password table in single server environment [2-5].

Recently, with the rapid growth of Internet service, more and more network architectures are used for multi-server environment. Hence, the issue of remote user authentication scheme using smart cards for multi-server environment has been received much attention. However, these designed schemes for the single server are not well suited for the multi-server environment. For example, if a user wants to access multiple service servers, it is infeasible to remember several identifiers and the corresponding passwords. Besides, it is an important topic for the secret keys distribution among the involved parties. Until to now, several papers have been devoted to the study of accessing the resources of multi-server network securely [6-13]. Taking computational cost into consideration, those schemes are divided into broad categories, one employs public-key cryptosystems and the other one employs only simple one-way hash function combined with symmetric cryptosystem. However, these published papers still have some weakness unsolved.

In general, a remote user's authentication scheme aiming at multi-server environment should satisfy the following merits [10]. (1) Single registration; (2) No password table; (3) Keeping free from the serious time synchronization problem; (4) Changing the password securely and freely (5) Preventing various well-known attacks such as guessing attack, forgery attack, server spoofing attack, etc. (6) Efficient performance for the users with low power computing devices. However, since high cost of the cards and the availability of card readers restrict the application of smart card. On the other hand, researchers assume that the authentication information stored in the smart card may be acquired by analyzing the leaked information [14] or monitoring the power consumption [15]. Hence, it leads to security flaws due to the leak of the secrets stored in the smart card. For example, the adversary may obtain the secrets to launch off-line password guessing attack or forgery attack. Furthermore, the system's reparability is also taken into consideration in practice [2]. These problems have restricted the application of smart cards to the small fields such as financial transactions. With the growth of portable storage devices such as USB memory thumbs, they are now common in offices anywhere and everywhere that today's mobile workers go, but they lacks for tamper-resistant property. Hence, the password-based authentication schemes using smart cards can not directly be applied to the remote user authentication without using smart cards.

In this paper, we first propose an ID-based remote user authentication scheme without using smart cards for multi-server environment. The proposed scheme uses one-time password authentication to enhance the security of password [16]. Our scheme provides a practical remote user authentication scheme while retaining all advantages of robust remote user authentication scheme for multi-server environment. Our scheme is highlighted with the following features: (1) It achieves mutual authentication and session key agreement; (2) It prevents from the security attacks due to the disclosure of the secrets stored in common storage device; (3) Each registered server does not maintain any verification table; (4) The public key of each registered server is authenticated without the need of explicit certificate; (5) The private key of each registered server can not revealed by the third trust party (TTP) and the other servers. The remainder of the paper is organized as follows. In section II, we give some preliminaries, including bilinear pairings and the related computational problems. Section III shows the details of the proposed scheme. After that, we make security analysis and make functionality comparisons among the related schemes in section IV and V. Finally, the conclusion is given in section VI.

II. Preliminaries

In this section, we introduce bilinear pairings and the related computational problems.

A. Bilinear pairings

Bilinear pairings namely the Weil pairings or Tate pairings may be used in important applications of cryptography. Suppose $< G_1, + >$ be an additive cyclic group of order q generated by P, where q is a prime and $< G_2, \times >$ a multiplicative cyclic group of the same order as in G_1 . A bilinear pairing is a map $\hat{e} : G_1 \times G_1 \to G_2$ on the elliptic curve. In view of shortness, the related properties are omitted and referred to [ ].

B. Computation problems

For providing higher security level of the proposed authentication scheme, some important mathematical assumptions are introduced on elliptic curves.

B-1 Elliptic Curve Discrete Logarithm Problem (ECDLP)

Given $Q = k * P$, where P, Q$\in G_1$. It is relatively ease to calculate Q given k and P, but it is relatively hard to determine k given Q and P.

B-2 Computational Diffie-Hellman Problem (CDHP)

For $a, b \in Z_q^*$, given $P, aP, bP \in G_1$, it is hard to find abP.

III. The proposed scheme

In this section, we propose an ID-based remote user authentication without using smart card for multi-server environment. Without loss of generality, the system's parties are composed of one registration center (RC), m users (U_i) and n service servers (S_j) . Our scheme involves three-party authentication key exchange (3P-AKE) protocol and is divided into some phases, including setup phase, registration phase, login phase, verification phase and password change phase. Different phases of work are described as follows and shown in Fig. 1-3.

A. Setup Phase

Let G_1 be an additive cyclic group of a prime order q generated by P and G_2 be a multiplicative cyclic group of the same order. Define $H : \{0,1\}^* \to G_1$ and $h : \{0,1\}^* \to \{0,1\}^n$ be cryptographic hash functions.

When RC permits the entry of S_j , RC and S_j cooperate to generate the key pair of S_j using self-certificated public keys (SCK), which are an efficient alternative to certificate based Public Key Infrastructure (PKI) [17]. Under this situation, SCK can reduce communication and management overheads of system's public keys. Instead of verifying public key using an explicit signature on the corresponding public key, the server's public key based on SCK is obtained using his identity along with its public key parameter without the need of concrete certificate. The protocol is described below.

S1: Private Key generation: S_j chooses a random number $k_j \in Z_q^*$ and computes $K_j = k_j P$. And then sends K_j and corresponding identity SID_j to RC over a secure channel. After receiving K_j and SID_j , RC checks if S_j is eligible. If yes, RC assigns a random number r_j , and computes $R_j = K_j + r_j P$. Finally, RC uses his own secret key s_{RC} to compute the signature parameter $\bar{x}_j$ as follows:

$$\bar{x}_j = h(SID_j \| R_j) s_{RC} + r_j \qquad (1)$$

Then, $\bar{x}_j$ is transmitted securely to S_j . After that, S_j obtains the corresponding private key as follows:

$$x_j = \bar{x}_j + k_j \qquad (2)$$

S2: Public key Extraction: Through the above pre-deployment, the corresponding public key Pub_j of S_j can be computed by everyone who acquires the public key parameter R_j and SID_j from the public server's registration table. Under this situation, Pub_j can be obtained as following equation:

$$Pub_j = h(SID_j \| R_j) Pub_{RC} + R_j \quad (3)$$

Equation (3) can be proved as follows:

$$
\begin{aligned}
Pub_j &= x_j P \\
&= (\bar{x}_j + k_j)P \\
&= (h(SID_j \| R_j)s_{RC} + r_j)P + k_j P \\
&= h(SID_j \| R_j)s_{RC}P + (r_j P + k_j P) \\
&= h(SID_j \| R_j)Pub_{RC} + R_j
\end{aligned}
$$

B. Registration phase

If U_i wants to access the resource of the system, he performs some steps during registration phase.

R1: U_i selects his identity ID_i and password PW_i. Next, U_i chooses a random number b_i, computes the hashed password $hpw_i = h(PW_i \| b_i)$, and sends $< ID_i, hpw_i >$ to RC over a secure channel.

R2: After receiving $< ID_i, hpw_i >$ at time T_i, RC checks if the user U_i is a registered user. If no, create an entry for U_i in the registration table and stores ID_i, $H(ID_i)$ and $T_{reg} = T_i$ in this entry; otherwise, only update the value of T_{reg} with time T_i in the existing entry for U_i. In this paper, T_{reg} is denoted as the registration time for a new user or re-registered user. Next, RC computes the identity signature S_{ID_i} with $(s_{RC}H(ID_i))$ and $\mathrm{Re}\, g_{PW_i}$ with $(hpw_i^{-1}S_{ID_i})$. After that, RC delivers $\{\mathrm{Re}\, g_{PW_i}, T_i\}$ to U_i over a secure channel.

R3: U_i stores $\{\mathrm{Re}\, g_{PW_i}, T_i\}$ along with b_i into a common storage device.

C. Login phase

Whenever the user U_i wants to access the sources of S_j, he performs the following steps.

L1: U_i submits ID_i, PW_i and SID_j. After that, U_i generates a random number $n_i \in Z_q^*$ and computes $N_i = n_i P$. And then $p_i = h(N_i)$, $L_i = n_i Pub_{RC}$ and $k_i = h(ID_i \| SID_j \| p_i \| T_i)$ is calculated.

L2: U_i computes the hashed password $hpw_i = h(PW_i \| b_i)$, the dynamic identity $CID_i = p_i H(ID_i)$, $TID_i = k_i \mathrm{Re}\, g_{PW_i}$ and one-time password $TPD_i = (hpw_i \cdot p_i)P$. Finally, U_i sends $< CID_i, TID_i, TPD_i, L_i >$ to S_j over a public channel.

D. Verification phase

1) Authentication of server and RC

After receiving the login message $< CID_i, TID_i, TPD_i, L_i >$, S_j and RC will run the following steps to achieve mutual authentication. Furthermore, RC is responsible for the verification of U_i. Once the identity of U_i is assured, S_j can derive the secret key TK_{ij} shared with U_i. The procedures are discussed below.

V1: S_j chooses a random number $n_j \in Z_q^*$ and computes $N_j = n_j P$. Next, S_j computes the long-term shared secret key $AK_1 = x_j Pub_{RC}$ on the security of CDHP.

V2:

S_j computes $Auth_j = h(SID_j \| AK_1 \| N_j \| L_i)$ and sends $< CID_i, TID_i, TPD_i, L_i, N_j, SID_j, Auth_j >$ to RC.

V3: According to SCK mentioned above, after acquiring the public parameter R_j from the public server's registration table, RC computes the public key Pub_j of S_j as equation (3). Next, RC computes $N_{RC} = n_{RC}P$ and the long-term secret key $AK_2 = s_{RC}Pub_j$. Then, RC checks if the received $Auth_j$ is equal to $h(SID_j \| AK_2 \| N_j \| L_i)$. If yes, S_j is authentic; other, reject the connection.

V4: RC computes $p_i = s_{RC}^{-1} Pub_{RC}$ and extracts $H(ID_i)$ via $p_i^{-1}CID_i$. And then check if $H(ID_i)$ exists in user's registration table. Next, RC computes $k_i = h(ID_i \| SID_j \| p_i \| T_i)$ and checks if whether $\hat{e}(TID_i, TPD_i)$ is equal to $\hat{e}(H(ID_i), p_i k_i Pub_{RC})$. If both checks are correct,

the identity of U_i is assured and continue executing next step, reject otherwise.

V5: RC chooses a random number $n_{RC} \in Z_q^*$ and computes $N_{RC} = n_{RC}P$. After that, RC computes $TK_{ij} = h(ID_i \| SID_j \| p_i \| N_j)$ $Auth_{RC} = h(SID_j \| AK_2 \| N_j \| N_{RC} \| L_i)$, $C_1 = Auth_{RC} \oplus TK_{ij}$ and $C_2 = h(Auth_{RC} \| TK_{ij})$. Finally, RC sends $< C_1, C_2, N_{RC} >$ to S_j .

V6: S_j computes $Auth_{RC}^* = h(SID_j \| AK_1 \| N_j \| N_{RC} \| L_i)$ and $TK_{ij}^* = C_1 \oplus Auth_{RC}^*$. And then verify whether C_2 is equal to $h(Auth_{RC}^* \| TK_{ij}^*)$. If yes, RC is authentic.

2) Authentication of server and user

After the authentication of server and RC, S_j derives TK_{ij} , which is the temporary secret key shared with U_i . Then S_j and U_i performs the following steps to achieve mutual authentication.

V7: S_j computes $C_3 = h(TK_{ij} \| N_j)$ and sends $< C_3, N_j >$ to U_i .

V8: U_i checks the validity of S_j by way of computing $TK_{ij}^* = h(ID_i \| SID_j \| p_i \| N_j)$ and comparing C_3 with $h(TK_{ij}^* \| N_j)$. If they are equal, the identity of S_j is assured. On the other hand, U_i chooses a random number $n_i^{'}$ and computes $N_i^{'} = n_i^{'}P$ Next, he calculate $C_4 = h(TK_{ij}^* \| N_j \| N_i^{'})$ and sends $< C_4, N_i^{'} >$ to S_j .

V9: After receiving $< C_4, N_i^{'} >$ from U_i , S_j calculates $h(TK_{ij} \| N_j \| N_i^{'})$ and compares it with the received C_4 . If they are equal, U_i is authentic. At the same time, both S_j and U_i will store the common session key $SK_{ij} = h(n_i^{'} N_j) = h(n_j N_i^{'})$ for the sequential sensitive information protection.

E. Password change phase

If the user U_i wants to change his password for some reason, first he sends a password change request to RC . When RC believes that the user's identity is U_i and finish mutual authentication between them, they obtain a common session key sk_{ij} . We omit the authentication process since it is similar to that mentioned above. Next, U_i performs the following step to update the password.

PC1: U_i selects PW_i^{new} and computes $nhpw_i = h(PW_i^{new} \| b_i)$. And then compute $PC_{i1} = sk_{ij} \oplus nhpw_i$ and $PC_{i2} = h(sk_{ij} \| nhpw_i)$. After that, U_i send $< PC_{i1}, PC_{i2} >$ to RC .

After receiving $< PC_{i1}, PC_{i2} >$, RC uses the session key sk_{ij} to perform the following step.

PC2: RC extracts the new hashed password $nhpw_i$ with $PC_{i1} \oplus sk_{ij}$. Check if the authentication tag PC_{i2} is equal to the computed $h(sk_{ij} \| \oplus nhpw_i)$. If yes, compute the new authentication information $Reg_{PW_i}^{new} = nhpw_i^{-1} S_{ID_i}$, $PC_{i3} = sk_{ij} \oplus Reg_{PW_i}^{new}$ and $PC_{i4} = h(sk_{ij} \| \oplus \mathrm{Re}\, g_{PW_i}^{new})$.

PC3: Similarly, RC extracts $Reg_{PW_i}^{new}$ with $PC_{i3} \oplus sk_{ij}$ and checks the correctness of PC_{i4} with the computed $h(sk_{ij} \| \oplus \mathrm{Re}\, g_{PW_i}^{new})$. If the validity of PC_{i4} is confirmed, Reg_{PW_i} is replaced with $Reg_{PW_i}^{new}$ in the common storage device.

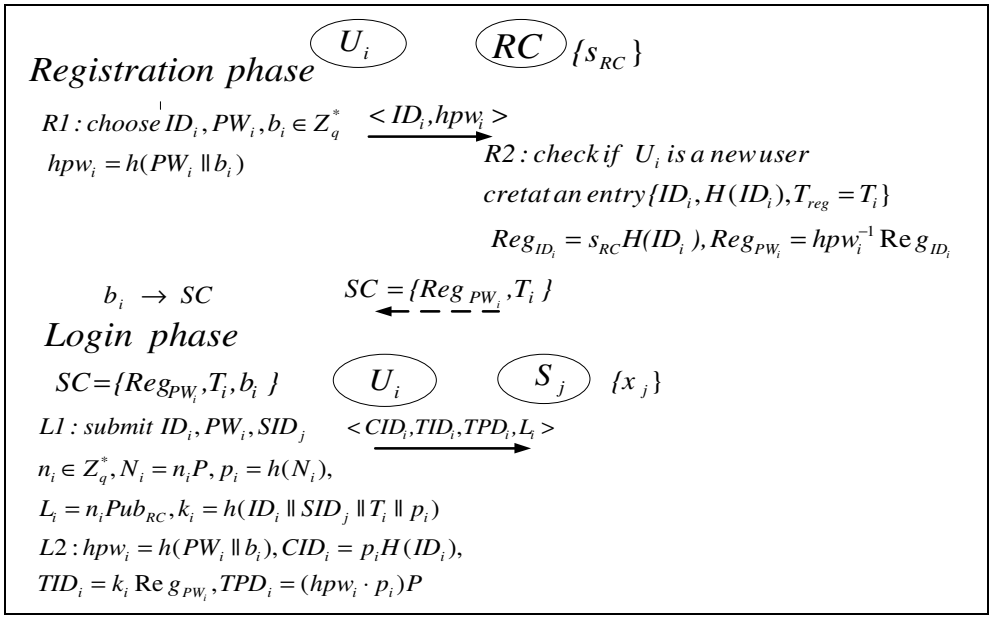

Fig.1 Registration phase

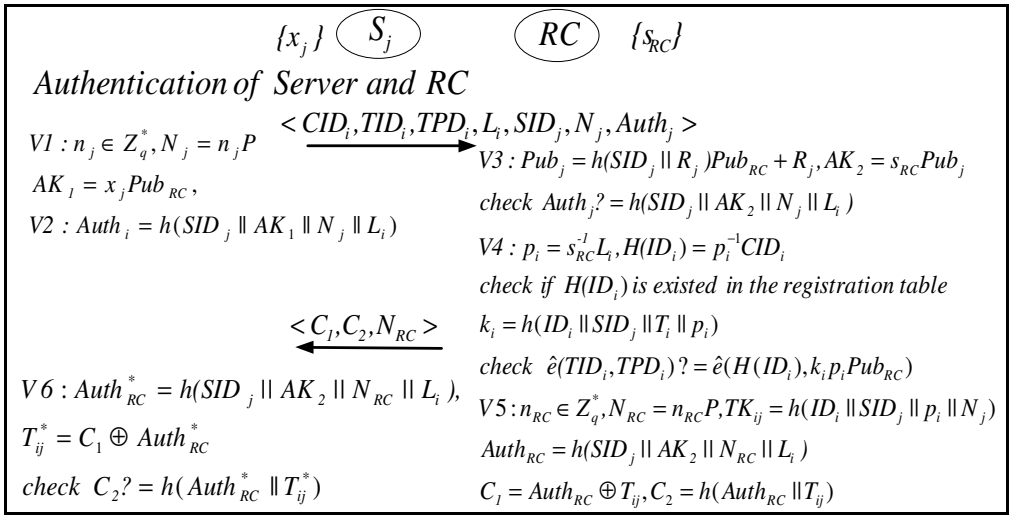

Fig.2 Authentication of server and RC in verification phase

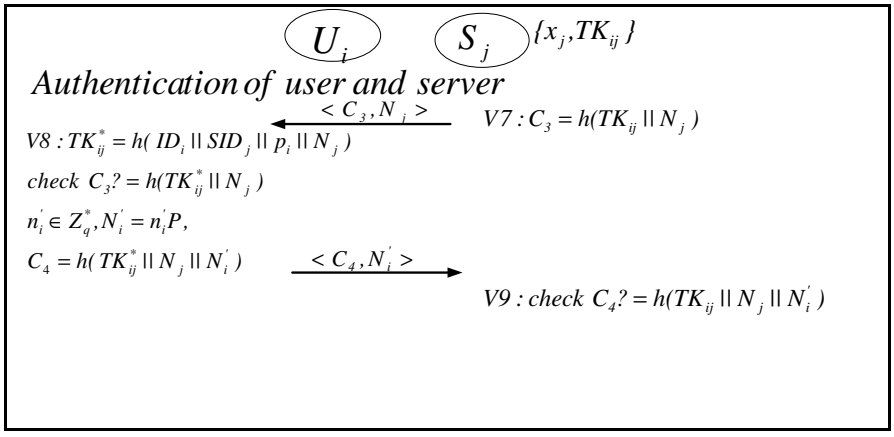

Fig.3 Authentication of server and user in verification phase

IV. Security analysis

In this session, let us discuss the security of the proposed scheme. It aims at matching all of the criteria for robust remote user authentication protocol. Furthermore, the proposed scheme can offer nice properties such as user identity and forward secrecy.

A. Satisfy the criteria of robust remote user authentication protocol

Theorem1: Our scheme achieves mutual authentication and session key agreement.

Proof: According our scheme, U_i sends the login message $< CID_i, TID_i, TPD_i, L_i >$ to S_j . After receiving the login message, S_j requests RC to verify the identity of U_i. To recognize the identity of both sides, S_j and RC achieve mutual authenticated using hash message authentication code (HMAC) since they can compute the common secret key $AK_1 (AK_2)$ on the security of CDHP. After that, U_i is authenticated by RC and S_j using the following operations.

(1) RC checks if the computed $\hat{e}(TID_i, TPD_i)$ is equal to the computed $\hat{e}(H(ID_i), k_i p_i Pub_{RC})$ based on BLS short signature scheme [18]. If yes, RC check the legality of S_j and response message $< C_1, C_2, N_{RC} >$ to S_j . Next, S_j can derive the secret key TK_{ij} shared with U_i from $< C_1, C_2, N_{RC} >$.

The verification works because of the following deduction:

$$\hat{e}(TID_i, TPD_i) = \hat{e}(k_i \operatorname{Re} g_{PW_i}, (hpw_i \cdot p_i)P)$$
$$= \hat{e}((k_i \cdot hpw_i^{-1})S_{ID_i}), (hpw_i \cdot p_i)P)$$
$$= \hat{e}(k_i S_{ID_i}, p_i P) = \hat{e}(k_i s_{RC} H(ID_i), p_i P)$$
$$= \hat{e}(H(ID_i), k_i p_i s_{RC} P)$$
$$= \hat{e}(H_1(ID_i),), k_i p_i Pub_{RC})$$

(2) The server S_j authenticates the identity of U_i by checking the validity of C_4 with the temporary secret key TK_{ij} .

Similarly, S_j is authenticated by checking the validity of C_3 with the temporary secret key TK_{ij} .

After the verification is finished, S_j an U_i can negotiate the session key sk_{ij} on the security of CDHP.

Theorem 2: Our scheme does not keep any verifier table in the server and RC .

Proof: In our scheme, RC checks if not only U_i is eligible but also $\hat{e}(TID_i, TPD_i)$ is equal to the computed $\hat{e}(H(ID_i), k_i p_i Pub_{RC})$. It is obvious that RC verifies the user's identity without any verification table or password table.

Theorem 3: Our scheme does not require time synchronization and delay time limitation.

Proof: To use timestamps for authentication, all parties must maintain local clocks that are periodically synchronized in a secure manner with a reliable source of time. Between synchronizations with the reliable time source, local clocks may drift. In our scheme, the transmitted messages among the parties have no concern with timestamps.

Theorem 4: Our scheme allows the user to choose password freely and update password securely.

Proof: In our scheme, U_i can select his favorite string and submit it to RC in registration phase. Furthermore, U_i must validate the old password to start password change.

Theorem 5: Our scheme withstands the following well-known attacks.

(1) Replay attack

Proof: The adversary may replay the same message of the receiver or the sender from a previous session to pass the verification of the system. He may relay

the previous message to masquerade U_i or S_j. Clearly, it cannot works because our scheme involves the temporary secret key TK_{ij} $(= h(ID_i \| SID_j \| p_i \| N_j))$ to recognize the identity of both sides. No one besides U_i or S_j can derive TK_{ij}.

(2) Impersonation attack

Proof: The adversary may intercept and analyze the login message aiming at the legal user. Next, he constructs a valid login message to pass the verification of the system. According to our scheme, if the adversary, i.e., U^a, obtains the private key s_{RC} of RC or the identity signature S_{ID_i} of the user U_i, he can construct a valid login request message. This former case obviously can be ruled out since the private key s_{RC} is kept secret by RC. If U^a intercepts previous login request message $< CID_i, TID_i, TPD_i, L_i >$, he cannot derive the user's identity S_{ID_i} from $TID_i (= k_i \cdot hpw_i^{-1}) S_{ID_i}$ without knowing k_i and hpw_i.

(3) Portable storage device loss attack

Proof: With the rapid growth of flash memory, the current trend for portable storage devices are towards small size. So, the results easily bring about other attacks. If the mobile storage device of U_{i_i} is lost or stolen for some reason, password guessing attack is effective and powerful among various attacks. The key to password guessing attack determines that the attacker is able to verify the correctness of the guessed password. In our scheme, the adversary may steal the authentication information $\{\mathrm{Re}\, g_{PW_i}, T_i, b_i\}$ stored in a common storage device or intercepts the login request message $< CID_i, TID_i, TPD_i, L_i >$. Even if $\{\mathrm{Re}\, g_{PW_i}, T_i, b_i\}$ is stolen or the login request message $< CID_i, TID_i, TPD_i, L_i >$ is intercepted, they cannot leak any redundancy to verify the guessed password. Hence, off-line password guessing attack fails. On the other hand, the adversary may guess the password corresponding to the portable storage device by way of typing the guessed password. Since our scheme can validate the guessed password, the number of guessing password can be restricted to withstand online guessing attack.

(4) Malicious insider attack

Proof: In general, the insider of the system is assumed to be trusted. However, the insider attack should be taken into consideration for real environment. We summarize the published schemes [6-13] and class insider attack into two types as follows.

- *Insider attack from RC*

Proof: As we know, the user U_i submits the identity ID_i and $hpw_i (= h(PW_i \| b_i))$ to RC. Because the privileged insider cannot derive the password PW_i from hpw_i without knowing b_i, he cannot masquerade U_i to access the resources of the other system using the password PW_i.

- *Insider attack from the server*

In the following, we show that the insider of the server S_j with the secret value $h(ID_i \| SID_j \| p_i \| N_j)$ cannot masquerade U_i to cheat other server S_k. We assume that the privileged insider replays the previous message $< CID_i^*, TID_i^*, TPD_i^*, L_i^* >$ to S_k. After achieving mutual authentication with RC, the server S_k can derive the secret key $h(ID_i \| SID_j \| p_i \| N_j^{'})$ shared with U_i. Because previous nonce N_j is not equal to $N_j^{'}$, the insider cannot compute $h(ID_i \| SID_j \| p_i \| N_j^{'})$ by himself. In other word, the insider of S_j cannot pass the verification of S_k without knowing the common secret key.

On the other hand, the insider of S_j cannot masquerade other server S_k since the long-term private x_k of S_k cannot be derived based on SCK.

(5) High reparability

Proof: It is assumed that the portable storage device

is lost or stolen. Once the corresponding password is leaked, the adversary can masquerade the legal user to access the server's resource. Under this situation, our scheme can allow the user to submit another selected password without changing the user's identity. After receiving the user's request for registration, *RC* only updates the registration time and submits the related parameters to the user over a secure channel.

B. Offer nice properties

(1) Protecting user's identity

Proof: If the adversary wants to trace the legal user, he may intercept and analyze the transmitted message in a public key channel. In our scheme, U_i sends a login message $< CID_i, TID_i, TPD_i, L_i >$ to S_j . If the adversary analysis CID_i , it is infeasible to recognize the identity of U_i since $H(ID_i)$ is protected with p_i . Moreover, $< CID_i, TID_i, TPD_i, L_i >$ is dynamic since n_i is different in each session. Therefore, our scheme can achieve user's anonymity [6].

(2) Forward secrecy

Proof: Forward secrecy is defined as the assurance that any previous session keys will not compromised if the system's secrets are leaked. In our scheme, any session keys $sk_{ij} (= h(n_i' n_j P))$ is dynamic in each session and unconcerned with the system's secrets such as s_{RC} or x_j .

V. Functionality comparison

In this section, we make functionality comparison between our scheme and other related schemes in Table 1. Obviously, it demonstrates that our scheme can offer nice properties while retaining all advantages of robust authentication scheme for multi-server environment. As for performance analysis, we focus on the computation cost. According to the proposed scheme, the computation cost is concerned with bilinear pairings operations, multiplication operations on elliptic curve, hash operations. Although the computation cost of our scheme is higher than that of hash-based authentication schemes. However, the computation cost of our scheme does not require expensive bilinear pairings operation or modular exponentiation operation at user's side. Hence, our scheme is well applied to the devices with the limited communication power.

Table 1 Functionality comparison between our scheme and other related schemes

	Ours	[13]	[12]	[10]	[11]
C1	◯	◯	◯	◯	X
C2	◯	◯	◯	◯	◯
C3	◯	◯	◯	◯	X
C4	◯	◯	◯	◯	X
A1	◯	◯	X	X	◯
A2	◯	◯	X	◯	X
A3	◯	◯	X	X	X
P1	◯	X	◯	◯	X
P2	◯	◯	X	X	X
P3	◯	◯	◯	X	X
P4	◯	X	X	X	X

C1: mutual authentication; C2: no verification table; C3: no time synchronization; C4: password updated securely and freely; A1: prevention of forgery attack; A2: prevention of server spoofing attack; A3: prevention of insider attack; P1: forward secrecy; P2: high reparability; P3: user's anonymity; P4: no smart cards cooperation.

VI. Conclusions

The issue of remote user authentication scheme using smart cards for multi-server environment has been received much attention recently. Although many authentication schemes using smart card for multi-server environment are presented successively, they are not suitable to the application without using smart card. In this paper, we first propose an ID-based remote user authentication scheme without using smart card for multi-server environment. Furthermore, the proposed scheme first uses one-time password authentication to enhance the security of password for multi-server environment. We show that our scheme not only retains all advantages of robust authentication scheme but also offers several nice properties, such as user's identity protection and forward secrecy. Moreover, our scheme involves SCK to manage the public keys

among the different service servers without concrete certificate. Moreover, the security of the private keys among the different servers can be achieved under the assumption that the insider of the registration center with privilege is untrusted.

References

[1] L. Lamport, Password authentication with insecure communication, Communication of the ACM 24 (1981) 28-30.

[2] C. Fan, Y. Chan, Z. Zhang, Robust remote authentication scheme with smart cards, IEEE Transactions on Consumer Electronics 50 (1) (2004) 204-207.

[3] T. H. Chen, W. B. Lee, A new method for using hash functions to solve remote user authentication, Computer and Electrical Engineering 34 (2008) 53-62.

[4] M. L. Das, A. Saxena, and V. P. Gulati, A dynamic ID-based remote user authentication scheme, IEEE Trans. Consum. Electron. 50 (2) (2004) 629-631.

[5] M. L. Das, A. Saxena, V. P. Gulati and D. B. Phatak, A novel remote user authentication scheme using bilinear pairings, Computers and Security 25 (3) (2006) 184-189.

[6] W. B. Lee, C. C. Chang, User identification and key distribution maintaining anonymity for distributed computer network, Computer System Science, 15(4) (2000) 211-214.

[7] W. J. Tsuar, C. C. Wu, W. B. Lee, A flexible User Authentication for Multi-server Internet Services, Networking-JCN2001LNCS, Springer-Verlag, 2093 (2001) 174-183.

[8] L. Li, I. Lin and M. Hwang, A remote Password Authentication Scheme for Multi-server Architecture Using Neural Networks, IEEE Trans. On Neural Networks, 12 (6) (2001) 1498-1504.

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

[10] W. S. Juang, Efficient multi-server password authenticated key agreement using smart cards, IEEE. Transactions on Consumer Electronics, 50 (1) (2004) 251-255.

[11] C. Chang, J. S. Lee, An efficient and secure multi-server password authentication scheme using smart cards, IEEE. Proceeding of the 2004 International Conference on Cyberworlds.

[12] Y. P. Liao, S. S. Wang, A secure dynamic ID-based remote user authentication scheme for multi-server environment, Computer Standards and Interfaces 31 (1) (2009) 24-29.

[13] H. C. Hsiang, and W. K. Shih, "Improvement of the secure dynamic ID based remote user authentication scheme for multi-server environment, Computer Standards and Interfaces (2009), accepted and in press.

[14] P. Kocher, J. Jaffe and B. Jun, Differential power analysis, proceeding of Advances in Cryptology (CRYPTO'99) (1999) 399-397.

[15] T. S. Messergers, E. A. Dabbish, and R. H. Sloan, Examining smart card security under the threat of power analysis attacks, IEEE Transactions on Computers 51 (5) (2002) 541-552.

[16] S. Luo, J. Hu, and Z. Chen, An identity-based one-time password scheme with anonymous authentication, IEEE. 2009 International Conference on Networks Security, Wireless Communications and Trusted Computing, (2009) 864-867.

[17] M. Girault, Self-certified public keys, Advances in Cryptology, Eurocrypt'91, Springer-Verlag, (1991) 491-497.

[18] M. L. Das, A. Saxena, V. P. Gulati and D. B. Phatak, "A novel remote user authentication scheme using bilinear pairings," Computers and Security, vol. 25, no. 3, pp. 184-189, 2006.